Official journal of the JHL 2689-9620
www.light-am.com

Review Open Access

# Deep holography

Guohai Situ[1,2*]

## Abstract

With the explosive growth of mathematical optimization and computing hardware, deep neural networks (DNN) have become tremendously powerful tools to solve many challenging problems in various fields, ranging from decision making to computational imaging and holography. In this manuscript, I focus on the prosperous interactions between DNN and holography. On the one hand, DNN has been demonstrated to be in particular proficient for holographic reconstruction and computer-generated holography almost in every aspect. On the other hand, holography is an enabling tool for the optical implementation of DNN the other way around owing to the capability of interconnection and light speed processing in parallel. The purpose of this article is to give a comprehensive literature review on the recent progress of *deep holography*, an emerging interdisciplinary research field that is mutually inspired by holography and DNN. I first give a brief overview of the basic theory and architectures of DNN, and then discuss some of the most important progresses of deep holography. I hope that the present unified exposition will stimulate further development in this promising and exciting field of research.

**Keywords:** Deep learning, deep neural networks, digital holography, computer-generated hologram, optical neural networks

## Introduction

Since the pioneering works by Gabor[1], Leith and Upatnieks[2,3] and Denisyuk[4], holography has become an important and widespread technique that has found applications in various fields of optical engineering, ranging from optical imaging and microscopy[5,6], metrology[5,7] to three-dimensional (3D) display[8,9].

Physically, holography is a two-stage process: the recording, and the reconstruction, of a wavefront. Nowadays, both these processes can be performed either optically or digitally. We refer the kind of holography that the recording is performed optically by a digital camera while the reconstruction is digitally as digital holography (DH)[5–7]. In contrast, the kind of holography that the recording (synthesis) is performed digitally while the reconstruction is optically is called computer-generated holography (CGH)[8,9].

For the optical recording of a wavefront, one would prefer a light source with a certain level of coherence[10,11], in particular, for off-axis holography[3], because the lack of coherent light sources only allows interference patterns to be formed in the vicinity of the optical axis. As a result, only the hologram of a small object can be recorded by using an in-line setup. Furthermore, the reconstructed image is blurred owing to the superposition of a fuzzy defocused twin image, which was then difficult to effectively eliminate[12], although many efforts have been elaborately made[13] in the history of holography. Thanks to the invention of DH[14,15], coherence is not a fundamental limit for contemporary holographic imaging techniques any more. Light sources with short-coherence[16,17] and even incoherence[18] can be used for holographic recording.

One of the great advantages of DH is the capability of numerical reconstruction of a digitally recorded hologram.

Correspondence: Guohai Situ (ghsitu@siom.ac.cn)
[1]Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai 201800, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China

In this way, the fuzzy defocused twin image superposing with the in-focus image can be removed numerically. Conventional this can be done by physics-based approaches[19–26], phase-retrieval approaches[27–31], or more generalized inverse problem approaches[32–40].

With the recent prosperous development of a new class of optimization tools called deep neural networks (DNN)[41,42], we have witnessed the emergence of a new paradigm of solving inverse problems in various fields of optics and photonics by using DNN[43–48]. This shift of paradigm also has significant influence to the field of DH[49,50] in many aspects. Indeed, in additional to holographic reconstruction[51–57], DNN has also been proposed for phase aberration compensation[58], focus prediction[59–64], extension of depth-of-field[65], speckle reduction[66–69], resolution enhancement[70], and phase unwrapping[71–73], just to name a few.

DNN has also been used for the design of CGH[74–81], a technique whose invention was mainly attributed to Lohmann's pioneering works[82,83]. As suggested by the name, the objective of CGH is to artificially encode a target object within a space volume into a hologram called computer-generated hologram so that it can reconstruct the desired wavefront within that space volume under the illumination of a proper coherent light. The optically reconstructed wavefront can be a perfect reference of an optical surface for holographic testing[84–86] or a 3D object/scene for holographic display[8,9]. Conventional approaches for the encoding of a computer-generated hologram are either to take it as an optimization problem, which can be solved by iterative phase-retrieval algorithms[27,28,87], or non-iterative interference-specific or diffraction-specific algorithms[88,89]. Although it can be sped up by using the trick of look-up table[90], the use of DNN still promises the most dramatic increment in terms of calculation efficiency[74–81].

Holography has been used the other way around, i.e., as a way to implement optical neural networks (ONN), in particular, the Hopfield model[91,92] and fully-connected neural networks[93–97]. With the development of optical material manufacturing technologies such as 3D printing and metamaterials, multi-layers fully-connected neural networks can be implemented in a modern fashion[98,99].

These recent progresses suggest that the distinct fields of holography and deep learning have incorporated into each other, forming a new interdisciplinary field, the name of which can be coined as *deep holography*. In this article, I will give a comprehensive literature review of this emerging but exciting field. The structure of this article is organized as follows: In section 1 I will first give a concise introduction to deep neural networks. Then I will discuss in detail how DNN is used to solve various problems in holography, and *vice versa*, in section 2 and section 3, respectively. Finally, the perspective of further development will be discussed in Sec. 4.

## Deep Neural Networks: A Concise Introduction

DNN can be regarded as a category of machine learning algorithms that are designed to extract information from raw data and represent it in some sort of model[42]. Specifically, a neural network (NN) is built on a collection of connected units called artificial neurons, which are typically organized in layers, an idea somehow inspired by the biological neuron in the mammalian brain. As schematically shown in Fig. 1a, a modern NN consists of three kinds of layers: the input layer, the output layer, and the hidden layers. The input layer usually represents the signal to be processed and the output layer represents the expected result that one wishes the network to produce. So the widths ($P$), i.e., the number of neurons, of these two layers are task-specific. Data processing is mainly performed by the hidden layers that lay between the input and the output layers. Each successive hidden layer uses the outputs from its upstream layers as its input, processes it, and then passes the result to a downstream layer. In this manuscript, we use the digit $l = 1,\ldots,L$ to enumerate the layers, where $L$ is called the depth of the NN. A neural network is *deep* if it has many layers. The depth of modern *deep* neural networks ranges from 8 layers in AlexNet[100] to 152 layers in ResNet[101], which has the potential to increase to more than 1000 layers[102]. The requirement of computation resource dramatically increases along with the up-scaling of the DNN, i.e., the number of hidden layers and hidden neurons. For example, a neural network used for DH have a depth up to 20 layers in a typical proof-of-principle demonstration[52]. It usually takes tens of hours to train on a training set consisting of about thousands of holograms with a modern graphic workstation. Unfortunately, given a problem to be solved by DNN, it is not trivial at all to determine how deep it should be[103,104]. Hornik has proved that, for any continuous function $\boldsymbol{y} = f(\boldsymbol{x})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are data (vectors) in the Euclidean or non-Euclidean space, there is always an NN, no matter how shallow it is, that can approximate the function $f$ with an infinitesimal error, i.e., $\mathrm{NN}\{\boldsymbol{x}\} \to \boldsymbol{y}$, provided that it is sufficiently wide[105]. Practically, however, one still needs a good rule of thumb to configure the number of layers ($L$) and the numbers of neurons in each layer ($P^{(l)}$). It is commonly believed that the performance of DNN is heavily dependent on the network architecture, which is defined in part by $L$, $P^{(l)}$, and the types of connections between layers, the quality of the raw data, and the

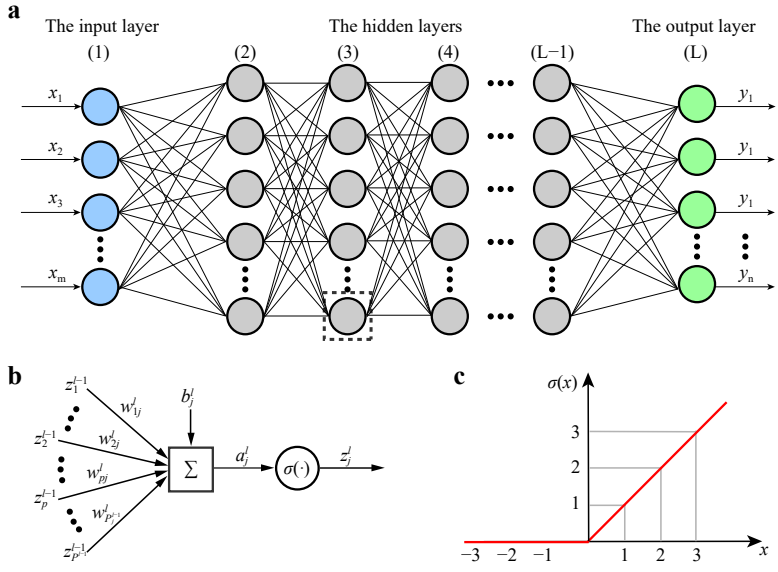


**Fig. 1** A conceptual architecture of DNN.

technique to train the network on them.

Perhaps the most well-known and easiest to understand DNN is the so-called feedforward neural networks. The architectures of all the other DNNs that are widely used in holography[51–56,58–62,64–81] are developed on the base of it. Thus it is worthy of discussing it in detail.

## Feedforward neural networks

As shown in Fig. 1a, a feedforward neural network, or multilayer perceptron (MLP) has one input layer, one output layer, and one or many hidden layers. Each layer may have a different number of neurons called the perceptron. The connections between the neurons in the layers form an acyclic graph[106]. The objective of a feedforward neural network is to optimize an NN model $f_{\mathrm{NN}}$ that approximates a continuous function $f$, which maps $\boldsymbol{x}$ in the input space to $\boldsymbol{y}$ in the output space through a set of parameters $\Theta$ that are learned from the raw data.

## The artificial neuron

The basic unit in a DNN is the artificial neuron. As shown in Fig. 1b, an artificial neuron simply calculates the weighted sum of all the quantities outputted from the neurons in its immediately upstream layer, and passes the resulting quantities to the neurons in the next layer. Let us take the $j^{\mathrm{th}}$ neuron at the $l$ layer for example, the input to this neuron can be written as[41,42]

$$a_j^{(l)} = \sum_{p=1}^{P} w_{pj}^{(l)} z_p^{(l-1)} + b_j^{(l)} \tag{1}$$

where $z_p^{(l-1)}$ is the output from the $p^{\mathrm{th}}$ neuron at the $(l-1)^{\mathrm{th}}$ layer, $w_{pj}^{(l)}$ is the weighting factor that connects these two neurons, and $b_j^{(l)}$ is a bias. The values of the network parameters $w_{pj}^{(l)}$ and $b_j^{(l)}$ are to be learned from a set of raw data called the training set. One can think of their values as the connection strengths between the two neurons. The $j^{\mathrm{th}}$ neuron at the $l$ layer then can be activated if the quantity $a_j^{(l)}$ is significant (for example, $>0$), and this value is passed on to the next layer. Otherwise, this neuron is dead, and should have no contribution to the neurons in the downstream layer. Analogously, one can think of the input signal being an electric current that flows through the network from the input layer to the output layer. Each neuron in the hidden layers acts like a gate that controls the amount of incoming current that is allowed to pass through to the downstream neurons. The "gate" function in a NN is not just a simple "0" and "1" binary function as in the digital electric circuit, but has a form of an activation function. Fig. 1c plots the Rectified Linear Unit (ReLU), which is one of the most important activation function nowadays used in DNN. It is defined as[42]

$$z = \sigma(a) \triangleq \max(0, a) \tag{2}$$

ReLU is widely employed in most of the modern neural network architectures, as it has a number of benefits over other old-fashioned activation functions such as Sigmoid and Tanh[42]: (a) It can be applied to minimize the interaction effects; (b) It is simple and easy to compute, and thus leads to an increment of efficiency in the network training; (c) It helps avoid the vanishing gradient problem; and (d) It is sparsely activated because the output is zero for all negative inputs. However, ReLU sometimes dies, referring

to the situation that an neuron has a zero activation value. This dying ReLU issue causes slow-learning because the optimization algorithm is gradient-based and does not adjust the unit weights if the gradient is zero in an inactive neuron. Thus, extensions and alternatives such as Leaky ReLU (or LReLU for short), exponential linear unit (ELU), and parametric ReLU (PReLU) are highly desirable when it happens[107].

The width of the input layer is the number of pixels of the image one wishes the network to process. The width of the output layer is usually task-dependent. For example, in the applications of holographic reconstruction[51–54,56], the width of the output layer is the same as the input layer. Whereas in the application of holographic autofocusing[59–62], the width of the output layer is simply 1, which gives the focusing distance. The width of each hidden layer is dependent on task in hand and the choice of the network architecture. Indeed, the width of the $l^{\text{th}}$ layer and that of the $(l-1)^{\text{th}}$ layer may not be the same in most of the cases. Thus, what Eq. 1 implies is that it transforms a $P$-dimensional signal to a $J$-dimensional space. This can be more clearly seen by writing Eq. 1 in the form of

$$\boldsymbol{a}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{z}^{(l-1)} + \boldsymbol{b}^{(l)} \tag{3}$$

The substitution of Eq. 1 into Eq. 2 yields the output from the $l$ layer

$$\boldsymbol{z}^{(l)} = f^{(l)}\left(\boldsymbol{z}^{(l-1)}; \boldsymbol{w}^{(l)}, \boldsymbol{b}^{(l)}\right) = \sigma\left(\boldsymbol{a}^{(l)}\right) \tag{4}$$

where $f^{(l)}$ is defined as the transform from the $(l-1)^{\text{th}}$ layer to the $l^{\text{th}}$ layer. From a more theoretical point of view, deep learning relies on this kind of mapping between spaces of different dimensions[108].

### The feedforward neural network model

Now we can mathematically express the feedforward neural network model as

$$\begin{aligned}\boldsymbol{y} = \delta\Big(\boldsymbol{W}^{(L)}\sigma\Big(\ldots\sigma\Big(\boldsymbol{W}^{(2)}\sigma\Big(\boldsymbol{W}^{(1)}\boldsymbol{z}^{(0)} + \boldsymbol{b}^{(1)}\Big) \\ + \ \boldsymbol{b}^{(2)}\Big) + \ldots\Big) + \boldsymbol{b}^{(L)}\Big)\end{aligned} \tag{5}$$

where $\boldsymbol{z}^{(0)} \triangleq \boldsymbol{x}$ is the input signal, and $\delta(\cdot)$, the activation function at the output layer. It is not necessary to be the ReLU function as in the hidden layer. For example, it takes the form of a softmax function

$$\delta(z_j) = \frac{\exp[z_j]}{\sum_{k=1}^{K}\exp[z_k]} \tag{6}$$

for autofocusing in holography[59–61].

The set of network parameters $\Theta$ then can be defined as $\Theta \triangleq \{\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \ldots, \boldsymbol{W}^{(L)}, \boldsymbol{b}^{(L)}\}$. Then one can write the feedforward NN model in Eq. 5 in a more compact form

$$\boldsymbol{y} = f^{(L)} \circ f^{(L-1)} \circ \ldots \circ f^{(1)} = f_{\text{NN}}(\boldsymbol{x}; \Theta) \tag{7}$$

This simply tells the fact that a feedforward NN model $f_{\text{NN}}$ is to approximate a function $f$ and map the input $\boldsymbol{x}$ to the output $\boldsymbol{y}$ through a neural network specified by the set of parameters $\Theta$.

### The training of the network

Although the universal approximation theorem[105] guarantees that a feasible NN model $f_{\text{NN}}$ exists for an arbitrary given training set, Eq. 7 does not provide any clue to its architecture and weight configuration. In terms of DNN, the network architecture is defined on a set of hyperparameters such as the depth $L$ and the width $P^{(l)}$ of each layer that one needs to set up, mostly by a rule of thumb. Many efforts have been made to clarify this point, but it is still an open question[103,104]. The weighing factors $\boldsymbol{W}$ and $\boldsymbol{b}$ are to be determined by a learning process, which consists of repeated steps of optimal adjustment of the parameters in $\Theta$.

For the supervised learning methods that are mainly used in the community of holography, the parameters in $\Theta$ are learned from a large set of labeled data $S = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{K}$. It consists of many pairs of $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ with $\boldsymbol{x}_i$ being the signal (such as a hologram) one wishes the network to process, and $\boldsymbol{y}_k$, the associated correct result (the reconstructed object, the focusing distance, etc.) that are already known. Thus it is possible to compare the calculated output, denoted by $\hat{\boldsymbol{y}}_k$, with the correct answer $\boldsymbol{y}_k$, and evaluate their difference for each neuron at the output layer. This leads one to define the loss function $\mathcal{L}[f_{\text{NN}}(\boldsymbol{x}; \Theta), \boldsymbol{y}]$. Thus one can then formulate the NN learning as the optimization of the parameters in $\Theta$ so as to minimize the loss function

$$\arg\min_{\Theta} \mathcal{L}[f_{\text{NN}}(\boldsymbol{x}; \Theta), \boldsymbol{y}] \tag{8}$$

An instinct philosophy to train a neural network is to adjust the values of $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ and see if the loss function decreases or not. An efficient and straightforward way to do this is to evaluate the gradient of the loss function with respect to $\Theta$. Note that DNN has a layered architecture, one needs to calculate the gradient of the loss function with respect to the weights and bias one by one from the output layer back to the input layer. This can be done by the algorithm of back propagation[109].

To develop the error back propagation model, let us first define the loss function of layer $l$

$$\mathcal{L}^{(l)} = \mathcal{L} \circ f^{(L)} \circ f^{(L-1)} \circ \ldots \circ f^{(l)} \tag{9}$$

Then the back gradient of the loss function with respect to the parameters $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ at layer $l$ can be formulated by using the recurrence relation[110]

$$\frac{\partial \mathcal{L}^{(l)}}{\partial \boldsymbol{W}^{(l)}} = \frac{\partial \mathcal{L}^{(l+1)}}{\partial f^{(l)}} \frac{\partial f^{(l)}}{\partial \boldsymbol{W}^{(l)}} \tag{10}$$

$$\frac{\partial \mathcal{L}^{(l)}}{\partial \boldsymbol{b}^{(l)}} = \frac{\partial \mathcal{L}^{(l+1)}}{\partial f^{(l)}} \frac{\partial f^{(l)}}{\partial \boldsymbol{b}^{(l)}} \tag{11}$$

$$\frac{\partial \mathcal{L}^{(l)}}{\partial \boldsymbol{z}^{(l-1)}} = \frac{\partial \mathcal{L}^{(l+1)}}{\partial f^{(l)}} \frac{\partial f^{(l)}}{\partial \boldsymbol{z}^{(l-1)}} \tag{12}$$

From the recurrence relations Eq. 10 – Eq. 12, one can derive $\partial \mathcal{L}/\boldsymbol{W}^{(l)}$ and $\partial \mathcal{L}/\boldsymbol{b}^{(l)}$ using the chain rule[111]. Then the architectural parameters at layer $l$ can be updated using the strategy of gradient descent[110]

$$\boldsymbol{W}^{(l)} \leftarrow \boldsymbol{W}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(l)}} \tag{13}$$

$$\boldsymbol{b}^{(l)} \leftarrow \boldsymbol{b}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^{(l)}} \tag{14}$$

where $\eta$ is the learning rate, or step size, in the gradient descent method. It determines how many the parameters should be adjusted each time. The convergence will not get to the right place if the learning rate is either too large or too small. Thus an ideal $\eta$ value is desirable in training a neural network. However, the determination of its value is yet a comprehensive theoretical study[112]. Empirically, setting the learning rate $\eta = 10^{-4}$ should work pretty well for many applications in holography[52,65]. But is can be adjusted during the iteration process as well[53,64].

The number of labeled data pairs ($K$) in the training set should be sufficiently large in order for the network to learn the statistics of the data. Indeed, it can easily go up to tens of thousands in a typical DNN for holographic reconstruction[52]. The calculation of the error back propagation is then extremely time-consuming. Thus a practical and intuitive way to evaluate the error is to randomly select a small batch of labeled data in each epoch (which means a period of time) and calculate the gradients of the loss function, and use them to update Θ. This is a trick called stochastic gradient descent (SGD). Furthermore, it is also possible to employ the method of adaptive moment estimation[113], or *Adam* for short, that adds a momentum term to speed up the learning process, and adaptively shrinks the learning rate along with the progress of the learning process to achieve faster convergence.

As apparently suggested by Eq. 10 – Eq. 12, the specific form of the loss function plays a crucial role in the optimization process and the explicit setting of Θ that it converges to[114]. Thus one should choose the most appropriate loss function depending on the problem in hand[62]. Some of the widely adopted loss functions in DNN include the averaged mean squared error (MSE) loss, the $L_1$, or mean absolute error (MAE), loss, and the cross entropy loss.The MSE loss is one of the defined as

$$\begin{aligned}\mathcal{L}[f_{\mathrm{NN}}(\boldsymbol{x};\Theta),\boldsymbol{y}] &= \mathbb{E}_S \|\boldsymbol{y}_k - f_{\mathrm{NN}}(\boldsymbol{x}_k;\Theta)\|^2 \\ &= \frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{y}_k - f_{\mathrm{NN}}(\boldsymbol{x}_k;\Theta)\|^2\end{aligned} \tag{15}$$

where the subscript $k$ denotes the $k^{\mathrm{th}}$ pair of data in the training set, and $\|\cdot\|^2$ denotes the Euclidean distance between the correct output $\boldsymbol{y}_k$ and the calculated output $\hat{\boldsymbol{y}}_k = f_{\mathrm{NN}}(\boldsymbol{x}_k;\Theta)$ with respect to a setting of Θ. The MSE loss, and the root of it, i.e., the root mean square error (RMSE) loss, are widely used to train DNN for holographic reconstruction[52,64]. Alternatively, the MAE loss is defined as[115]

$$\mathcal{L}[f_{\mathrm{NN}}(\boldsymbol{x};\Theta),\boldsymbol{y}] = \frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{y}_k - f_{\mathrm{NN}}(\boldsymbol{x}_k;\Theta)\| \tag{16}$$

where $\|\cdot\|$ is the $L_1$ norm. And the cross entropy loss is defined as the inner product of $\boldsymbol{y}_k$ and $\hat{\boldsymbol{y}}_k$

$$\mathcal{L}[f_{\mathrm{NN}}(\boldsymbol{x};\Theta),\boldsymbol{y}] = -\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}_k \cdot \log f_{\mathrm{NN}}(\boldsymbol{x}_k;\Theta) \tag{17}$$

When more than one criterion are concerned, one can defined a combined loss function that is a weighted sum of several parts[64,116]. This is in particular useful for holography because of the complex nature of an optical wavefront. For example, if one wishes to measure both the amplitude and phase of the reconstructed wavefront, he/she can define a loss function as $\mathcal{L} = \mathcal{L}_{\mathrm{amp}} + \alpha \mathcal{L}_{\mathrm{phase}}$, a linear combination of the errors in both the amplitude and phase[54]. Alternatively, one can also define a complex loss function[78].

I will show later on that a loss function does not have to define on the training set Θ, but on a physical model. That is, $\mathcal{L}\{H[f_{\mathrm{NN}}(\boldsymbol{y})],\boldsymbol{y}\}$, where $H$ is a forward physical model that maps the object space to the measured image space[55].

When the training process is completed, the performance of the neural network should be validated by using a set of data that have not been used for training in any way. The performance is usually evaluated by using the test error $\mathcal{L}_{\mathrm{test}} = \mathbb{E}_S \|\boldsymbol{y}_m, f(\boldsymbol{x}_m;\Theta)\|^2$. This metric also quantifies the ability of *generalization* of the trained network[42].

## The Convolutional Neural Networks

In the feedforward NN model described by Eq. 7, the neurons in neighboring layers are typically fully connected with the weight and bias parameters independent of each other. Although DNN has been employed to solve many problems in computational imaging[117–119], ranging from ghost imaging to imaging through scatterers, there are several issues with it. First, as there are too many parameters to train, it often has the issue of overfitting. Second, it requires a large memory footprint to temporally

store the parameter set $\Theta$ and thus the training usually takes a lot of time. Third, it ignores the intrinsic structure that the data to be processed may have. This is in particular important for the tasks of speech and image processing. Images, in particular, have significant intrinsic structures. For example, neighboring pixels may have similar values; the image may be shift-invariant, etc. It is therefore highly demanded to have units in a neural network to learn these features.

Inspired by the physiological mechanism of visual cortexes[120], a convolutional neural network (CNN) also has a layered structure. Indeed, it consists of an input layer, an output layer and multiple hidden layers. But the hidden layers in CNN do not have to be fully connected. Instead, each convolutional layer in CNN has a filter called kernel function, denoted by $w$, to convolve with the incoming data $z$ from an upstream layer, and extract a feature map of it at a certain level of abstraction. Instead of Eq. 1, the calculated feature map $a(i,j)$ can be mathematically written as[42]

$$a(i,j) = (z * w)(i,j) = \sum_m \sum_n z(m,n) w(i-m, j-n) \quad (18)$$

where $(i,j)$ and $(m,n)$ stand for the neurons at two neighboring layers. Equation (18) means that the elements of the kernel function, $w(m,n)$, will apply to many neurons in the layer. In other words, all those neurons share the weighting parameters in contrast to the case of DNN that each neuron is tied to a unique weight. This parameter sharing mechanism guarantees that the network just needs to optimize a much smaller set of parameters for each layer. It is because of this reason that the requirement for memory footprint and computation efficiency can be significantly reduced in comparison to DNN[42]. Indeed, the size of the kernel function is typically from $3\times3$ to $5\times5$ for many applications in holography[52,64], which is very small in comparison to that of a layer.

Note that a natural image has various features in one level of abstraction. For example, an image of a human face may contain edges with different orientations. Thus it is preferable to use multiple filters in one layer to extract all these edge orientation features, generating multiple feature maps. Usually these feature maps are arranged in a three-dimensional volume as they are to pass to a downstream layer. Denoting the width $M$ and height $N$ as the transverse size of each feature map, and the depth $U$ as the number of feature maps, the value of the $(i,j)^{\text{th}}$ pixel in the $t^{\text{th}}$ feature map in the $l^{\text{th}}$ convolutional layer ($l \geq 2$) can be written as

$$\begin{aligned} a^{(l)}(i,j;v) &= f^{(l)}\left(z^{(l-1)}; w^{(l)}, b^{(l)(t)}\right) \\ &= \sigma\Bigg(\sum_{u=1}^{U}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} w^{(l)}(m,n;u,v) \\ &\quad \times z^{(l-1)}(i+m, j+n; u) + b^{(l)}(t)\Bigg) \end{aligned} \quad (19)$$

where $b^{(l)}(t)$ is a bias term for the $v^{\text{th}}$ feature map in the $l$ layer, $u$ denotes the $u^{\text{th}}$ feature map in the $(l-1)^{\text{th}}$ layer, $w$ is the corresponding kernel function, and $z^{(l-1)}$ is the output from the upstream layer. One can clearly see from Eq. 19 that the convolution algorithm is actually implemented by cross-correlation in CNN, in contrast to what we are familiar with in terms of Fourier optics[121]. However, this does not change the resulting feature maps except their indices. We adapt this custom and call both Eq. 18 and Eq. 19 convolution.

The numerical calculation of the convolution in Eq. 19 requires moving the filter across spatial dimensions of the input data $z^{(l-1)}$. In conventional digital image processing and numerical implementation of convolution in optics[122], the filter is moved one pixel to the right and one pixel to

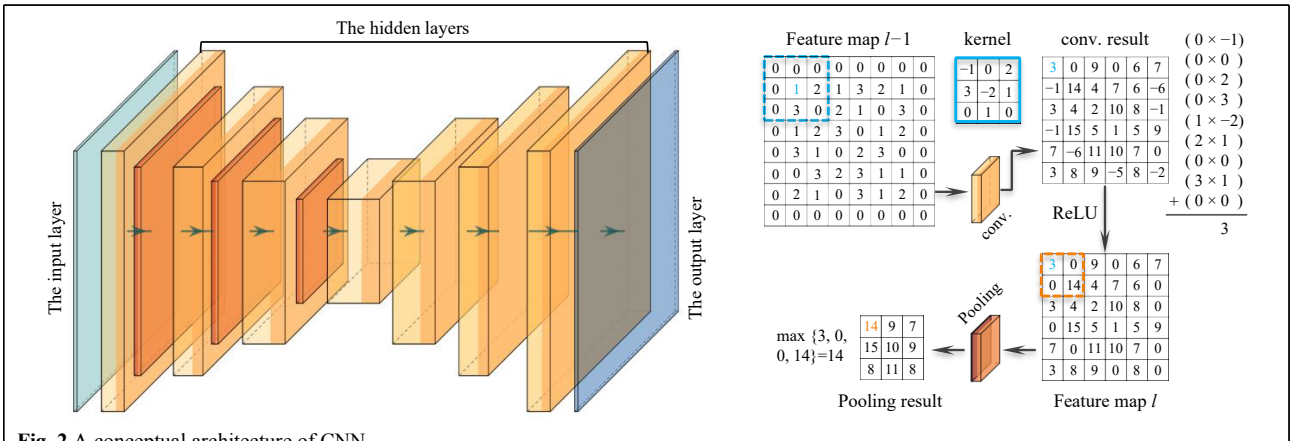


**Fig. 2** A conceptual architecture of CNN.

the bottom at a time. In the language of deep learning, this means that the stride is equal to 1. But it is not necessary to be like this in CNN. Indeed, the stride of 2 is commonly used. More specifically, for an input image of size $N \times N$ and a kernel of size $M \times M$ with stride $= k$, the resulting output will be of size $[(N-M)/k+1] \times [(N-M)/k+1]$.

The output feature maps $a^{(l)}(i, j; t)$ are then passed through an activation function, usually the ReLU function defined by Eq. 2, to a pooling layer, which performs nonlinear down-sampling. This can be done in many ways. But the one that is most commonly used and has good performance is maxpooling[123], which partitions each incoming feature map into a set of non-overlapping rectangle regions by using a filter with the size of $\kappa \times \kappa$ and outputs the maximum value of each region. Thus, the spatial size of the resulting feature map is reduced by a factor of $\kappa$. As a consequence, the number of parameters, memory footprint and amount of computation in the network can be reduced accordingly. The reduction of network parameters will of course improve the situation of overfitting. Maxpooling also guarantees that the most significant features and their rough location relative to the other features can be passed to the downstream layer.

In a typical CNN model, the convolutional layer, the ReLU layer and the pooling layer are arranged in sequence, forming the basic building block[42]. Usually several "convolution-ReLU-pooling " blocks are arranged in cascade, each of which performs the same set of operations as described above. In the end, only the most significant features (activated features) of the input data can be retained after the data stream passes through several blocks. In the applications of image recognition[100] and focused distance determination in holographic reconstruction[62], a flatten layer is usually used to reshape the three-dimensional feature volume into a one-dimensional vector, which is then sent to fully-connected layer described by Eq. 1 for further analysis. In the applications of holographic reconstruction[51−53] and aberration compensation[58], however, one wishes to reconstruct the object function and need a path to transform the activated feature maps back to the image pixels. This can be implemented by adding a deconvolutional network[124,125], which consists of a series of unpooling (reverse maxpooling), rectifying, and transpose convolution operations that upsample the feature maps many times until it reaches the size of the input hologram. The elegant U-Net[126] operates in a similar way, except that the unpooling layers are replaced by "up-convolution " layers.

A CNN model can be trained in the way as we described in Sec.. It involves the calculation of the gradient of the loss function with respect to the weight of every kernel function, which is then used to update the weight, usually according to the Adam method[113]. The back propagation model is a little different. One can refer to Ref. 42 for more details.

When the network goes deeper, it becomes very difficult to train because of the problems such as gradient vanishing and exploding[127]. He et al. proposed the residual neural network, or ResNet for short, to address this problem[101]. The most distinguished feature of ResNet is that two distanced layers can be connected directly through the short-cut. That says, the signal goes through a series of "convolution-ReLU-convolution " blocks instead of the "convolution-ReLU-maxpooling " in CNN. The result is then added to the input of this block. Thus, the forward propagation model can be formulated as[101]

$$\boldsymbol{z}^{(l)} = f^{(l)}\left(f^{(l-1)}\left(\boldsymbol{z}^{(l-2)}; \boldsymbol{W}^{(l-2,l-1)}, \boldsymbol{b}^{(l-1)}\right);\ \boldsymbol{W}^{(l-1,l)}, \boldsymbol{b}^{(l)}\right) + \boldsymbol{z}^{(l-2)} \tag{20}$$

where $f^{(l)}$ is the CNN forward propagation function defined in Eq. 19, $\boldsymbol{W}^{(l-1,l)}$ is the weight that connects layer $l-1$ and layer $l$. It is clearly seen that the input of layer $l-2$, $\boldsymbol{z}^{(l-2)}$, is directly connected to the $l^{\text{th}}$ layer. As it does not need to undergo the nonlinear transform, the gradient will flow easily during back propagation.

Nowadays, a common and practical strategy to design a DNN to solve holographic problems is to take U-Net as a backbone, and incorporate into it the ResNet ingredient of short-cuts[53,55,58,71,128−130]. The U-Net architecture can also be extended to allow the extraction of features of different size by introducing multiple channels in the down-sampling convolutional blocks[51,52,59,131−133]. Another interesting extension of U-Net is the so-called U-Net++[134], which has a pyramid shape architecture. It essentially consists of an encoder and a decoder that are connected through a series of nested, dense skip pathways, bridging the semantic gap between the feature maps of the encoder and the decoder prior to fusion[135].

### The Generative adversarial networks

Generative adversarial networks (GAN's) are NN's that learn to generate synthetic instances of data with the same statistical characteristics as the training data[136]. GAN is able to keep a parameter count significantly smaller than other methods with respect to the amount of data used to train the network. It the field of holography, GAN has been used for wavefront reconstruction[67,137,138], enhancement[139] and image classification[140]. It can be trained on paired data[139], unpaired data[67,138] or even unsupervisely[140] in some cases.

Architecturally, GAN is constituted of two neural networks, one of which is called the generator, and the

other, the discriminator. As shown in Fig. 3, the two networks are pitted one against the other (and thus "adversarial") in order to generate new, synthetic instances of data that can pass for real data. Explicitly, the generator G is a deconvolutional neural network[124,125] that generates new images as real as possible from a given noise variable input $z$, whereas the discriminator D is a CNN-based classifier that estimates the probability of a generated image and determines if it looks like a real image from the training set or not.

To proceed, let us denote the probability distribution of the input variable $z$ as $p_z$, that of the generator over data $x$ as $p_g$, and that of the discriminator over real sample $x$ as $p_r$. The purpose of GAN is to make sure that the discriminator's decisions over the real data are accurate by maximizing $\mathbb{E}_{x\sim p_r(x)}[\log D(x)]$, while the discriminator outputs a probability $D(G(z))$ that is close to zero by maximizing $\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$ for a given generative data instance $G(z)$, where $z\sim p_z(z)$.

Thus, one can see that $D$ and $G$ are actually playing a minimax game so that the objective is to optimize the following loss function[136]

$$\begin{aligned}\min_G\max_D \mathcal{L}(D,G) =&\mathbb{E}_{x\sim p_r(x)}[\log D(x)]\\ &+\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]\\ =&\mathbb{E}_{x\sim p_r(x)}[\log D(x)]\\ &+\mathbb{E}_{x\sim p_g(x)}[\log(1-D(x)]\end{aligned} \tag{21}$$

where

$$\mathcal{L}(G,D)=\int_x p_r(x)\log[D(x)]+p_g(x)\log[1-D(x)]\mathrm{d}x \tag{22}$$

Note that for any $(a,b)\in\mathbb{R}_2\backslash\{0,0\}$, the function $y\to a\log(y)+b\log(1-y)$ achieves its maximum in $[0,1]$ at $y=a/(a+b)$. It is then straightforward to obtain the best value of the discriminator[136]

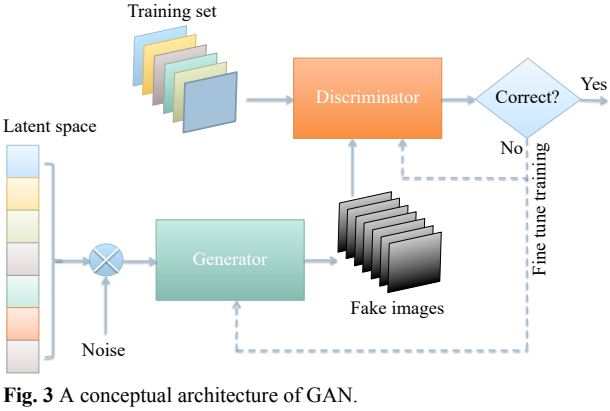


**Fig. 3** A conceptual architecture of GAN.

$$D^*(x)=\tilde{x}^*=\frac{p_r(x)}{p_r(x)+p_g(x)}\in[0,1] \tag{23}$$

Once the generator is trained to its optimal, $p_g=p_r$. Thus $D^*(x)=1/2$, and the loss function $\mathcal{L}(G,D^*)=-2\log 2$.

GAN can be trained by using SGD-like algorithm such as Adam[113] as in the case of CNN. But the discriminator and the generator should be trained against a static adversary[141]. That is, one should hold the generator values constant while training the discriminator, and *vice versa*.

There are several adaptations of GAN, among which the cycle-GAN[142] and conditional GAN (cGAN)[143] have been adopted for holography. Different from GAN, the purpose of the generator in cycle-GAN is not to generate an image from noise, but to take a hologram as its input and extract the most significant features via a series of convolutional layers, and then build a reconstructed image of the same size as the input hologram from these transformed features using a series of transpose convolutional layers. The most distinguished idea behind cycle-GAN is the introduction of a cycle consistency loss

$$\begin{aligned}\mathcal{L}_{\text{cycle}}(G,F)=&\mathbb{E}_{z\sim p_r(z)}[\|F(G(z))-z\|]\\ &+\mathbb{E}_{x\sim p_r(x)}[\|F(G(x))-x\|]\end{aligned} \tag{24}$$

that imposes a constrain to the model. The generator $G$ generates an object image $x$ from a hologram $z$, and $F$ generates the hologram $z$ of $x$. Thus, the total loss function can be defined as

$$\mathcal{L}(G,F,D_x,D_z)=\mathcal{L}_{\text{GAN}_x}+\mathcal{L}_{\text{GAN}_z}+\mathcal{L}_{\text{cycle}} \tag{25}$$

where $D_x$ and $D_z$ are the discriminators of $x$ and $z$, respectively, and $\mathcal{L}_{\text{GAN}_x}(G,D_x,x,z)$ and $\mathcal{L}_{\text{GAN}_z}(G,D_z,x,z)$, defined by Eq. 22, are the conventional GAN loss of the objects $x$ and holograms $z$ in the training set. It is clearly seen that $x$ and $z$ show up independently in each term of $\mathcal{L}(G,F,D_x,D_z)$, meaning that they do not need to pair up in the training set.

## The generation of training data

The training of DNN usually requires a large set of data, the size of which is typically ranging from a few thousands to tens of thousands in a typical proof-of-concept demonstration. The amount of labeled data is far less than that is used for deep learning applications in other communities such as computer vision. For example, AlexNet[100] was trained on a set composed of 1.2 million images.

In the case of supervised training the data used for training should be labeled so that every input data $\boldsymbol{x}_k$ is paired up with a corresponding ground truth data $\boldsymbol{y}_k$. But it is not necessary to do so in some other cases[67,140] like

unsupervised training. Optical acquisition of these data usually takes the most time, and requires the optical instruments in use to be stable during the long period. Otherwise, the data pairs cannot be registered well enough to match each other[51]. However, since holography is extremely sensitive to environment vibration[10], it is unavoidable to capture such vibration in the holograms during the time of acquisition (usually tens of hours depending on the number of holograms required to train the DNN), resulting in the instability of the fringe patterns. However, we have shown that DNN can be well trained on these "noisy" data[52].

An alternative and more flexible way to generate the training data is to use a numerical simulator provided that the physical system that describes the data link from the source to the detector can be accurately modeled. For example, this strategy has been applied to phase unwrapping[72] as well as speckle removal[66,68], ghost imaging[131], STORM[144,145] and diffraction tomography[146].

The raw data are mainly taken from MNIST[147], Faces-LFW[148] and CelebAMask-HQ[149], which are publicly available. In most of the proof-of-concept experiments, a spatial light modulator (SLM) is used to display these images in order to form the holograms of them by using a standard holographic system. However, in most of the practical applications of holography, it is not the 2D hand-written digits, English letters[147], or 2D human faces[148,149] that are of interest. Thus, DNN trained on these data sets is difficult to be generalized[150] to cope with most of the objects in the real world.

Recently, Ulyanov et al. have shown that the structure of a generator network can capture a great deal of low-level image statistics prior to any learning[151]. This can be generalized to a more general DNN such as the U-Net by incorporating a physical model into it, resulting in an untrained neural network that does not require any data to train[55]. It can be used to reconstruct the holograms of realistic objects. Indeed, over the past year, untrained DNN has been applied by to solve problems of holographic reconstruction[55–57], phase unwrapping[73], phase microscopy[152], diffraction tomography[153] and imaging[154], and even ghost imaging[155]. I will discuss it in more detail later on.

## Physics-informed DNN

So far I have introduced several important DNN models that are widely used in optics and holography. Indeed, DNN has been shown to outperform conventional physics-based approaches. For example, DNN allows twin-image-free reconstruction from a single-shot in-line digital holography[52]. A major reason for the success is that DNN, given enough data, can learn feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features[156] even explicit formulation of a system's exact physical nature is impossible owing to its complexity[119,132].

However, it is also well-known that DNN has a black-box issue[157]: the information stored in DNN is represented by a set of weights and connections that provides no direct clues to how the task is performed or what the relationship is between inputs and outputs[158]. When it is used to solve real-world physical problems, DNN has met with limited success due to a number of reasons: First, DNN requires a large amount of labeled data for training, which is rarely available in real application settings[159]. As discussed above, for most of the leaning-based methods for optical imaging and holography, an SLM is required to display the ground-truths. Frequently, the publicly available dataset such as the MNIST[147] database is used for demonstration. But this is hard to generalize to real-world samples owing to the issue that DNN models can only capture relationships in the available training data[150]. Second, DNN models often produce physically inconsistent results[160] when violating fundamental constraints. Third, the output is unexplainable[161].

Thus, it is highly desirable to take the benefits of both DNN models and physics models, and develop physics-informed or physics-guided DNN[162–165]. Barbastathis and coworkers[45] have concluded three different ways to incorporate a physical model into DNN, namely, recurrent physics-informed DNN, cascaded physics-informed DNN, and single-pass physics-informed DNN. In contrast, Ba and coworkers have concluded four different ways[166]: physical fusion, residual physics, physical regularization, and embedded physics. One can see that both these two ways of classification are somewhat equivalent.

According to Ba et al.[166], physical fusion is the most straightforward way. It feeds directly the solution from a physics model as (part of) the input to a DNN model. Barbastathis and coworkers[45] term this method as single-pass physics-informed DNN. This strategy has been employed in the very first work on learning-based holographic reconstruction, in which Rivenson et al.[51] used a conventional diffraction-based algorithm[5] to reconstruct a blurred wavefront from a hologram, and then used a trained DNN to improve the quality. This method has also been used for other problems such as ghost imaging[117] and phase retrieval[167].

In contrast, residual physics is to add the physical solution to the DNN output so that the DNN model only needs to learn the mismatch between the model-based solution and the ground truth[168]. Physical regularization, on

the other hand, harnesses the regularization term from a set of physical constraints to penalize the network solutions. The regularization term can be appended as part of the loss function explicitly or through a reconstruction process from physics[160]. These two concepts are similar to the recurrent physics-informed DNN and cascaded physics-informed DNN discussed in[45].

More exciting is the strategy of embedded physics. As shown in Fig. 4, the central idea is to take the physical model inside the network optimization loop: the physical model takes care of the well-posed forward propagation while DNN, the ill-posed backward propagation, in each iteration[55–57,138,152–155]. The error between the forward calculated output and the measured data can be used to estimate a defined loss function, which is then used to update the weights based on an SGD-like algorithm.

Here I would also like to draw the attention of the readers to an emerging strategy, which I call network approximating physics. By the name, it is to approximate a physical model by using a DNN[78,169]. For example, Shi et al. proposed to approximate the Fresnel zone plates through successive application of a set of learned $3 \times 3$ convolution kernels[78] in order to build a DNN model that can approximate the Fresnel diffraction and occlusion.

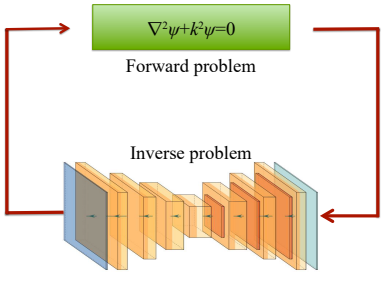


**Fig. 4** A typical architecture of an embedded physics DNN.

## DNN-inspired holography

After the brief introduction of deep learning neural networks in Sec. 1, now I will review some of the recent studies on the applications of deep learning in holography in this section. Before going into the detail, it is worthy of mentioning that the idea of using NN to for holography is not new. It has been proposed and demonstrated many years ago[170–174]. But the performance of neural networks was limited at that time because they were not deep enough due to the limited computation power. Indeed, one can find that some of the ideas demonstrated recently have been proposed at that time.

### Digital holographic reconstruction

A hologram can be formed by the superposition of the object beam $u_o(x,y)$ that carries the information of an object of interest and a reference beam $u_r(x,y)$, where $(x,y)$ is the spatial coordinates in the hologram plane

$$I(x,y) = |u_o(x,y) + u_r(x,y)|^2 = u_o^*(x,y)u_r(x,y) + |u_o(x,y)|^2 + |u_r(x,y)|^2 + u_o(x,y)u_r^*(x,y) \quad (26)$$

where the symbol * stands for phase conjugate.

### Conventional approaches

Intuitive approaches for holographic reconstruction are based on the physical model of diffraction, i.e., the numerical calculation of the diffraction process of the wave field[6]. In the off-axis geometry with a sufficient high carrier frequency all the three terms in Eq. 26 are well separated in the Fourier space, and therefore one can simply apply a spatial filter to remove the two unwanted terms. However, spatial filtering inevitably results in the loss of high-frequency components, which greatly hinder the reconstructed image quality[32]. In addition, one can use only a small part of the spatial bandwidth product (SBP) that the camera can offer[175–177] in this case. In in-line DH the reconstructed image are overlapped with the twin-image and the zeroth-order terms. Since the removal of the zeroth-order is comparatively straightforward, most of the studies on in-line holographic reconstruction is to deal with the twin image term.

Physics-based approach relies on some physical models as suggested by the name. Back in 1951, Bragg and Rogers[13] had realized that the twin image is actually the out-of-focus copy of the reconstructed object image, and it can be eliminated by the subtraction of the defocused wavefront from the other. But this method is technically tricky, and can be implemented only after the invention of DH[178,179]. The most widely used strategy nowadays is to tune some physical parameters of the optical system and acquire the corresponding holograms so as to set up a small linear equation system that relates the recorded holograms and the tuning parameters and solve for the object wavefront. For example, one can introduce multiple phase retardations stepwise in the reference beam and acquire the phase-shifted holograms[20–22], or move the camera along the propagation direction[23–25], or slightly tune the wavelength of the illumination laser beam[26]. However, as the control of these parameters is extremely difficult for very short wavelength radiation, these methods are infeasible for electron holography[180], X-ray holography[181], or $\gamma$–ray holography[182]. In this case, one should implement the phase shift by using an amplitude element such as the Chinese Taiji lens[183] or the Greek-ladder zone plate[184].

Mathematically, the twin image artifact arises due to the missing of the phase when the hologram is recorded[185]. This

suggests that the twin image artifact can be resolved if the missing phase of the hologram $|u_0(x,y)+u_r(x,y)|$ can be retrieved. This is the fundamental logic behind the phase-retrieval approach. Effectively, phase retrieval can be solved by using either a deterministic algorithm that is called now the transport-of-intensity equation (TIE)[186], or an iterative algorithm such as the Gerchberg-Saxton (GS)[27] and/or the Hybrid-Input-Output (HIO) algorithm[28]. This is in particular useful when the coherence of radiation source in used is poor (X-ray, for example). Thus the communities of X-ray holography and electron holography have made intensive studies since the late 1980s[185,187–190]. Along with the improvement of the technique and better modeling of the objective function, people now can achieve the reconstruction of the whole wavefront[29–31].

Phase retrieval is actually an inverse source problem of image reconstruction from magnitude[191,192]. It can be formulated as a more general class of inverse problems. The inverse problem approach treats the DH image reconstruction as a pure digital signal processing (DSP) problem, and solves it by using various numerical algorithms, such as statistical model[32,33], sparsity-enforcing prior[34], least squares[35], regularization[36,37], and compressive sensing[38–40]. A critical issue with it, from the computational point of view, is that the two-dimensional (2D) hologram must be rearranged as a one-dimensional (1D) vector in contrast to treating it as a 2D array in the two other aforementioned approaches. It thus requires the calculation of very large matrices, which is too heavy to do efficiently[193].

### Learning-based approaches

Several strategies have been proposed to solve the problem of holographic reconstruction. The most straightforward approach is the end-to-end DNN[52,54]. For example, Wang et al.[52] took the advantages of ResNet[101] and U-Net[126], and developed an alternative approach called eHoloNet for end-to-end holographic reconstruction. eHoloNet receives the raw digital hologram as the input, and produces the artifact-free object wavefront, which is a phase profile in their study, as the output. They treat the holographic reconstruction as solving an ill-posed inverse problem for the function $\mathcal{R}$ that maps directly the hologram space to the object space

$$\mathcal{R}_{\text{learn}} = \underset{\mathcal{R}_\theta,\theta\in\Theta}{\arg\min} \sum_{n=1}^{N} \mathcal{L}(u_{o,n}, \mathcal{R}_\theta\{I_n\}) + \Psi(\theta) \qquad (27)$$

where $\theta$ is an explicit setting of the network parameters $\Theta$, $\mathcal{L}(\cdot)$ is the loss function to measure the error between the $n^{\text{th}}$ phase object $u_{o,n}$, and the corresponding in-line hologram $\mathcal{R}_\theta\{I_n\}$, and $\Psi(\theta)$ is a regularizer on the parameters with the aim of avoiding overfitting[194]. They demonstrated their approach using 10,000 handcraft images from the MNIST dataset[147] and 12,651 images of the USAF resolution chart. All these images were resized to 768 × 768 pixels and displayed on a phase-only SLM (Holoeye, LETO), making them effectively phase objects. The in-line digital holograms of all the 22,651 phase objects were acquired by a Michelson interferometer. 9000 pairs of handcraft images and their holograms and 11,623 pairs of resolution charts and their holograms were used to train the eHoloNet, respectively. The lefts were used for test.

In order to reconstruct both the intensity and phase simultaneously from a single digital hologram, Wang et al. proposed a Y-shaped architecture[54]. The loss function then is defined as

$$\mathcal{L} = \lambda\mathcal{L}_I + \mathcal{L}_P \qquad (28)$$

where $\mathcal{L}_I$ and $\mathcal{L}_P$, defined according to Eq. 15, denote the loss function of the intensity and phase of the complex wavefront, and the weight $\lambda = 0.01$ in their experiments so as to enforce the significance of the phase.

The end-to-end approach can be implemented via GAN[67,137] as well. One advantage to use GAN is that the training data do not need to pair up.

The second approach is the physics fusion or single-pass physics-informed DNN[51,53]. As discussed in Sec., this is a two-step process. First, the complex wavefront was reconstructed by using the conventional numerical free space propagation back to the object plane. As aforementioned, the reconstructed wavefront is usually overlapped with the twin image, and the zeroth-order artifacts. The amplitude and phase of the reconstructed wavefront were then sent separately into a DNN, which has been trained to remove all these artifacts[51]. In their study, Rivenson et al. adopted a network architecture based on ResNet[101], as shown in Fig. 5a. The network was trained by the directly reconstructed amplitude and phase using numerical free space propagation algorithm and the corresponding ground truths (which are reconstructed by using phase retrieval algorithms from multiple holograms[195,196]). 100 image pairs were used to train the network. The results are shown in Fig. 5b. This method can be applied to off-axis DH to improve the quality of the reconstructed image as well[53].

The third one, physics-informed DNN is an exciting approach for holographic reconstruction. For example, Wang et al. have proposed a physics-enhanced DNN (PhysenNet)[55] that employs a strategy of incorporating a physical imaging model into a conventional DNN.

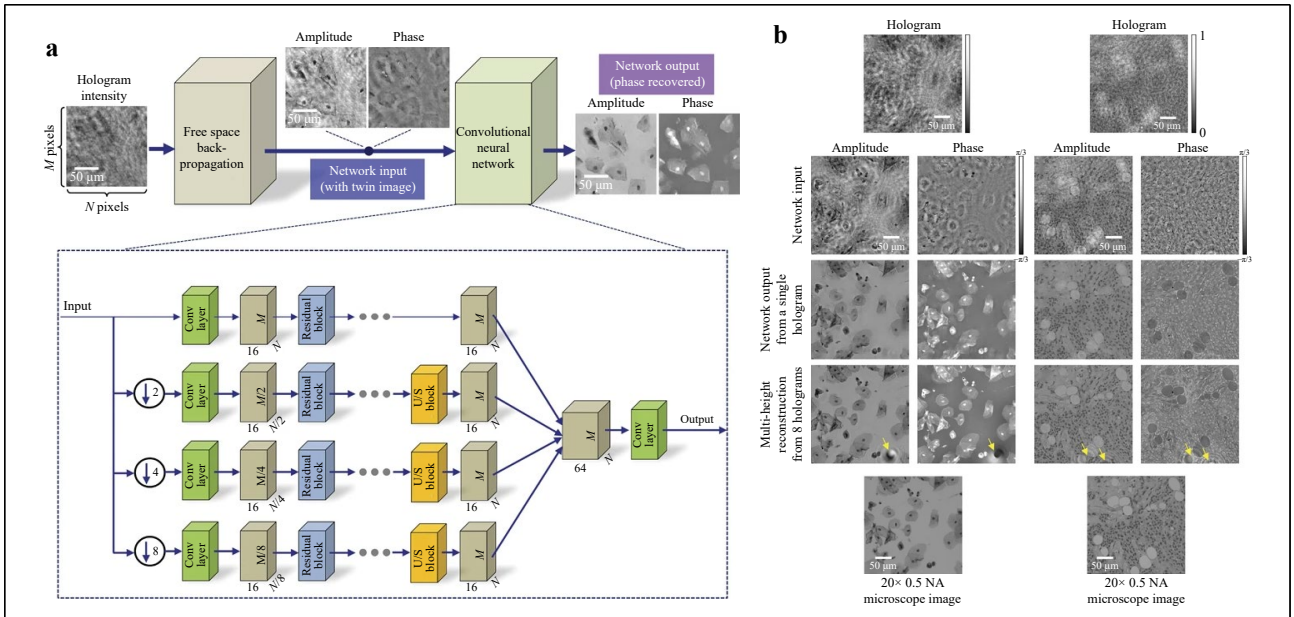


**Fig. 5 a** The DNN architecture and the pipeline for holographic reconstruction proposed by Rivenson et al.[51] and **b** the experimental results. (adapted from 51).

PhysenNet has two apparent advantages. First, it does not need any data to pre-train. This can be clearly seen in the objective function

$$\mathcal{R}_{\theta^*} = \arg\min_{\theta\in\Theta} \mathcal{L}(H(\mathcal{R}_\theta\{I\}, I) \tag{29}$$

where $I$ is the hologram or intensity pattern from which we wish to reconstruct the phase, and $H$ is the physical model, which is the Fresnel transform in their explicit case. It can be any other image formation process that can be accurately modeled[55–57,73,152–155]. Eq. 29 suggests that PhysenNet just requires the data to be process ($I$ in this case) as its input. The interplay between the physical model and the randomly initialized DNN provides a mechanism to optimize the network parameters, and produce a good reconstruction. Second, the reconstructed image satisfies the constraint imposed by the physical model so that it is interpretable[163]. The experimental results are plotted in Fig. 6.

The DNN model in PhysenNet can be replaced by other neural networks dependent on the task in hand. For example, Zhang et al. have demonstrated the incorporation of a phase imaging model into GAN[138].

## Phase unwrapping

Holographically reconstructed phases are usually wrapped owing to the $2\pi$-phase ambiguities and thus need unwrapping, which is also a typical ill-posed inverse problem. Conventional phase unwrapping techniques estimate the phase either by integrating through the confined path (referred to as path-dependent methods) or by minimizing the energy function between the wrapped phase and the approximated true phase (referred to as minimum-norm approaches)[72]. DNN provides a very feasible solution to this kind of problem because it can resolve the issues such as error accumulation, high computational time and noise sensitivity that conventional techniques frequently encounter.

Actually, the idea of using neural networks for phase unwrapping has been proposed by Takeda et al.[171] and Kreis et al.[172,173] in the 1990s. But with the developments of DNN techniques and computer power, much deeper neural networks are available now. There are ways to treat the phase unwrapping problem from the DNN point of view. A straightforward way is to take it as a regression problem, and develop a DNN to map a wrapped phase to an unwrapped phase. This can be done, for example, by using a U-Net trained on labeled data[71]. One research line is to improve the network design, aiming to enhance the phase quality. For example, Zhang et al.[198] have proposed a DNN model called DeepLabV3+, which can achieve noise suppression and strong feature representation capabilities. They demonstrated that it is out-performed the conventional path-dependent and minimum-norm algorithms. It is also possible to unwrap a phase by using an untrained DNN in a way similar to PhysenNet[55]. For example, Yang et al. have experimentally demonstrated that the proposed method faithfully recovers the phase of complex samples on both real and simulated data[73].

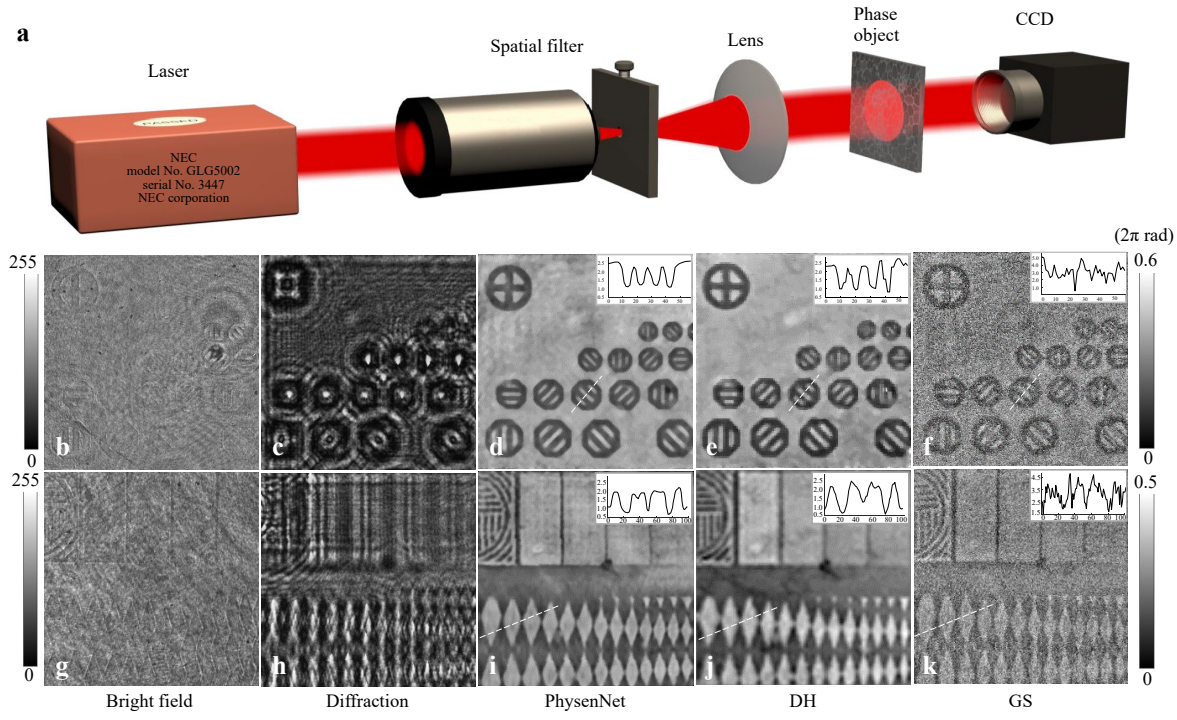


**Fig. 6** Experimental results. **a** Experimental set-up. **b** and **g** show two different parts of the phase object, **c** and **h** show the diffraction patterns, **d** and **i** show the phase images reconstructed by PhysenNet, **e** and **j** show the phase images reconstructed via off-axis digital holography, and **f** and **k** show the phase images reconstructed with the GS algorithm. (after 55).

Alternatively, one can treat phase unwrapping as a classification problem. For example, Zhang et al.[197] have demonstrated it by transferring phase unwrapping into a multi-class classification problem and introduced an efficient segmentation network to identify the classes. Their experimental results are plotted in Fig. 7.

Learning-based phase unwrapping algorithms have been applied to solve the problems in many different fields of studies, such as biology[199] and Fourier domain Doppler optical coherence tomography[200].

## Autofocusing

Autofocusing is about the automatic determination of the numerical calculation of the free space propagation distance of the wavefront from the hologram plane[201]. This is in particular important for the applications of DH in industrial and biological inspection[202]. Conventionally, the focused distance is determined by a criterion function with respect to the reconstruction distance. The criterion function can be defined in many ways, such as the entropy of the reconstructed image, the magnitude differential[201], and sparsity[203], and usually has a local maximum or minimum value at the focal plane.

Learning-based autofocusing algorithms employ different strategies. The prediction of the focusing distance is not made by searching a local extreme value of a criterion function, but by directly analyzing a digital hologram by using a deep neural network. One can think of autofocusing as a regression problem or a classification problem. The regression approach is to train the network by using a stack of artifact-free reconstructed images that are paired up with a hologram[59–61]. Each image in the stack is associated with a number that indicates the reconstruction distance. All these numbers are used to rectify the output layer during the training process. Taking the advantage of the U-Net[126] and ResNet[101], Wu et al[65] proposed the HIDEF (Holographic Imaging using Deep learning for Extended Focus) CNN. This allows the direct reconstruction at the correct distance when a hologram is inputted to the trained HIDEF CNN. Jaferzadeh et al. proposed a DNN model with a regression layer as the top layer to estimate the best reconstruction distance[204].

The classification approach was proposed by Ren et al[62] and Shimobaba et al[63]. For example Ren and coworkers experimentally recorded the 5000 holograms of several objects (a resolution chart, a testis slice, a ligneous dicotyledonous stems, an earthworm crosscut, etc.) at 10 different distances; and use the holograms and the associated distance values to train their neural network.

An alternative strategy is to take the focusing distance as

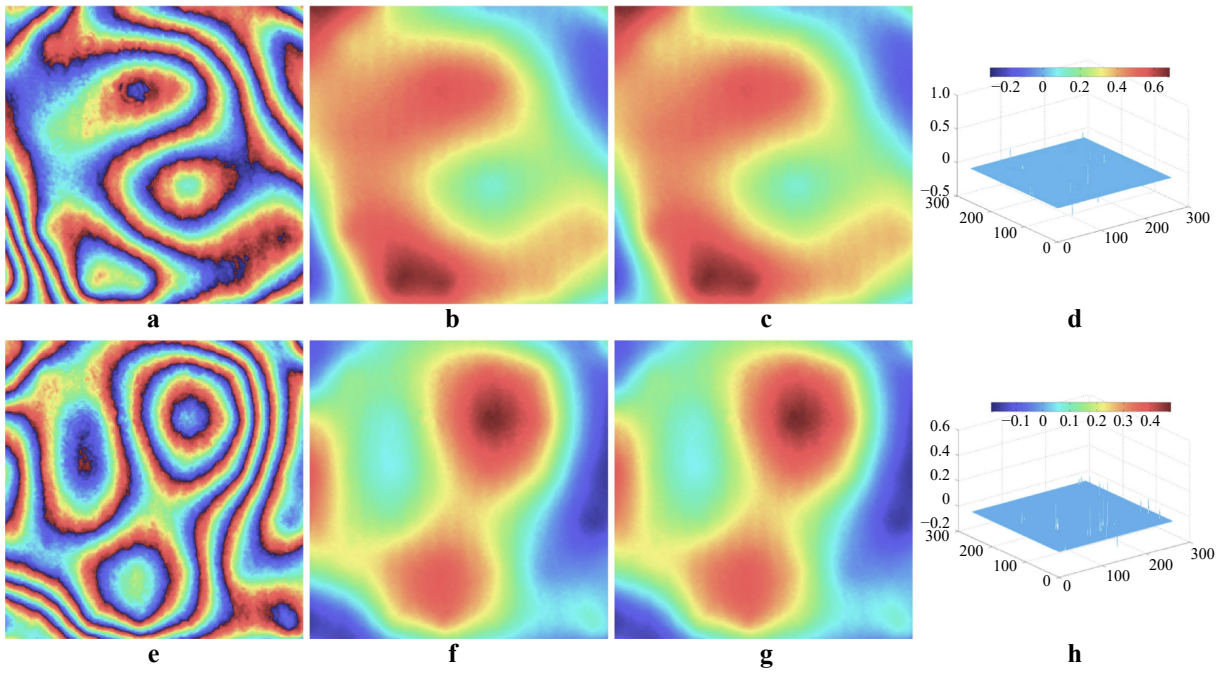


**Fig. 7** Unwrapping results on real data. From left to right are: wrapped phases [input, **a**, **e**], reconstructed unwrapped phases by the DNN [**b**, **f**] and MG [**c**, **g**], and differences [**d**, **h**]. (after 197).

an uncertain parameter, and ask the neural network to optimize automatically[135]. In this case, the objective function can be written as

$$[\mathcal{R}_{\theta^*}, d] = \arg\min_{\theta\in\Theta,\, d} \mathcal{L}(H[\mathcal{R}_\theta(I), d], I) \tag{30}$$

where the uncertain focusing distance $d$ enters the the physical model $H$ now, and will be optimized by the network.

The objective function in the form of Eq. 30 is similar to Eq. 29. This suggests that the only input required by the neural network is the hologram $I$, and the DNN does not need to be pre-trained on any dataset. As shown in Fig. 8, the algorithm will converge to the exact distance value along with proceeding of the iteration.

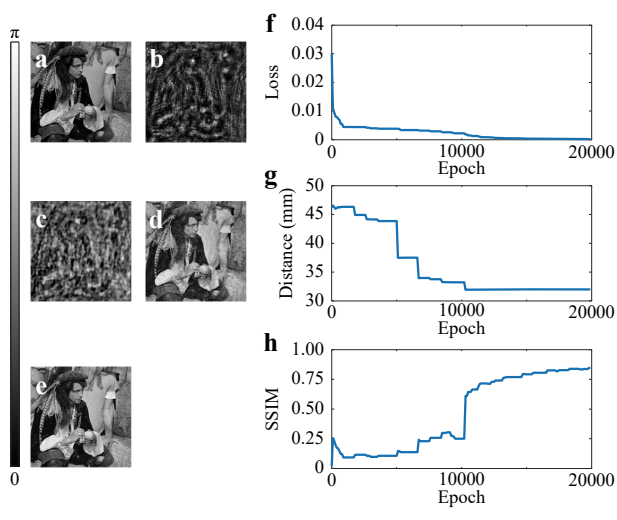


**Fig. 8** The reconstruction process of a phase object. **a** the original phase object. **b** the diffraction pattern. The retrieved phase at the epoch of **c** 900, **d** 10300, and **e** 19800. The behavior of **f** the loss function, **g** estimated distance, and **h** the SSIM value as a function of the number of epoch. (after 135).

### Phase Aberration Compensation

Learning-based approaches have also been used for phase aberration compensation in digital holographic microscopy[58,152,205]. Again, phase aberration compensation can be formulated as a classification[58] or a regression[152,205] problem. In the work by Nguyen et al.[58], the role DNN plays is to segment the reconstructed and unwrapped phase. The phase aberration then can be determined by Zernike polynomial fitting, and its conjugate can be numerically calculated to compensate the aberration. Nguyen et al. experimentally took the holograms of 306 breast cancer cells as the input and the corresponding manually segmented maps as the output to train their neural network, which is also a U-Net + ResNet architecture in this case. They used the softmax function defined in Eq. 6 in the last layer in their neural network to calculate the prediction probability of background/cell potential, and the cross-entropy loss defined in Eq. 17 for

back propagation. The experimental results are plotted in Fig. 9.

In contrast, the regression approach proposed by Xiao et al.[205] endeavors to optimize the coefficients for constructing the phase aberration map that act as responses corresponding to the input aberrated phase image. Embedded physics DNN can be used for this problem as well. Encapsulating the image prior and the system physics, Bostan et al.[152] have proposed an untrained DNN that can simultaneously reconstruct the phase and pupil-plane aberrations by fitting the weights of the network to the captured images.

### Suppression of Speckle

As a coherent imaging modality, DH reconstruction is also influenced by the coherence of the illumination laser source[11,206], which naturally results in speckle[207]. The elimination of speckle noise has been one of the main issues in DH. Conventionally, this can be done either optically or digitally. Optical methods usually require multiple measurements under different conditions. Digital methods can work on a single hologram but the reduction of speckle results in the loss of information as well. Bianco et al have given a very nice review of the most important speckle removal techniques[208].

Recently, Jeon et al.[66] have demonstrated that, by using DNN, it is possible to remove the speckle without any degradation of the image quality. The network architecture they used is again the combination of U-Net and ResNet. For supervised learning, one needs to pair up the speckled images and the corresponding speckle-free ones in order to train the network. But speckle-free images are unlikely to be obtainable from experimentally acquired holograms. So they used numerically generated speckled images from speckle-free images according to the model $\boldsymbol{y} = \mathcal{R}(\boldsymbol{x}) + \mathcal{N}(0, \varsigma^2)$, where $\mathcal{R}(\varsigma)$ is the Rayleigh distribution with scale parameter $\varsigma$, and $\mathcal{N}(\mu, \varsigma^2)$ is the Gaussian distribution with the mean $\mu$ and standard deviation $\varsigma$, to train their network, and test it with experimentally acquired holograms. Similar DNN can be applied to remove the speckle noise in phase image from holographic interferometry[68,69]. The strict requirement of labeled data can be released by using a more suitable network architecture such as Noise2Noise[209]. For example, Yin et al.[210] have demonstrated a speckle removal DNN without using clean data.

### Computer-generated holography

CGH has been recognized as the most promising true-3D display technology since it can account for all human visual cues such as stereopsis and eye focusing[8,9,88,89] as well as a powerful tool for the test of optical elements[84–86]. In particular, for the application in holographic display, it requires the generated holograms to be reasonably large in size. But the calculation of such holograms within acceptable time has been one of the main challenges in this field[211]. Although iterative phase-retrieval algorithms[27,28,87,212] have been intensively employed for this task, modern approaches for CGH calculation are non-iterative[88,89,211,213], in combination of acceleration techniques such as look-up table[90,214] and the use of GPU[215].

The use of DNN has dramatically accelerated the calculation of CGH[74–76,78–81]. People have used U-Net-based architecture to generate phase-only holograms[74] and binary holograms[216], Y-shaped architecture to generate multi-depth

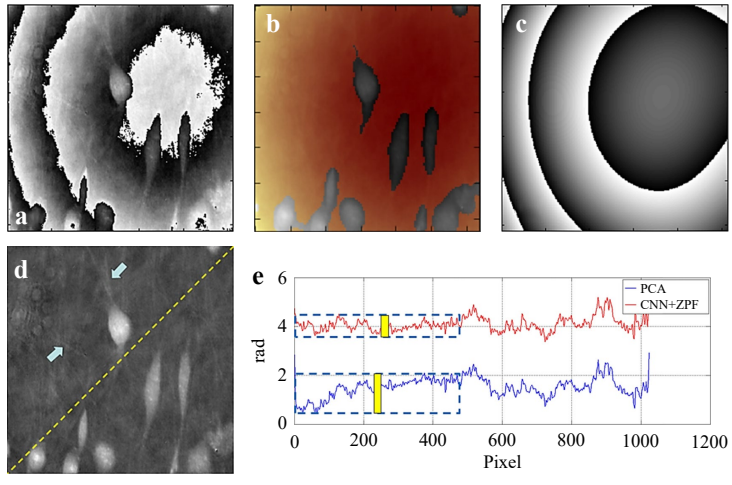


**Fig. 9 a** Phase aberration, **b** unwrapped phase overlaid with CNN's image segmentation mask, where background (color denoted) is fed into ZPF, **c** conjugated residual phase using CNN + ZPF, **d** fibers are visible after aberration compensation and are indicated by blue arrows, and **e** phase profile along the dash line in **d**. Yellow bars denote the flatness of region of interest. (after 58).

holograms[76], and autoencoder-based DNN for the fast generation of high-resolution holograms[80,81]. DNN has also been used to improve the quality of holographic display[217], and it allows to train in the loop[77]. Eybposh et al. have demonstrated an unsupervised learning based on GAN to achieve a fast hologram computation[75], although there is argument that this indirect training strategy may not obtain an optimal hologram[81]. The superb DNN-based algorithms allow the design of CGH to generate not just scalar but even arbitrary 3D vectorial fields in an instant and accurate manner[218].

A DNN-based CGH synthesis technique called tensor holography for true 3D holographic display has been proposed recently by Shi et al.[78]. Tensor holography is a physics-informed DNN technique. It imposes underlying physics (Fresnel diffraction) to train a CNN as an efficient proxy for both. Tensor holography was trained on MIT-CGH-4K Fresnel holograms dataset, consisting of 4000 pairs of RGB-depth (RGB-D) images and the corresponding 3D holograms that take the occlusion effect into account. Thus their DNN takes the 4-channel RGB-D image as its input, and predicts a color hologram as a 6-channel image (RGB amplitude and RGB phase), which can be used to drive three optically combined SLMs or one SLM in a time-multiplexed manner to achieve full-color holographic display.

## Holography-inspired DNN

Holography has been one of the important avenues to implement optical neural networks (ONN). Early researches include the optical implementations of fully-connected neural networks[93–97] and the Hopfield model[91,92], which is the base of the recurrent neural network (RNN). Rather than using logical neurons as in the digital counterpart, holographic neural networks rely on interconnection[219] that a hologram is inherently capable of[96,97]. In a fully-connected holographic neural network, the weights are stored in the pixels (neurons) of the holograms. Each neuron in a layer (hologram) performs a simple modulation of the light that impinges onto it from an upstream layer and subsequently illuminates a downstream layer. A holographic neural networks can be created in a photorefractive crystal[93,94], which is inherently a 3D device that has a potential to store billions of weights. Thus it is in principle promising to solve large-scale inverse problems. However, interest in further developments has been on the wane owing to some unfortunate reasons[220].

Modern implementation of holographic neural networks takes the advantage of diffraction[98], and thus named diffraction deep neural networks ($D^2NN$). In the hardware implementation, the holograms in $D^2NN$ are actually diffractive optical elements (DOE), which can be fabricated using 3D printing (see Fig. 10)[98], nanomanufacturing[221], or electric addressable digital micro-mirror devices (DMD)[222]. The weights are stored in the pixels as in conventional holographic neural networks[93–97]. There are good reasons to implement $D^2NN$ on DMD and SLM: it allows the creation of parallel and reconfigurable network connections as well as the addition of nonlinearity to each layer via fast square-law detection. Such cascaded nonlinear operations strongly amplify the dimensionality of data representation so that they can leverage for challenging computations[222].

As many other optical neural networks[220], the original $D^2NN$ engine was trained off-line[98]. But efforts have been made to implement better training strategy. For example, Zhou et al. have demonstrated an *in situ* back propagation training[223], Xiao et al. have proposed a back propagation technique that updates the unitary weights through the gradient translation from Euclidean to Riemannian space[224], and a method to implement optical dropout[225].

People have also investigated the applications of $D^2NN$ for logic operations[226,227], optical information processing[228,229], holographic reconstruction[230], pulse shaping[231], spectrally encoded single-pixel machine vision[232].

## Conclusion and perspective remarks

To conclude, I have reviewed the recent progresses on the field of *deep holography*, describing how holography and deep neural networks can benefit from each other. I would like to emphasize that this is a rapidly developing field. New and exciting results are published every few days. It is impossible to cover all the works within a single literature review article. It is also challenging to catch up with all the progresses. But I do identify a few lines of trend for further studies. For DNN-inspired holography, instead of trying different DNN architectures, an important trend is to incorporate a physical model into a DNN model[162–166]. Indeed, this idea has been received intensive attentions from the researchers in diverse fields[160–165] and will be a future direction. I have discussed five different ways of incorporating a physical model and showed their applications in solving various problems with respect to holography. Of particular interest is PhysenNet as it does not require any data to train in advance and the network prediction is satisfied with the constraint imposed by the physical model. However, the optimization is slow in comparison to conventional data-driven DNN[55,73,152] and thus lightweight network architectures and more efficient training algorithms are highly demanded. Perhaps it is also possible to take the advantages of both the physics-

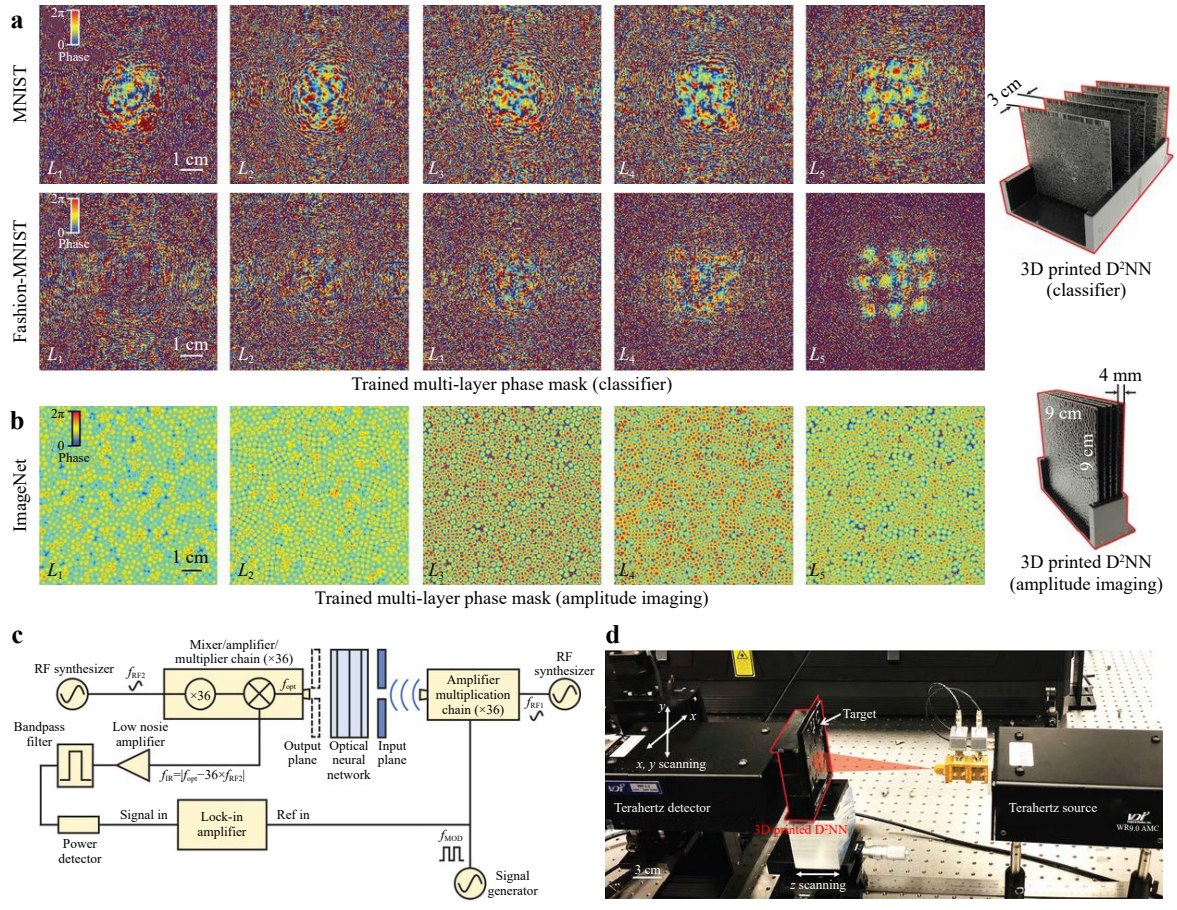


**Fig. 10** Experimental testing of **a** 3D-printed D²NN. The optimized phase maps of five different layers of the MNIST classifier, fashion product classifier, and **b** the amplitude imager. Schematic **c** and photo **d** of the experimental terahertz setup. (after 98).

informed and conventional data-driven methods[233], so that the training can be more efficient and the generalization, significantly improved.

For holography-inspired DNN, most of the studies published so far focus on optical inference. The capability of light-speed processing in parallel of holographic neural networks indeed guarantees tremendous inference power, even outperforming Nvidia's top of the line Tesla V100 tensor core GPU in certain task[222]. But the advantage of optics cannot be fully utilized if on-line training of the network cannot be perform optically. Initial efforts have been making along this line[222–225]. But there are many possibilities out there to efficiently implement most of the fundamental functions in a DNN. Furthermore, the performance of D²NN is explicitly determined by the pixel number, pixel size, and frame rate of SLM as these factors are related to the scale and reconfigurability of matrix computation and the capability of interconnection. Current liquid crystal devices are too slow; DMD is fast but its pixel pitch is too large. One promising way to get around relies on the advancing of novel optical materials and devices that can modulate the wave in the sub-wavelength scale[234] in high speed.

From a higher level point of view, all the DNN methods I have discussed herein belong to a wide class of algorithms called neural computing, which itself is also fast evolving. Indeed, one can have an impression of the evolution of DNN from feedforward NN to CNN and U-Net, and so on. Neural computing provides us more powerful algorithms such as the spiking neurosynaptic networks (SNN)[235] that can model the behavior and learning potential of the brain. The applicability and potential of these new algorithms in holography are a still open question.

Nevertheless, I hope I have convinced you that the field of deep holography as a whole is rich and exciting. It is a cross-disciplinary that requires holography and neural computing and many others. The mergence of them stimulates the development of each other, and gives us a fantastic field to explore.

**Acknowledgements**
This work was supported by the National Natural Science Foundation of China (62061136005, 61991452), the Sino-German Center (GZ1391), and the Key Research Program of Frontier Sciences of the Chinese Academy of Sciences (QYZDB-SSW-JSC002).

**Author contributions**
G.S. wrote the paper.

**Conflict of interest**
The author declares that he has no conflict of interest.

Received: 01 September 2021 Revised: 06 February 2022 Accepted: 15 February 2022
Accepted article preview online: 16 March 2022
Published online: 30 March 2022

## References

1. Gabor, D. A new microscopic principle. *Nature* **161**, 777-778 (1948).
2. Leith, E. N. & Upatnieks, J. New techniques in wavefront reconstruction. *Journal of the Optical Society of America* **51**, 1469-1473 (1961).
3. Leith, E. N. & Upatnieks, Reconstructed wavefronts and communication theory. *Journal of the Optical Society of America* **52**, 1123–1130 (1962).
4. Denisyuk, Y. N. On the reflection of optical properties of an object in a wave field of light scattered by it. *Dokl. Akad. Nauk SSSR* **144**, 1275-1278 (1962).
5. Schnars, U. & Jüptner, W. Digital Holography: Digital Hologram Recording, Numerical Reconstruction, and Related Techniques. (Berlin: Springer, 2005).
6. Kim, M. K. Digital holographic microscopy. in Digital Holographic Microscopy (ed Kim, M. K.) (New York: Springer, 2011).
7. Asundi, A. Digital Holography for MEMS and Microsystem Metrology. (Hoboken: Wiley, 2011).
8. Benton, S. A. & Bove, V. M. Jr. Holographic Imaging. (Hoboken: John Wiley & Sons, 2008).
9. Matsushima, K. Introduction to Computer Holography: Creating Computer-Generated Holograms as the Ultimate 3D Image. (Cham: Springer, 2020).
10. Caulfield, H. J. Handbook of Optical Holography. (New York: Academic Press, 1979).
11. Claus, D., Iliescu, D. & Rodenburg, J. M. Coherence requirement in digital holography. *Applied Optics* **52**, A326-A335 (2013).
12. Gabor, D. Holography, 1948-1971. *Proceedings of the IEEE* **60**, 655–668 (1972).
13. Bragg, W. L. & Rogers, G. L. Elimination of the unwanted image in diffraction microscopy. *Nature* **167**, 190-191 (1951).
14. Goodman, J. W. & Lawrence, R. W. Digital image formation from electronically detected holograms. *Applied Physics Letters* **11**, 77-79 (1967).
15. Schnars, U. & Jüptner, W. Direct recording of holograms by a CCD target and numerical reconstruction. *Applied Optics* **33**, 179-181 (1994).
16. Pedrini, G. & Tiziani, H. J. Short-coherence digital microscopy by use of a lensless holographic imaging system. *Applied Optics* **41**, 4489-4496 (2002).
17. Zhang, Y. Z. et al. Application of short-coherence lensless Fourier-transform digital holography in imaging through diffusive medium. *Optics Communications* **286**, 56-59 (2013).
18. Rosen, J. & Brooker, G. Fresnel incoherent correlation holography (FINCH): a review of research. *Advanced Optical Technologies* **1**, 151-169 (2012).
19. Cuche, E., Marquet, P. & Depeursinge, C. Spatial filtering for zero-order and twin-image elimination in digital off-axis holography. *Applied Optics* **39**, 4070-4075 (2000).
20. Yamaguchi, I. & Zhang, T. Phase-shifting digital holography. *Optics Letters* **22**, 1268-1270 (1997).
21. Meng, X. F. et al. Two-step phase-shifting interferometry and its application in image encryption. *Optics Letters* **31**, 1414-1416 (2006).
22. Liu, J. P. & Poon, T. C. Two-step-only quadrature phase-shifting digital holography. *Optics Letters* **34**, 250-252 (2009).
23. Zhang, Y. et al. Reconstruction of in-line digital holograms from two intensity measurements. *Optics Letters* **29**, 1787-1789 (2004).
24. Situ, G. et al. Generalized in-line digital holographic technique based on intensity measurements at two different planes. *Applied Optics* **47**, 711-717 (2008).
25. Das, B. & Yelleswarapu, C. S. Dual plane in-line digital holographic microscopy. *Optics Letters* **35**, 3426-3428 (2010).
26. Ryle, J. P., L i, D. & Sheridan, J. T. Dual wavelength digital holographic laplacian reconstruction. *Optics Letters* **35**, 3018-3020 (2010).
27. Gerchberg, R. W. & Saxton, W. O. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik* **35**, 237-246 (1972).
28. Fienup, J. R. Phase retrieval algorithms: a comparison. *Applied Optics* **21**, 2758-2769 (1982).
29. Latychevskaia, T. & Fink, H. W. Solution to the twin image problem in holography. *Physical Review Letters* **98**, 233901 (2007).
30. Latychevskaia, T. & Fink, H. W. Simultaneous reconstruction of phase and amplitude contrast from a single holographic record. *Optics Express* **17**, 10697-10705 (2009).
31. Rong, L. et al. Iterative solution to twin image problem in in-line digital holography. *Optics and Lasers in Engineering* **51**, 553-339 (2013).
32. Sotthivirat, S. & Fessler, J. A. Penalized-likelihood image reconstruction for digital holography. *Journal of the Optical Society of America A* **21**, 737-750 (2004).
33. Cetin, M., Karl, W. C. & Willsky, A. S. Edge-preserving image reconstruction for coherent imaging applications. *Proceedings of the International Conference on Image Processing. Rochester*, NY, USA: IEEE, 2002.
34. Denis, L. et al. Inline hologram reconstruction with sparsity constraints. *Optics Letters* **34**, 3475-3477 (2009).
35. Kamau, E. N. et al. Least-squares based inverse reconstruction of in-line digital holograms. *Journal of Optics* **15**, 075716 (2013).
36. Bourquard, A. et al. A practical inverse-problem approach to digital holographic reconstruction. *Optics Express* **21**, 3417-3433 (2013).
37. Schretter, C. et al. Regularized non-convex image reconstruction in digital holographic microscopy. *Optics Express* **25**, 16491-16508 (2017).
38. Brady, D. J. et al. Compressive holography. *Optics Express* **17**, 13040-13049 (2009).
39. Rivenson, Y., Stern, A. & Javidi, B. Overview of compressive sensing techniques applied in holography[Invited]. *Applied Optics* **52**, A423-A432 (2013).
40. Zhang, W. H. et al. Twin-image-free holography: a compressive sensing approach. *Physical Review Letters* **121**, 093902 (2018).
41. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436-444 (2015).
42. Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning. (Cambridge: MIT Press, 2016).
43. Goda, K., et al. AI boosts photonics and vice versa. *APL Photonics* **5**, 070401 (2020).

44. Jin, K. H. et al. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing* **26**, 4509-4522 (2017).
45. Barbastathis, G., Ozcan, A. & Situ, G. On the use of deep learning for computational imaging. *Optica* **6**, 921-943 (2019).
46. Jo, Y. et al. Quantitative phase imaging and artificial intelligence: a review. *IEEE Journal of Selected Topics in Quantum Electronics* **25**, 6800914 (2019).
47. Ma, W. et al. Deep learning for the design of photonic structures. *Nature Photonics* **15**, 77-90 (2021).
48. Wiecha, P. R. et al. Deep learning in nano-photonics: inverse design and beyond. *Photonics Research* **9**, B182-B200 (2021).
49. Rivenson, Y., W u, Y. C. & Ozcan, A. Deep learning in holography and coherent imaging. *Light: Science & Applications* **8**, 85 (2019).
50. Sheridan, J. T. et al. Roadmap on holography. *Journal of Optics* **22**, 123002 (2020).
51. Rivenson, Y. et al. Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light: Science & Applications* **7**, 17141 (2018).
52. Wang, H., L yu, M. & Situ, G. eHoloNet: a learning-based end-to-end approach for in-line digital holographic reconstruction. *Optics Express* **26**, 22603-22614 (2018).
53. Zhang, G. et al. Fast phase retrieval in off-axis digital holographic microscopy through deep learning. *Optics Express* **26**, 19388-19405 (2018).
54. Wang, K. Q. et al. Y-Net: a one-to-two deep learning framework for digital holographic reconstruction. *Optics Letters* **44**, 4765-4768 (2019).
55. Wang, F. et al. Phase imaging with an untrained neural network. *Light: Science & Applications* **9**, 77 (2020).
56. Lawrence, H. et al. Phase retrieval with holography and untrained priors: tackling the challenges of low-photon nanoscale imaging. arXiv: 2012.07386 (2021).
57. Niknam, F., Qazvini, H. & Latifi, H. Holographic optical field recovery using a regularized untrained deep decoder network. *Scientific Reports* **11**, 10903 (2021).
58. Nguyen, T. et al. Automatic phase aberration compensation for digital holographic microscopy based on deep learning background detection. *Optics Express* **25**, 15043-15057 (2017).
59. Pitkäaho, T., Manninen, A. & Naughton, T. J. Performance of autofocus capability of deep convolutional neural networks in digital holographic microscopy. Proceedings of the Digital Holography and Three-Dimensional Imaging 2017. JeJu Island Republic of Korea: Optical Society of America, 2017.
60. Pitkäaho, T., Manninen, A. & Naughton, T. J. Focus prediction in digital holographic microscopy using deep convolutional neural networks. *Applied Optics* 58-A208 (2019).
61. Ren, Z. B., Xu, Z. M. & Lam, E. Y. Autofocusing in digital holography using deep learning. Proceedings of SPIE 10499, Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XXV. San Francisco, California, United States: SPIE, 2018.
62. Ren, Z. B., X u, Z. M. & Lam, E. Y. Learning-based nonparametric autofocusing for digital holography. *Optica* **5**, 337-344 (2018).
63. Shimobaba, T., Kakue, T. & Ito, T. Convolutional neural network-based regression for depth prediction in digital holography. Proceedings of the 27th International Symposium on Industrial Electronics. Cairns, QLD, Australia: IEEE, 2018.
64. Huang, L. Z. et al. Holographic image reconstruction with phase recovery and autofocusing using recurrent neural networks. *ACS Photonics* **8**, 1763-1774 (2021).
65. Wu, Y. C. et al. Extended depth-of-field in holographic imaging using deep-learning-based autofocusing and phase recovery. *Optica* **5**, 704-710 (2018).
66. Jeon, W. et al. Speckle noise reduction for digital holographic images using multi-scale convolutional neural networks. *Optics Letters* **43**, 4240-4243 (2018).
67. Yin, D. et al. Digital holographic reconstruction based on deep learning framework with unpaired data. *IEEE Photonics Journal* **12**, 3900312 (2020).
68. Yan, K. T. et al. Fringe pattern de-noising based on deep learning. *Optics Communications* **437**, 148-152 (2019).
69. Tahon, M., Montresor, S. & Picart, P. Towards reduced CNNs for de-noising phase images corrupted with speckle noise. *Photonics* **8**, 255 (2021).
70. Ren, Z. B. et al. Fringe pattern improvement and super-resolution using deep learning in digital holography. *IEEE Transactions on Industrial Informatics* **15**, 6179-6186 (2019).
71. Wang, K. Q. et al. One-step robust deep learning phase unwrapping. *Optics Express* **27**, 15100-15115 (2019).
72. Spoorthi, G. E., Gorthi, R. K. S. S. & Gorthi, S. PhaseNet 2.0: phase unwrapping of noisy data based on deep learning approach. *IEEE Transactions on Image Processing* **29**, 4862-4872 (2020).
73. Yang, F. S. et al. Robust phase unwrapping via deep image prior for quantitative phase imaging. *IEEE Transactions on Image Processing* **30**, 7025-7037 (2021).
74. Horisaki, R., Takagi, R. & Tanida, J. Deep-learning-generated holography. *Applied Optics* **57**, 3859-3863 (2018).
75. Eybposh, M. H. et al. DeepCGH: 3D computer-generated holography using deep learning. *Optics Express* **28**, 26636-26650 (2020).
76. Lee, J. et al. Deep neural network for multi-depth hologram generation and its training strategy. *Optics Express* **28**, 27137-27154 (2020).
77. Peng, Y. F. et al. Neural holography with camera-in-the-loop training. *ACM Transactions on Graphics* **39**, 185 (2020).
78. Shi, L. et al. Towards real-time photorealistic 3D holography with deep neural networks. *Nature* **591**, 234-239 (2021).
79. Kang, J. W. et al. Deep-learning-based hologram generation using a generative model. *Applied Optics* **60**, 7391-7399 (2021).
80. Liu, S. C. & Chu, D. P. Deep learning for hologram generation. *Optics Express* **29**, 27373-27395 (2021).
81. Wu, J. C. et al. High-speed computer-generated holography using an autoencoder-based deep neural network. *Optics Letters* **46**, 2908-2911 (2021).
82. Lohmann, A. W. & Paris, D. P. Binary Fraunhofer holograms, generated by computer. *Applied Optics* **6**, 1739-1748 (1967).
83. Brown, B. R. & Lohmann, A. W. Computer-generated binary holograms. *IBM Journal of Research and Development* **13**, 160-168 (1969).
84. Birch, K. G. & Green, F. J. The application of computer-generated holograms to testing optical elements. *Journal of Physics D: Applied Physics* **5**, 1982-1992 (1972).
85. Osten, W., Baumbach, T. & Jüptner, W. Comparative digital holography. *Optics Letters* **27**, 1764-1766 (2002).
86. Pruss, C., et al. Computer-generated holograms in interferometric testing. *Optical Engineering* **43**, 2534-2540 (2004).
87. Wyrowski, F. Diffractive optical elements: iterative calculation of quantized, blazed phase structures. *Journal of the Optical Society of America A* **7**, 961-969 (1990).
88. Slinger, C., Cameron, C. & Stanley, M. Computer-generated holography as a generic display technology. *Computer* **38**, 46-53 (2005).
89. Bove, V. M. Jr. Display holography' s digital second act. *Proceedings of the IEEE* **100**, 918-928 (2012).
90. Lucente, M. Interactive computation of holograms using a look-up

table. *Journal of Electronic Imaging* **2**, 28-34 (1993).
91. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* **79**, 2554-2558 (1982).
92. White, H. J. & Wright, W. A. Holographic implementation of a hopfield model with discrete weightings. *Applied Optics* **27**, 331-338 (1988).
93. Psaltis, D. et al. Holography in artificial neural networks. *Nature* **343**, 325-330 (1990).
94. Hsu, K. Y., L i, H. Y. & Psaltis, D. Holographic implementation of a fully connected neural network. *Proceedings of the IEEE* **78**, 1637-1645 (1990).
95. Keller, P. E. & Gmitro, A. F. Design and analysis of fixed planar holographic interconnects for optical neural networks. *Applied Optics* **31**, 5517-5526 (1992).
96. Jutamulia, S. & Yu, F. T. S. Overview of hybrid optical neural networks. *Optics & Laser Technology* **28**, 59-72 (1996).
97. Kaikhah, K. & Loochan, F. Computer generated holograms for optical neural networks. *Applied Intelligence* **14**, 145-160 (2001).
98. Lin, X., et al. All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004-1008 (2018).
99. Yan, T., et al. Fourier-space diffractive deep neural network. *Physical Review Letters* **123**, 023901 (2019).
100. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe Nevada: Curran Associates Inc., 2012, 1097-1105.
101. He, K. M. et al. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016.
102. He, K. M. *et al.* Identity mappings in deep residual networks. Proceedings of the 14th European Conference on Computer Vision. Amsterdam, The Netherlands: Springer, 2016.
103. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**, 281-305 (2012).
104. Lee, S. et al. Background information of deep learning for structural engineering. *Archives of Computational Methods in Engineering* **25**, 121-129 (2018).
105. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* **4**, 251-257 (1991).
106. Patterson, J. & Gibson, A. Deep Learning: A Practitioner's Approach. (Sebastopol, CA: O'reilly, 2017).
107. Hansen, C. Activation functions explained -GELU, SELU, ELU, ReLU and more. at https://mlfromscratch.com/activation-functions-explained/.
108. Lei, N. et al. Geometric understanding of deep learning. arXiv: 1805.10451 (2018).
109. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **322**, 533-536 (1986).
110. Ionescu, C., Vantzos, O. & Sminchisescu, C. Matrix backpropagation for deep networks with structured layers. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile: IEEE, 2015, 2965-2973.
111. Rojas, R. Neural Networks: a Systematic Introduction. (Berlin, Heidelberg: Springer, 1996).
112. Yuan, Y. X. Step-sizes for the gradient method. *AMS/IP Studies in Advanced Mathematics* **42**, 785-796 (2008).
113. Kingma, D. & Ba, J. Adam: a method for stochastic optimization. arXiv: 1412: 6980 (2017).
114. Zhao, H., et al. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging* **3**, 47-57 (2017).
115. Menard, S. Coefficients of determination for multiple logistic regression analysis. *The American Statistician* **54**, 17-24 (2000).
116. Xue, Y. J., et al. Reliable deep-learning-based phase imaging with uncertainty quantification. *Optica* **6**, 618-629 (2019).
117. Lyu, M. et al. Deep-learning-based ghost imaging. *Scientific Reports* **7**, 17865 (2017).
118. Lyu, M. et al. Exploit imaging through opaque wall via deep learning. arXiv: 1708.07881 (2017).
119. Lyu, M. et al. Learning-based lensless imaging through optically thick scattering media. *Advanced Photonics* **1**, 036002 (2019).
120. Hubel, D. H. & Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology* **195**, 215-243 (1968).
121. Goodman, J. W. Introduction to Fourier Optics. (Englewood: Roberts & Company, 2004).
122. Voelz, D. G. Computational Fourier Optics: A MATLAB Tutorial. (Bellingham: SPIE, 2011).
123. Scherer, D., Müller, A. C. & Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. Proceedings of the 20th International Conference on Artificial Neural Networks. Thessaloniki, Greece: Springer, 2010, 92-101.
124. Zeiler, M. D. et al. Deconvolutional networks. Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, CA, USA: IEEE, 2010.
125. Mohan, R. Deep deconvolutional networks for scene parsing. arXiv: 1411.4101 (2014).
126. Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional networks for biomedical image segmentation. arXiv: 1505.04597 (2015).
127. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010.
128. Shimobaba, T. et al. Digital holographic particle volume reconstruction using a deep neural network. *Applied Optics* **58**, 1900-1906 (2019).
129. Zhang, Z. D. et al. Holo-UNet: hologram-to-hologram neural network restoration for high fidelity low light quantitative phase imaging of live cells. *Biomedical Optics Express* **11**, 5478-5487 (2020).
130. Chang, T. et al. Calibration-free quantitative phase imaging using data-driven aberration modeling. *Optics Express* **28**, 34835-34847 (2020).
131. Wang, F. et al. Learning from simulation: an end-to-end deep-learning approach for computational ghost imaging. *Optics Express* **27**, 25560-25572 (2019).
132. Zheng, S. S. et al. Incoherent imaging through highly nonstatic and optically thick turbid media based on neural network. *Photonics Research* **9**, B220-B228 (2021).
133. Zheng, S. S. et al. Non-line-of-sight imaging under white-light illumination: a two-step deep learning approach. *Optics Express* **29**, 40091-40105 (2021).
134. Zhou, Z. W. et al. Unet++: a nested U-net architecture for medical image segmentation. Proceedings of the Deep Learning in Medical Image Analysis - and - Multimodal Learning for Clinical Decision Support - 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018. Granada, Spain: Springer, 2018.
135. Zhang, X. Y., Wa ng, F. & Situ, G. BlindNet: an untrained learning approach toward computational imaging with model uncertainty. *Journal of Physics D: Applied Physics* **55**, 034001 (2022).
136. Goodfellow, I. J. *et al.* Generative adversarial networks. Proceedings of

the 27th International Conference on Neural Information Processing Systems. Montreal, Canada, 2014.
137. Moon, I. et al. Noise-free quantitative phase imaging in Gabor holography with conditional generative adversarial network. *Optics Express* **28**, 26284-26301 (2020).
138. Zhang, Y. H. et al. PhaseGAN: a deep-learning phase-retrieval approach for unpaired datasets. *Optics Express* **29**, 19593-19604 (2021).
139. Wu, Y. C. et al. Bright-field holography: cross-modality deep learning enables snapshot 3D imaging with bright-field contrast using a single hologram. *Light: Science & Applications* **8**, 25 (2019).
140. Mangal, J. et al. Unsupervised organization of cervical cells using bright-field and single-shot digital holographic microscopy. *Journal of Biophotonics* **12**, e201800409 (2019).
141. Salimans, T. et al. Improved techniques for training GANs. Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain: Curran Associates Inc., 2016.
142. Zhu, J. Y. et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of 2017 IEEE International Conference on Computer Vision. Venice, Italy: IEEE, 2017.
143. Isola, P. *et al.* Image-to-image translation with conditional adversarial networks. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017.
144. Nehme, E. et al. Deep-STORM: super-resolution single-molecule microscopy by deep learning. *Optica* **5**, 458-464 (2018).
145. Weigert, M. et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature Methods* **15**, 1090-1097 (2018).
146. Matlock, A. & Tian, L. Physical model simulator-trained neural network for computational 3D phase imaging of multiple-scattering samples. arXiv: 2103.15795 (2021).
147. LeCun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278-2324 (1998).
148. Huang, G. B. et al. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. (University of Massachusetts, 2007).
149. Lee, C. H. *et al.* MaskGAN: towards diverse and interactive facial image manipulation. Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA: IEEE, 2020.
150. Neyshabur, B. et al. Exploring generalization in deep learning. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, CA, USA: Curran Associates Inc., 2017.
151. Lempitsky, V., Vedaldi, A. & Ulyanov, D. Deep image prior. Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018.
152. Bostan, E. et al. Deep phase decoder: self-calibrating phase microscopy with an untrained deep neural network. *Optica* **7**, 559-562 (2020).
153. Zhou, K. C. & Horstmeyer, R. Diffraction tomography with a deep image prior. *Optics Express* **28**, 12872-12896 (2020).
154. Yang, D. Y. et al. Dynamic coherent diffractive imaging with a physics-driven untrained learning method. *Optics Express* **29**, 31426-31442 (2021).
155. Wang, F. et al. Far-field super-resolution ghost imaging with a deep neural network constraint. *Light: Science & Applications* **11**, 1 (2022).
156. Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* **2**, 1-127 (2009).
157. Sjöberg, J. et al. Nonlinear black-box modeling in system identification: a unified overview. *Automatica* **31**, 1691-1724 (1995).
158. Tzeng, F. Y. & Ma, K. L. Opening the black box -data driven visualization of neural networks. Proceedings of the VIS 05. IEEE Visualization, 2005. Minneapolis, MN, USA: IEEE, 2005.
159. Karpatne, A., et al. Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* **29**, 2318-2331 (2017).
160. Beucler, T. et al. Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters* **126**, 098302 (2021).
161. Buhrmester, V., Münch, D. & Arens, M. Analysis of explainers of black box deep neural networks for computer vision: a survey. *Machine Learning and Knowledge Extraction* **3**, 966-989 (2021).
162. Arrieta, A. B. et al. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* **58**, 82-115 (2020).
163. Willard, J. et al. Integrating scientific knowledge with machine learning for engineering and environmental systems. arXiv: 2003.04919v5 (2021).
164. Roscher, R. et al. Explainable machine learning for scientific insights and discoveries. *IEEE Access* **8**, 42200-42216 (2020).
165. Wijesinghe, P. & Dholakia, K. Emergent physics-informed design of deep learning for microscopy. *Journal of Physics: Photonics* **3**, 21003 (2021).
166. Ba, Y. H., Zhao, G. Y. & Kadambi, A. Blending diverse physical priors with neural networks. arXiv: 1910.00201 (2019).
167. Goy, A. et al. Low photon count phase retrieval using deep learning. *Physical Review Letters* **121**, 243902 (2018).
168. Zeng, T. J. & Lam, E. Y. Model-based network architecture for image reconstruction in lensless imaging. Proceedings of SPIE 11551, Holography, Diffractive Optics, and Applications X. SPIE, 2020.
169. Iten, R. et al. Discovering physical concepts with neural networks. *Physical Review Letters* **124**, 010508 (2020).
170. Takeda, M. & Goodman, J. W. Neural networks for computation: number representations and programming complexity. *Applied Optics* **25**, 3033-3046 (1986).
171. Takeda, M. Phase unwrapping by neural network. Proceedings of the FRINGE'93. Akademie Verlag, 1993, 136-141.
172. Kreis, T. M., Biedermann, R. & Jüptner, W. P. O. Evaluation of holographic interference patterns by artificial neural networks. Proceedings of SPIE 2544, Interferometry VII: Techniques and Analysis. San Diego, CA, United States: SPIE, 1995, 11-24.
173. Kreis, T., Jüptner, W. & Biedermann, R. Neural network approach to holographic nondestructive testing. *Applied Optics* **34**, 1407-1415 (1995).
174. Frauel, Y. & Javidi, B. Neural network for three-dimensional object recognition based on digital holography. *Optics Letters* **26**, 1478-1480 (2001).
175. Situ, G. & Sheridan, J. T. Holography: an interpretation from the phase-space point of view. *Optics Letters* **32**, 3492-3494 (2007).
176. Stern, A. & Javidi, B. Space-bandwidth conditions for efficient phase-shifting digital holographic microscopy. *Journal of the Optical Society of America A* **25**, 736-741 (2008).
177. Claus, D., Iliescu, D. & Bryanston-Cross, P. Quantitative space-bandwidth product analysis in digital holography. *Applied Optics* **50**, H116-H127 (2011).
178. Rogers, G. L. In-line soft-x-ray holography: the unwanted image. *Optics Letters* **19**, 67 (1994).
179. Xiao, T. Q. et al. Digital image decoding for in-line X-ray holography using two holograms. *Journal of Modern Optics* **45**, 343-353 (1998).
180. Tonomura, A. Applications of electron holography. *Reviews of Modern Physics* **59**, 639-669 (1987).

181. Tegze, M. & Faigel, G. X-ray holography with atomic resolution. *Nature* **380**, 49-51 (1996).
182. Korecki, P., Korecki, J. & Ślęzak, T. Atomic resolution $\gamma$-ray holography using the Mössbauer effect. *Physical Review Letters* **79**, 3518-3521 (1997).
183. Zhang, J. Y. et al. Phase-shifting lensless Fourier-transform holography with a Chinese Taiji lens. *Optics Letters* **43**, 4085-4087 (2018).
184. Zhang, S. M. et al. Phase-shifting radial-shearing digital holography with Greek-ladder zone plates. *Optics Letters* **43**, 5575-5578 (2018).
185. Liu, G. & Scott, P. D. Phase retrieval and twin-image elimination for in-line Fresnel holograms. *Journal of the Optical Society of America A* **4**, 159-165 (1987).
186. Teague, M. R. Deterministic phase retrieval: a Green's function solution. *Journal of the Optical Society of America* **73**, 1434-1441 (1983).
187. Barton, J. J. Removing multiple scattering and twin images from holographic images. *Physical Review Letters* **67**, 3106-3109 (1991).
188. Nugent, K. A. Twin-image elimination in Gabor holography. *Optics Communications* **78**, 293-299 (1990).
189. Tiller, J. B. et al. The holographic twin image problem: a deterministic phase solution. *Optics Communications* **183**, 7-14 (2000).
190. Bleloch, A. L., Howie, A. & James, E. M. Amplitude recovery in Fresnel projection microscopy. *Applied Surface Science* **111**, 180-184 (1997).
191. Levi, A. & Stark, H. Image restoration by the method of generalized projections with application to restoration from magnitude. *Journal of the Optical Society of America A* **1**, 932-943 (1984).
192. Shechtman, Y. et al. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE Signal Processing Magazine* **32**, 87-109 (2015).
193. Fournier, C. et al. Inverse problem approaches for digital hologram reconstruction. Proceedings of SPIE 8043, Three-Dimensional Imaging, Visualization, and Display 2011. Orlando, Florida, United States: SPIE, 2011.
194. McCann, M. T., J in, K. H. & Unser, M. Convolutional neural networks for inverse problems in imaging: a review. *IEEE Signal Processing Magazine* **34**, 85-95 (2017).
195. Misell, D. L. An examination of an iterative method for the solution of the phase problem in optics and electron optics: I. test calculations. *Journal of Physics D: Applied Physics* **6**, 2200-2216 (1973).
196. Greenbaum, A. & Ozcan, A. Maskless imaging of dense samples using pixel super-resolution based multi-height lensfree on-chip microscopy. *Optics Express* **20**, 3129-3143 (2012).
197. Zhang, J. C. et al. Phase unwrapping in optical metrology via denoised and convolutional segmentation networks. *Optics Express* **27**, 14903-14912 (2019).
198. Zhang, T. et al. Rapid and robust two-dimensional phase unwrapping via deep learning. *Optics Express* **27**, 23173-23185 (2019).
199. Dardikman-Yoffe, G. et al. PhUn-Net: ready-to-use neural network for unwrapping quantitative phase images of biological cells. *Biomedical Optics Express* **11**, 1107-1121 (2020).
200. Wu, C. C. et al. Phase unwrapping based on a residual en-decoder network for phase images in Fourier domain Doppler optical coherence tomography. *Biomedical Optics Express* **11**, 1760-1771 (2020).
201. Lyu, M. et al. Fast autofocusing in digital holography using the magnitude differential. *Applied Optics* **56**, F152-F157 (2017).
202. Bian, Y. X. et al. Optical refractometry using lensless holography and autofocusing. *Optics Express* **26**, 29614-29628 (2018).
203. Zhang, Y. B. et al. Edge sparsity criterion for robust holographic autofocusing. *Optics Letters* **42**, 3824-3827 (2017).
204. Jaferzadeh, K. et al. No-search focus prediction at the single cell level in digital holographic imaging with deep convolutional neural network. *Biomedical Optics Express* **10**, 4276-4289 (2019).
205. Xiao, W. et al. Sensing morphogenesis of bone cells under microfluidic shear stress by holographic microscopy and automatic aberration compensation with deep learning. *Lab on A Chip* **21**, 1385-1394 (2021).
206. Gopinathan, U., Pedrini, G. & Osten, W. Coherence effects in digital in-line holographic microscopy. *Journal of the Optical Society of America A* **25**, 2459-2466 (2008).
207. Dainty, J. C. et al. Laser Speckle and Related Phenomena. (Berlin, Heidelberg: Springer 1975).
208. Bianco, V. et al. Strategies for reducing speckle noise in digital holography. *Light: Science & Applications* **7**, 48 (2018).
209. Lehtinen, J. et al. Noise2Noise: learning image restoration without clean data. Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden: PMLR, 2018.
210. Yin, D. et al. Speckle noise reduction in coherent imaging based on deep learning without clean data. *Optics and Lasers in Engineering* **133**, 106151 (2020).
211. Blinder, D. et al. Signal processing challenges for digital holographic video display systems. *Signal Processing: Image Communication* **70**, 114-130 (2019).
212. Zhang, F. et al. An approach to increase efficiency of DOE based pupil shaping technique for off-axis illumination in optical lithography. *Optics Express* **23**, 4482-4493 (2015).
213. He, Z. H. et al. Progress in virtual reality and augmented reality based on holographic display. *Applied Optics* **58**, A74-A81 (2019).
214. Shimobaba, T. et al. Rapid calculation algorithm of Fresnel computer-generated-hologram using look-up table and wavefront-recording plane methods for three-dimensional display. *Optics Express* **18**, 19504-19509 (2010).
215. Ito, T. et al. Special-purpose computer HORN-5 for a real-time electroholography. *Optics Express* **13**, 1923-1932 (2005).
216. Goi, H., Komuro, K. & Nomura, T. Deep-learning-based binary hologram. *Applied Optics* **59**, 7103-7108 (2020).
217. Park, D. Y. & Park, J. H. Hologram conversion for speckle free reconstruction using light field extraction and deep learning. *Optics Express* **28**, 5393-5409 (2020).
218. Ren, H. R. et al. Three-dimensional vectorial holography based on machine learning inverse design. *Science Advances* **6**, eaaz4261 (2020).
219. Goodman, J. W. et al. Optical interconnections for VLSI systems. *Proceedings of the IEEE* **72**, 850-866 (1984).
220. Wetzstein, G. et al. Inference in artificial intelligence with deep optics and photonics. *Nature* **588**, 39-47 (2020).
221. Luo, X. H. et al. Metasurface-enabled on-chip multiplexed diffractive neural networks in the visible. arXiv: 2107.07873 (2021).
222. Zhou, T. K. et al. Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nature Photonics* **15**, 367-373 (2021).
223. Zhou, T. K. et al. In situ optical backpropagation training of diffractive optical neural networks. *Photonics Research* **8**, 6000940 (2020).
224. Xiao, Y. L. et al. Unitary learning for diffractive deep neural network. *Optics and Lasers in Engineering* **139**, 106499 (2021).
225. Xiao, Y. L. et al. Optical random phase dropout in a diffractive deep neural network. *Optics Letters* **46**, 5260-5263 (2021).
226. Qian, C. et al. Performing optical logic operations by a diffractive neural network. *Light: Science & Applications* **9**, 59 (2020).
227. Wang, P. P. et al. Orbital angular momentum mode logical operation using optical diffractive neural network. *Photonics Research* **9**, 2116-2124 (2021).
228. Kulce, O. et al. All-optical information-processing capacity of

diffractive surfaces. *Light: Science & Applications* **10**, 25 (2021).
229. Huang, Z. B. et al. All-optical signal processing of vortex beams with diffractive deep neural networks. *Physical Review Applied* **15**, 014037 (2021).
230. Rahman, S. S. & Ozcan, A. Computer-free, all-optical reconstruction of holograms using diffractive networks. arXiv: 2107.08177 (2021).
231. Veli, M. et al. Terahertz pulse shaping using diffractive surfaces. *Nature Communications* **12**, 37 (2021).
232. Li, J. X. et al. Spectrally encoded single-pixel machine vision using diffractive networks. *Science Advances* **7**, eabd7690 (2021).
233. Wang, F. et al. Single-pixel imaging using physics enhanced deep learning. *Photonics Research* **10**, 104-110 (2022).
234. Cai, X. D. et al. Dynamically controlling terahertz wavefronts with cascaded metasurfaces. *Advanced Photonics* **3**, 036003 (2021).
235. Lobo, J. L. et al. Spiking Neural Networks and online learning: an overview and perspectives. *Neural Networks* **121**, 88-100 (2020).