Optics EXPRESS

Learning from simulation: An end-to-end deep-learning approach for computational ghost imaging

FEI WANG,[1,2] HAO WANG,[1,2] HAICHAO WANG,[1,2] GUOWEI LI,[1,2] AND GUOHAI SITU[1,2,*]

[1] *Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai, 201800, China*
[2] *Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China*
[*] *Corresponding author: ghsitu@siom.ac.cn*

Abstract: Artificial intelligence (AI) techniques such as deep learning (DL) for computational imaging usually require to experimentally collect a large set of labeled data to train a neural network. Here we demonstrate that a practically usable neural network for computational imaging can be trained by using simulation data. We take computational ghost imaging (CGI) as an example to demonstrate this method. We develop a one-step end-to-end neural network, trained with simulation data, to reconstruct two-dimensional images directly from experimentally acquired one-dimensional bucket signals, without the need of the sequence of illumination patterns. This is in particular useful for image transmission through quasi-static scattering media as little care is needed to take to simulate the scattering process when generating the training data. We believe that the concept of training using simulation data can be used in various DL-based solvers for general computational imaging.

1. Introduction

In recent years, learning-based methods have been widely used to solve problems in divergent fields, such as visual object recognition, natural language processing, object detection, and among many others [1]. People have also proposed to use learning-based methods such as support vector regression (SVR) [2] and deep learning (DL) to solve inverse problems in optical imaging [3]. In the latter case, people have used DL in optical tomography [4], computational ghost imaging [5,6], digital holography [7–9], imaging through scattering media [10–12], fluorescence lifetime imaging [13], lensless imaging [14], and imaging in low light condition [5,6,15]. This kind of techniques usually require many hours or even days to experimentally collect tens of thousands of labeled data for neural network training. This is not easily affordable, in particular in the application of computational ghost imaging, which requires many exposures corresponding different modes of structured illumination to sample an object or a scene. Here we demonstrate that a practically usable neural network for computational imaging can be trained by using simulation data. Particularly, we take computational ghost imaging (CGI) for our demonstration.

It is well known that in ghost imaging [16–25] image reconstruction is achieved by the numerical correlation of two beams, an object beam and a reference beam. The object beam interacts with the object and is collected by a single-pixel camera whereas the reference beam never interacts with the object and is recorded by a high-spatial-resolution detector. GI was first demonstrated as a manifest of quantum entanglement of photon pairs [16], but soon after the demonstration of GI was also realized by classical light sources [18,19].

GI can be implemented in a single-beam geometry if the reference beam can be numerically evaluated, in the case that the random patterns applied to the object are known or pre-specifiable. In this way, the reconstruction can be computational, and thus it earns the term of computational

#365138 https://doi.org/10.1364/OE.27.025560
Journal © 2019 Received 16 Apr 2019; revised 11 Aug 2019; accepted 13 Aug 2019; published 26 Aug 2019

ghost imaging (CGI) [26,27]. Usually, spatial light modulators (SLMs) offer the ability to program the illumination beam with a sequence of, say, M, random patterns. The integral of each modulated image is acquired by a bucket detector that is synchronized with the SLM, yielding a series of one-dimensional (1D) intensity data of length M. This high-efficient detection scheme is in particular desirable in low light environments [28], X-ray imaging [29,30], multipspectral imaging [31], information security [32,33] and sensing in remote distance [34,35], to name a few.

In CGI, the number of resolution cells N that covers an object is usually equal to the number of illumination patterns M that were applied to the object according to the Nyquist sampling criterion [36]. Thus, in order to achieve high resolution imaging, one usually needs to have many incoherent illumination patterns to interacts with the object, i.e., M should be as large as possible. However, the sequential loading of a large set of illumination patterns to the SLM is usually time-consuming due to its modulation frequency [37]. Effects have been made to increase the imaging speed, for example by multiplexing the SLM [38], or replacing the SLM with a programmable LED array [39]. An alternative approach is to reduce the number of sampling M. When the sampling ration $\beta = M/N$ is reduced, one needs to reformulate the reconstruction as an inverse problem and employ optimization algorithms such as compressive sensing (CS) [40,41], Gerchberg-Saxton algorithm [42] and other iteration algorithms [17,43]. In particular, compressive ghost imaging (CSGI) [40,41] enables the reconstruction of an N-pixel image from $M(\ll N)$ measurements by exploiting the sparsity of the object. However, the degree of down-sampling is limited by the sparsity of the object [44] and the quality of the reconstructed image is sensitive to detection noise [5,6].

Deep learning has been used for CGI as well. It was first proposed by Lyu *et al.* [5]. They have demonstrated the reconstruction of high quality images with significant reduced number of sampling ($\beta \ll 1$). In their method, the same set of illumination patterns were applied to interact with the objects in both the training and test sets. As β is small, the images reconstructed directly using correlation [26,27] from the acquired bucket signals were severely corrupted by noise. These noisy images were paired up with the corresponding known object images and were used to train the DNN. The trained DNN then was used to improve the Signal-to-Noise-Ratio (SNR) of test images reconstructed in the first step. We note that the same strategy was used in [6].

However, in all the CGI algorithms proposed so far (including those based on DL), the sequence of random patterns interact with the object should be known. In the applications of remote sensing, atmosphere turbulence prevents the detection or evaluation of the exact disturbance of the random patterns, resulting in image degradation in the reconstruction [34,45]. Thus the other motivation of the present work is to design an end-to-end approach that can reconstruct the object image directly from the bucket signal, without the use of the illumination patterns. As schematically outlined in Fig. 1, the approach proposed here in this manuscript is inspired by the technique of deep learning. In most DL-based computational imaging, thousands pairs of labeled input-output data should be acquired to train a neural network. This will take a lot more time in CGI owing to the special way of data acquisition using a bucket detector. So the other motivation in this manuscript is to demonstrate the use simulation data to train the network. This requires the close simulation of the physical process of CGI. Once trained, the neural network can be used for image reconstruction from experimentally acquired bucket signals that are physically modulated by the same set of random patterns.

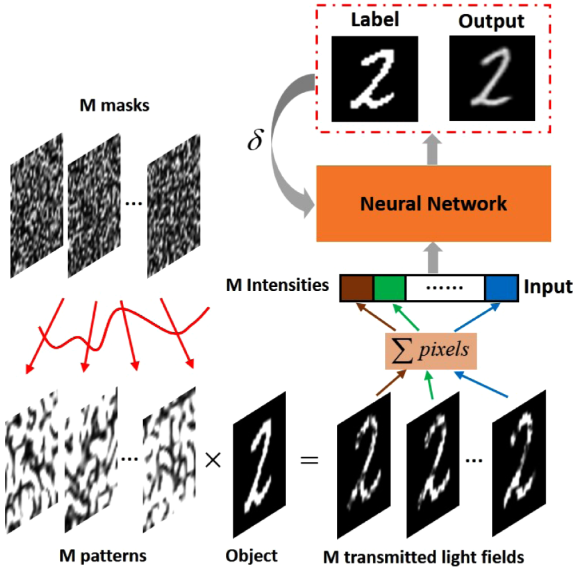

Fig. 1. Schematic illustration of the training pipeline of the proposed end-to-end deep learning ghost imaging. A 2D object function is sequentially multiplied by a real-valued random pattern, which is produced by numerical calculation of free space propagation of a random phase function. The resulting amplitude is then numerically propagated over a distance and reaches the bucket detector. This will end up with a digit which represents the total energy of the light field in this measurement. Running through this calculation process M times with M different random patterns, we will have a sequence of digits with the length of M associated with one object. Then we run the above calculation over again with many different object functions, and obtain associated bucket signals. Then we feed the ground-truth images and their corresponding bucket signals into a Neural Network and optimize its weights and biases. The neural network trained by simulation data produced in this way is used to reconstruct images from experimental acquired data under the illumination of the same set of M random patterns.

2. Method

2.1. Learning-based approach

For simplicity, we focus our discussion on the imaging of two-dimensional images, although GI can be used for 3D imaging. Let us denote the object as $T(x, y)$, where (x, y) is the transverse coordinates at the object plane. The random patterns that interact with the object are denoted by $I_m(x, y)$, where $m = 1, 2, \ldots, M$. The 1D signal acquired by the bucket detector then can be written as

$$S_m = \int I_m(x, y)T(x, y)\mathrm{d}x\mathrm{d}y. \tag{1}$$

Following the principle of GI, the image can be reconstructed by the correlation of the intensity fluctuation with the illumination speckle patterns

$$\widetilde{T}(x, y)_{GI} = \langle \Delta S_m \Delta I_m \rangle, \tag{2}$$

where $\langle\cdot\rangle$ denotes an ensemble average over M measurements and $\langle \Delta S_m \rangle = S_m - \langle S_m \rangle$, $\langle \Delta I_m \rangle = I_m - \langle I_m \rangle$.

In CSGI, the image reconstruction is treat as an optimization problem and solved using compressed sensing algorithms. This is to find a solution

$$\widetilde{T}(x,y)_{CS} = T^{'}(x,y) \tag{3}$$

that minimizes the L_1 norm in the sparse basis

$$\arg\min ||\Psi\{T^{'}(x,y)\}||_{L_1}, \tag{4}$$

subject to

$$\int I_m(x,y)T^{'}(x,y)\mathrm{d}x\mathrm{d}y = S_m, \quad \forall m = 1,2,\ldots,M, \tag{5}$$

where Ψ is the transform operator to the sparse basis.

The method we propose here employs a deep neural network to reconstruct the object image directly from the bucket signal S_m, $m = 1,2,\ldots,M$. The reconstruction process can be expressed as

$$\widetilde{T} = \mathcal{R}\{S_m\}, \tag{6}$$

where $\mathcal{R}\{\cdot\}$ represents the neural network that maps the bucket signal S_m back to the object space. This mapping from a 1D signal to a 2D image without knowing the transformation matrix is highly ill posed. Here we propose to learn a feasible neural network $\widetilde{\mathcal{R}}_{\text{learn}}$ from a set of labeled data each of which pairs up a known object $T(x,y)^j$ and the corresponding sequence of bucket signal S_m^j, where $j = 1,\ldots,J$, enumerates the total J different pairs of labeled data used for training. The network in this case can be written as

$$\widetilde{\mathcal{R}}_{\text{learn}} = \underset{\mathcal{R}_\theta,\theta\in\Theta}{\arg\min} \sum_{j=1}^{J} L(T(x,y)^j, \mathcal{R}_\theta\{S_m^j\}) + \varphi(\theta) \tag{7}$$

where Θ is the set of all possible parameters, $L(\cdot)$ is a loss function to measure the error between the network output $\mathcal{R}_\theta\{S_m^j\}$ and the ground-truth $T(x,y)^j$, and $\varphi(\theta)$ is a regularization term on the parameters to avoid overfitting [46]. The set Θ contains two types of parameters: non-learnable parameters and learnable parameters. The first type includes the parameters that specify the structure of the network, such as the type of the network, the number of layers, the number of neurons in each layer, and the size of the convolutional kernel, etc. that need to be manually specified before training. The other type includes the connection weights and bias between the neurons in two neighboring layers that should be adjusted automatically during the training phase. Once $\widetilde{\mathcal{R}}_{\text{learn}}$ has been learned, the object image can be reconstructed according to

$$\widetilde{T}(x,y)_{DL} = \widetilde{\mathcal{R}}_{\text{learn}}\{S_m\}. \tag{8}$$

In contrast, the image reconstruction process in previous DL-based methods [5,6] was given by

$$\widetilde{T}(x,y)^{'}_{GIDL} = \widetilde{\mathcal{R}}^{'}_{\text{learn}}\{\widetilde{T}(x,y)_{GI}\}, \tag{9}$$

where $\widetilde{T}(x,y)_{GI}$ is obtained according to Eq. (2) and

$$\widetilde{\mathcal{R}}^{'}_{\text{learn}} = \underset{\mathcal{R}^{'}_\theta,\theta\in\Theta}{\arg\min} \sum_{j=1}^{J} L(T(x,y)^j, \mathcal{R}^{'}_\theta\{\widetilde{T}(x,y)^j_{GI}\}) + \varphi(\theta) \tag{10}$$

Obviously, the input data to train the network $\widetilde{\mathcal{R}}^{'}$ are the images $\widetilde{T}(x,y)^j$ reconstructed by using the conventional GI algorithm and the corresponding ground truth $T(x,y)^j$. When the sampling ratio β is small, the SNR of $\widetilde{T}(x,y)^j$ is low. Then the neural network trained according to Eq. (10) is more like denoising [47].

2.2. Network structure

Inspired by ResNet [48] and eHoloNet [8], we propose a neural network whose structure is schematically shown in Fig. 2. The input of the network is the normalized bucket signal with the length of M, while the output should be the predicted object image. We mainly used three types of modules to connect the input to the output: Fully connected layers, convolution blocks, and residual blocks. Two fully connected layers of 1024×1 and 4096×1 in size, respectively, are used to discover the association of the values in the input bucket signal. The output of the second fully connected layer is reshaped to an image with the size of 64×64. Then we exploit the powerful convolutional neural network (CNN) to reconstruct the object image. We used 4 independent paths, each of which has a max-pooling layer to downsample the incoming image with a magnitude of different order of 2, creating 4 independent data flows. Then each data flow is sent to a set of 4 identical residual blocks, which are used to extract feature maps at different scales. Followed the residual layers, there are up-sampling layers that can restore the size of feature maps back to 64×64. Then the 4 paths are concatenated into one. The concatenated image then passes through 4 convolution layers and 1 max-pooling layer and yield the reconstructed image. In Fig. 2, a pair of digits in the format of $m - n$ is placed below each convolutional layer and up-sampling layer to denote the size of the input and the size of the output of this layer. We also used dropout layers and batch-normalization (BN) layers to prevent overfitting [49,50].

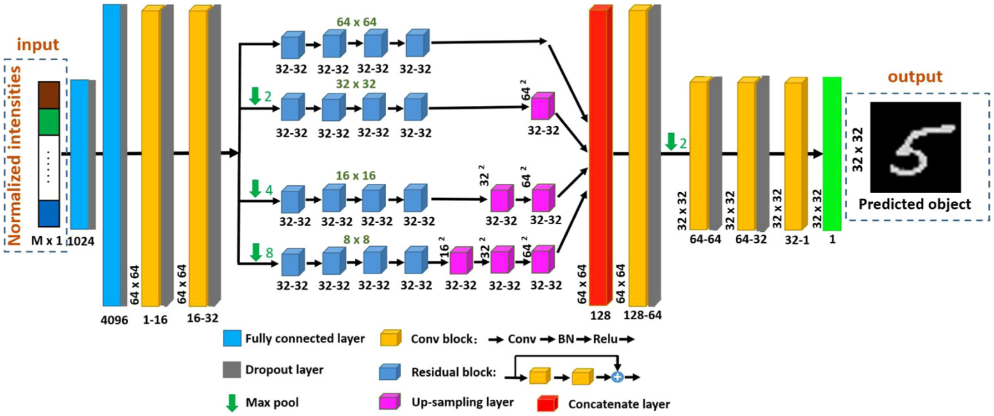

Fig. 2. Proposed neural network architecture to learn the image restore principle from measured intensities in GI.

2.3. Network training

As mentioned before, the training of the network is a process to optimize the values of the parameters in the set Θ. These parameters include the weighting factors and bias connected the neurons in two neighboring layers. In the case of supervised learning as in our study, we need a substantial collection of known images and their bucket signals as constraints to iteratively optimize the neural network so that it can reconstruct an expected image from a bucket signal in the test set. In the training process, we define the loss function as the mean square error (MSE) between the reconstructed image and the corresponding known image (ground truth):

$$L = \frac{1}{J' WH} \sum_{j=1}^{J'} \sum_{u=1}^{W} \sum_{v=1}^{H} \left(\mathcal{R}_\theta \{ S_m^j \} - T(x, y) \right)^2 \tag{11}$$

where W, H are the width and height of the reconstructed image, respectively, and $J' = 5$ is the mini-batch size in the stochastic gradient descent (SGD) method [51]. We adopted the Adam optimizer [52] to optimize the weights and set the learning rate at 0.01. The training step was 300, 000. The keep probability of the dropout layer was 0.9. The program was implemented in Python version 3.6 using TensorFlow. A graphics processing unit (NVIDIA Quadro P6000) was used to accelerate the computation.

2.4. Preparation of training data

The training of a neural network usually needs a lot of labeled data. In ghost imaging, the bucket signal is generated by actively illuminating an object with many structural patterns as discussed above. The collection of all the labeled data experimentally in time consuming, if not unfeasible. For example, we used 9, 000 images from the MNIST hand written digit databases [53] and their corresponding bucket signals each of which is $M = 64$ in length to train the network. The total number of measurement is 576, 000.

In order to reduce the cost of data collection, we propose a framework of using the simulated data to train the network, which is shown in Fig. 1. This was performed to simulate the process of bucket signal generation. The simulation should be as close as possible to the experiment. We assumed that the light source was a collimated coherent laser beam with the wavelength of 633 nm. It was modulated by a random pattern, and numerically propagated over a distance $d = 3$ m, resulting a pattern $I_m(x, y)$ that interacts with the object. The random pattern $I_m(x, y)$ was then multiplied with an object $T(x, y)$ in the training set, and the resulting 2D intensity disturbance was integrated to generate one data point of the bucket signal. In our study, a same set of M different random patterns of 32×32 in size was used in both the simulation and experiment. After the set of M patterns ran over with the same T, we obtain a bucket signal S_m of length M. We then paired up the resulting signal S_m and the corresponding 2D image T. Then we replaced the object and ran the above simulation process again, and obtained another pair of labeled data. This process was run over and over again across the whole 9, 000 objects in the training set were paired up with their corresponding bucket signals.

In both the simulation and experiment, we binarized and then resized the MNIST hand written digits in use to 32×32 so that we have $N = 1024$. The sampling ratio $\beta = M/1024$. In our study, we will examine how the value of β affects the reconstructed image. We mainly considered 5 different cases for M: 1024, 256, 64, 16 and 4 so that $\beta = 100$ %, 25 %, 6.25 %, 1.56 % and 0.39 %, respectively. In the simulation, the sampling interval of the random patterns and the object images is 128 μm in the transverse directions.

3. Results and discussions

3.1. Experimental setup

The experimental setup is schematically showed in Fig. 3, which is actually a CGI geometry. Polarized light emitted from a He-Ne laser (Thorlabs, HRS015) with the wavelength $\lambda = 633$ nm was first coupled into a polarization maintaining fiber then expanded and collimated by a lens L1. Then the collimated beam was splitted into two arms by a beam splitter BS1. The transmitted beam then was shane onto a digital mirror device (DMD), where a set of $M = 64$ random patterns, $I_m(x, y)$, $m = 1, \ldots, M$, were sequentially displayed. These random patterns were precomputed by Fresnel diffraction over 3 m (see Sec. 3.2.4 for details). The beam reflected from the SLM was reflected by BS1 and then projected onto an SLM (Pluto-Vis, Holoeye Photonics AG) using a 4f system consisting of lenses L2 and L3. The object image, $T(x, y)$, was displayed on the SLM. The beam reflected from the SLM was collected by a bucket detector through a lens L4. We used an sCMOS camera (Zyla 4.2 PLUS sCMOS, Andor Technology Ltd) in our experiment instead because we do not have a bucket detector. But we integrated each acquired image to produce the bucket signal S_m. This does not affect the measurement results.

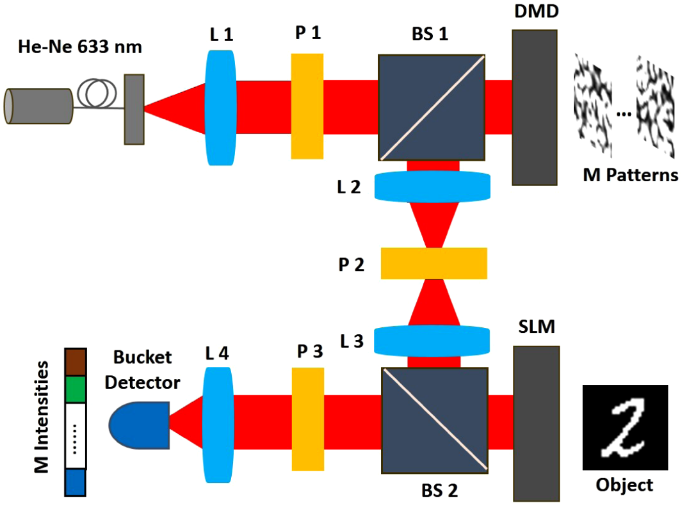

Fig. 3. Schematic diagram of the optical setup. L1, L2, L3, L4 are lenses with focal length of 80mm, 80mm, 80mm, 40mm. P1, P2 and P3 are linear polarizers. P1 and P3 are vertically polarized, and P2 is horizontally polarized. BS1 and BS2 are both beam splitter. DMD: digital micro-mirror device. SLM: spatial light modulator. Patterns used to modulate the light field were sequentially displayed on the DMD. The object was placed on the SLM.

Note that the pixel size of the SLM and the DMD we used in the experiment is 8 μm and 10.8 μm, respectively. So cares should be taken in both the simulation and experiment so that the training data obtained by numerical simulations were generated in the same conditions as in the experiment. Taking this into account, we approximately used 512×512 and 379×379 pixels to represent the object and the random patterns, respectively, to meet the 128 μm sampling interval in the simulation. Through the established experimental setup and prepared data set we collected the bucket signals corresponding to $1,000$ objects in the test set.

3.2. Results

The main results are plotted in Fig. 4. One can clearly see that, in both the simulation and experiment, the object image can be successfully reconstructed from the 1D bucket signal with the sampling ratio as low as $\beta = 1.56$ %, although the images are apparently distorted. But the images are nearly perfectly reconstructed when β increases to 6.25 %. However, no image can be reconstructed when $\beta = 0.39$ % in our experiment.

We compared the proposed DL-based (DLGI) method with the conventional correlation-based CGI, and compressive-sensing-based GI (CSGI) in terms of reconstruction performance with respect to the sampling ratio β. For conventional CGI, when β is high, for instance, $\beta = 100$ %, the reconstructed images still contains a lot of noise. This is common because the SNR in this case is just 1 [37]. As expected, the SNR becomes even worse when β becomes low, and becomes completely corrupted when $\beta = 6.25$ %.

The situations become much improved when CSGI is used to reconstruct the image. When $\beta = 100$ % and $\beta = 25$ % the object image can be perfect reconstructed in simulation using the TVAL3 algorithm [54]. But CSGI is sensitive to noise [5]. This affects the actual performance, as evidenced by the second row in Fig. 4.

Thus, we can see that the proposed method has the best performance among the three when the sampling ratio β is larger than 6.25 %. More evidences are shown in Fig. 5.

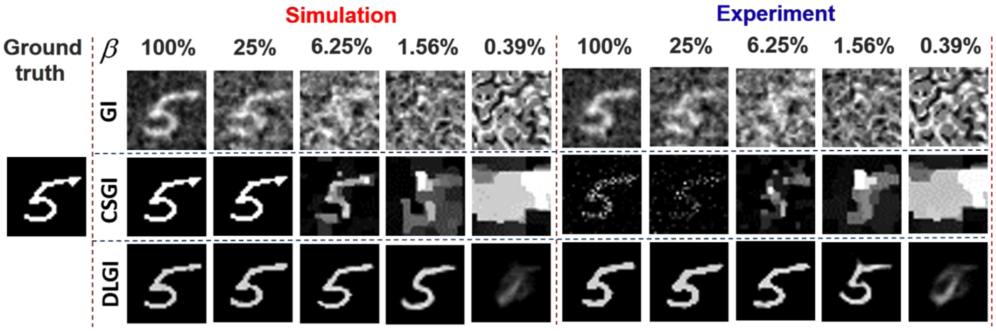

Fig. 4. Comparison of simulation and experiment results from GI, CSGI, DLGI at different β.

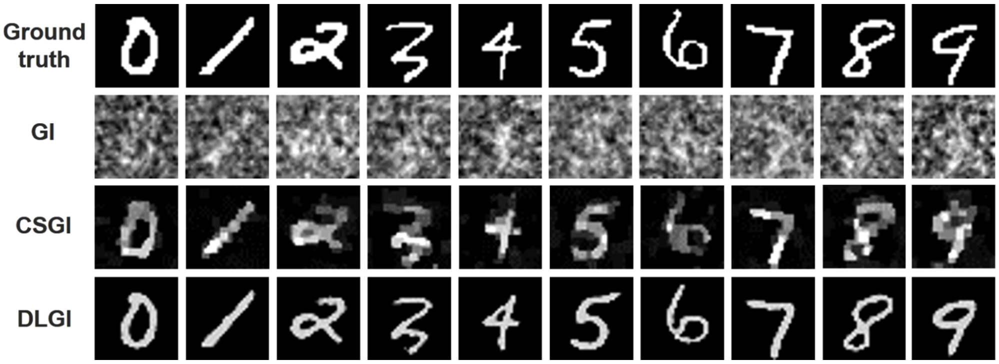

Fig. 5. Comparison of experiment results from GI, CSGI, DLGI when $\beta = 6.25\%$.

3.3. Discussions

3.3.1. Accuracy

Now we make a quantitative evaluation of the performance by using three metrics, namely, the prediction accuracy by Support Vector Machines (SVM), the model of which we used is described in [55], the root of mean square error (RMSE)

$$\mathrm{RMSE} = \left[\frac{1}{WH} \sum_{u=1}^{W} \sum_{v=1}^{H} (\widetilde{T}(x,y) - T(x,y))^2 \right]^{\frac{1}{2}} \tag{12}$$

the structural similarity index (SSIM) [56]

$$\mathrm{SSIM} = \frac{(2u_{\widetilde{T}} u_T + c_1)(\sigma_{\widetilde{T}T} + c_2)}{(u_{\widetilde{T}}^2 + u_T^2 + c_1)(\sigma_{\widetilde{T}}^2 + \sigma_T^2 + c_2)} \tag{13}$$

where $u_f, f \in \{T, \widetilde{T}\}$, is the mean of the image f, σ_f^2 is the variance, $\sigma_{\widetilde{T}T}$ is the covariance of $\widetilde{T}$ and T, and c_1 and c_2 are regularization parameters.

We calculated the averaged values of these three metrics of 100 reconstructed images randomly selected out of the total $1,000$ in the test set, and plot the results in Fig. 6. As expected, the conventional correlation-based CGI has the worst performance measured by all these three metrics. CSGI performs better than the conventional correlation-based CGI under different sampling conditions. This is expectable, as convinced also by Kate *et al.* [40]. The proposed

DLGI has the overall best performance measured by all these three metrics, in a very good consistence with the results plotted in Fig. 4. In particular, Fig. 6(c) suggests that the proposed method can reconstruct images nearly perfectly with the SPA value goes up to 0.92 even when β is as small as 6.25 %.

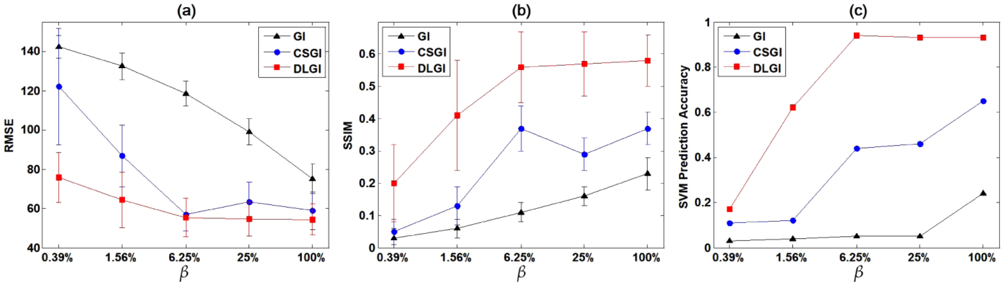

Fig. 6. Quantitative evaluation of GI, CSGI, DLGI in experiment. $N = \beta \cdot W \cdot H$ is the sampling times. Each marker represents the mean performance from 100 different test objects at different β. Each error bar represents the standard deviation. (a) RMSE indicator curve. (b) SSIM indicator curve. (c) SVM prediction accuracy indicator curve.

3.3.2. Robustness

The above experimental results demonstrate the usability of simulation data in the training of the neural network that can reconstruct ghost images from experimental data in the test set. The reason for this is that the simulation should be run in the geometric and sampling conditions as close to the experimental setup as possible. This is evidenced by the simulated and experimental acquired bucket signals of the same object plotted in Fig. 7. One can see clearly that the two sequences are highly coincident with each other. Indeed, we calculated the Pearson correlation coefficient (PCC) between them, and the value is 0.91. For all the 1,000 images in the test set, the averaged PCC is 0.9037.

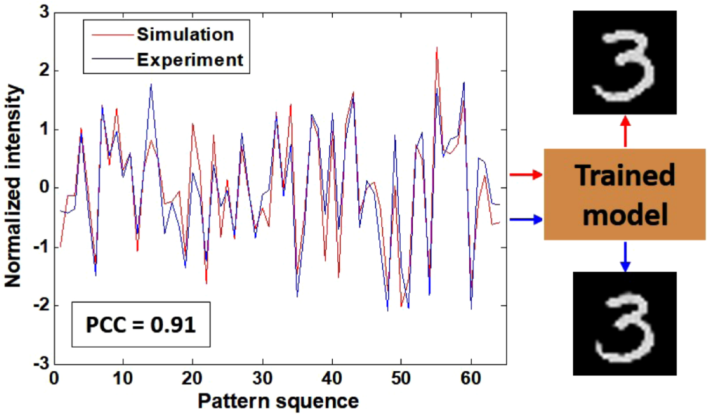

Fig. 7. Comparison of model recovery performance between simulation and experimental data.

3.3.3. Generalization

To examine the generalization of the trained neural network model, here we use it to recover the images of objects that are different from those in the training set. For convenience, we test it with English letters and double-seam patterns. Some of the reconstructed images are shown in Fig. 8. Although the network was trained by using the MNIST handwritten data set, it is clearly seen that it can be used to reconstruct the images of English letters and the double-seam patterns when the sampling ratio β is 25 % and above. The averaged SSIM is 0.69 when $\beta = 25$ %. Even when $\beta = 6.25$ % the reconstructed images are still distinguishable, although they are distorted. This averaged SSIM in this case is 0.57, at the same level as that of the reconstructed handwritten digits plotted in Fig. 6(b). We also observed that the images cannot be reconstructed when β is lower that 6.25%. This is reasonable because of insufficiency of acquired information, as also evidenced in Fig. 4.

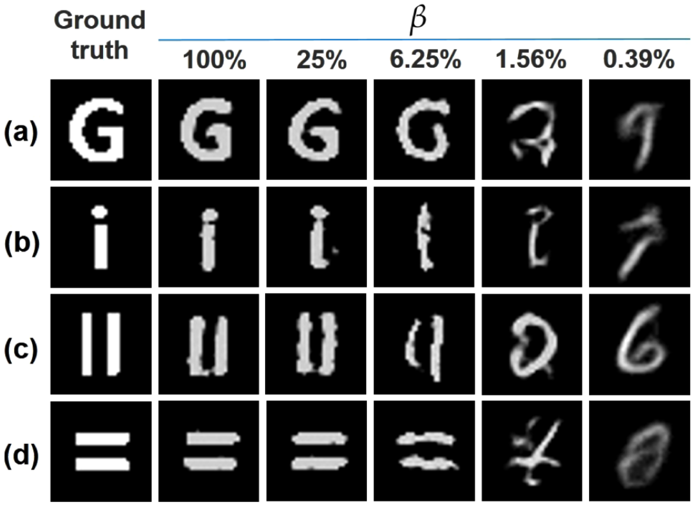

Fig. 8. Network generalization test results.

3.4. Image transmission through scattering layers

Image transmission through scattering media is a critical issue with ghost imaging [57]. As mentioned in the Introduction, deep learning has been used for coherent imaging through scattering media under the illumination of continuous coherent light [10–12]. Here we demonstrate that the proposed ghost imaging neural network can be used to solve this problem as well. To prove this concept, we placed a ground glasses (Thorlabs, DG100X100-220) in a position between the object and the bucket detector, and collected the bucket signal, as schematically shown in Fig. 9(a). We then used the same neural network trained previously (without any scattering) to reconstruct the image. The experimental results are shown in Fig. 9(b). The direct images captured by a camera are shown in the *speckle* columns, which are speckle patterns of course. However, one can see that the trained neural network can successfully restore the images of the objects hidden behind the diffuser. We observed that it does not mattered at all if we changed the diffuser or its position, suggesting that the proposed method is quite robust against the realization of the random modulation that is introduced by the diffuser. Different from the previous research by Li *et al* [11], the proposed method does not need to model the scattering process when calculating the bucket signals in the training process. Instead, it can be scalable to different scattering media by taking the advantage of ghost imaging, which reconstructs the object image using the integration its speckle patterns.

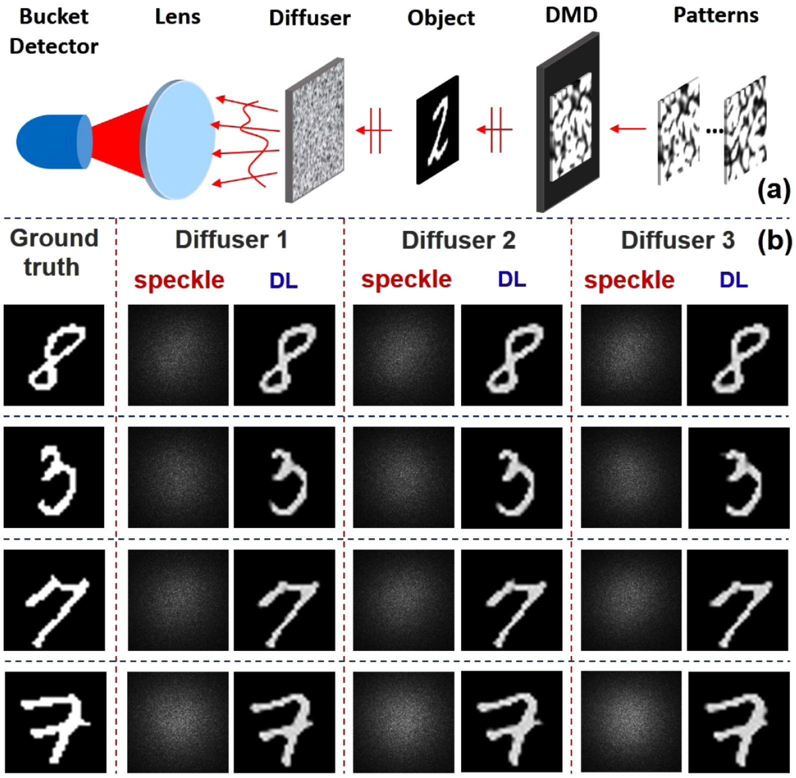

Fig. 9. Testing results of the proposed DLGI method dealing with scattering media when $\beta = 6.25$ %.

4. Conclusion

In conclusion, we have demonstrated a deep-learning-based image reconstruction method for ghost imaging in this paper. We have developed a neural network to restore the object image directly from the measured bucket signal. We have demonstrated that the network can be well trained by using only simulation data, so that the cost of training can be significantly reduced. This is achieved by closed simulating the experimental data acquisition process. We have analyzed the performance of the proposed ghost imaging method under different sampling ratio conditions, and compared it with conventional GI and CSGI. Our observation suggests that the proposed method has much better performance in comparison to the other two especially at low sampling ratio. This has significant potential to increase the time efficiency of data acquisition in practical applications. We have also demonstrated image transmission through scattering media using the proposed ghost imaging method. One advantage that the proposed method can offer is that one does not need to take the scattering into account when generating the simulation data to train the network owing to the mechanism of ghost imaging. This can be applied in the circumstance that the scattering layer is relatively static during the course of data acquisition.

Funding

Chinese Academy of Sciences (QYZDB-SSW-JSC002); Sino-German Center (GZ 1391).

References

1. Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature **521**(7553), 436–444 (2015).

2. R. Horisaki, R. Takagi, and J. Tanida, “Learning-based imaging through scattering media,” Opt. Express **24**(13), 13738–13743 (2016).
3. G. Barbastathis, A. Ozcan, and G. Situ, “On the use of deep learning for computational imaging,” Optica **6**(8), 921–943 (2019).
4. U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis, “Learning approach to optical tomography,” Optica **2**(6), 517–522 (2015).
5. M. Lyu, W. Wang, H. Wang, H. Wang, G. Li, N. Chen, and G. Situ, “Deep-learning-based ghost imaging,” Sci. Rep. **7**(1), 17865 (2017).
6. Y. Hu, G. Wang, G. Dong, S. Zhu, H. Chen, A. Zhang, and Z. Xu, “Ghost imaging based on deep learning,” Sci. Rep. **8**(1), 6469 (2018).
7. Z. Ren, Z. Xu, and E. Y. Lam, “Learning-based nonparametric autofocusing for digital holography,” Optica **5**(4), 337–344 (2018).
8. H. Wang, M. Lyu, and G. Situ, “eHoloNet: A learning-based end-to-end approach for in-line digital holographic reconstruction,” Opt. Express **26**(18), 22603–22614 (2018).
9. Y. Rivenson, Y. Zhang, H. Günaydin, D. Teng, and A. Ozcan, “Phase recovery and holographic image reconstruction using deep learning in neural networks,” Light: Sci. Appl. **7**(2), 17141 (2018).
10. M. Lyu, H. Wang, G. Li, S. Zheng, and G. Situ, “Learning-based lensless imaging through optically thick scattering media,” Adv. Photon. **1**(3), 036002 (2019).
11. Y. Li, Y. Xue, and L. Tian, “Deep speckle correlation: A deep learning approach toward scalable imaging through scattering media,” Optica **5**(10), 1181–1190 (2018).
12. S. Li, M. Deng, J. Lee, A. Sinha, and G. Barbastathis, “Imaging through glass diffusers using densely connected convolutional networks,” Optica **5**(7), 803–813 (2018).
13. G. Wu, T. Nowotny, Y. Zhang, H. Q. Yu, and D. D. Li, “Artificial neural network approaches for fluorescence lifetime imaging techniques,” Opt. Lett. **41**(11), 2561–2564 (2016).
14. A. Sinha, J. Lee, S. Li, and G. Barbastathis, “Lensless computational imaging through deep learning,” Optica **4**(9), 1117–1125 (2017).
15. A. Goy, K. Arthur, S. Li, and G. Barbastathis, “Low photon count phase retrieval using deep learning,” Phys. Rev. Lett. **121**(24), 243902 (2018).
16. T. B. Pittman, Y. H. Shih, D. V. Strekalov, and A. V. Sergienko, “Optical imaging by means of two-photon quantum entanglement,” Phys. Rev. A **52**(5), R3429–R3432 (1995).
17. F. Ferri, D. Magatti, L. A. Lugiato, and A. Gatti, “Differential ghost imaging,” Phys. Rev. Lett. **104**(25), 253603 (2010).
18. R. S. Bennink, S. J. Bentley, and R. W. Boyd, ““two-photon” coincidence imaging with a classical source,” Phys. Rev. Lett. **89**(11), 113601 (2002).
19. J. Cheng and S. Han, “Incoherent coincidence imaging and its applicability in x-ray diffraction,” Phys. Rev. Lett. **92**(9), 093903 (2004).
20. A. Gatti, E. Brambilla, M. Bache, and L. A. Lugiato, “Ghost imaging with thermal light: Comparing entanglement and classical correlation,” Phys. Rev. Lett. **93**(9), 093602 (2004).
21. F. Ferri, D. Magatti, A. Gatti, M. Bache, E. Brambilla, and L. A. Lugiato, “High-resolution ghost image and ghost diffraction experiments with thermal light,” Phys. Rev. Lett. **94**(18), 183602 (2005).
22. A. Gatti, E. Brambilla, M. Bache, and L. A. Lugiato, “Correlated imaging, quantum and classical,” Phys. Rev. A **70**(1), 013802 (2004).
23. R. S. Bennink, S. J. Bentley, R. W. Boyd, and J. C. Howell, “Quantum and classical coincidence imaging,” Phys. Rev. Lett. **92**(3), 033601 (2004).
24. G. Scarcelli, V. Berardi, and Y. Shih, “Can two-photon correlation of chaotic light be considered as correlation of intensity fluctuations?” Phys. Rev. Lett. **96**(6), 063602 (2006).
25. Y.-K. Xu, W.-T. Liu, E.-F. Zhang, Q. Li, H.-Y. Dai, and P.-X. Chen, “Is ghost imaging intrinsically more powerful against scattering?” Opt. Express **23**(26), 32993–33000 (2015).
26. J. H. Shapiro, “Computational ghost imaging,” Phys. Rev. A **78**(6), 061802 (2008).
27. Y. Bromberg, O. Katz, and Y. Silberberg, “Ghost imaging with a single detector,” Phys. Rev. A **79**(5), 053840 (2009).
28. P. A. Morris, R. S. Aspden, J. E. Bell, R. W. Boyd, and M. J. Padgett, “Imaging with a small number of photons,” Nat. Commun. **6**(1), 5913 (2015).
29. H. Yu, R. Lu, S. Han, H. Xie, G. Du, T. Xiao, and D. Zhu, “Fourier-transform ghost imaging with hard x rays,” Phys. Rev. Lett. **117**(11), 113901 (2016).
30. D. Pelliccia, A. Rack, M. Scheel, V. Cantelli, and D. M. Paganin, “Experimental X-ray ghost imaging,” Phys. Rev. Lett. **117**(11), 113902 (2016).
31. L. Bian, J. Suo, G. Situ, Z. Li, J. Fan, F. Chen, and Q. Dai, “Multispectral imaging using a single bucket detector,” Sci. Rep. **6**(1), 24752 (2016).
32. P. Clemente, V. Durán, V. Torres-Company, E. Tajahuerce, and J. Lancis, “Optical encryption based on computational ghost imaging,” Opt. Lett. **35**(14), 2391–2393 (2010).
33. B. Javidi, A. Carnicer, M. Yamaguchi, T. Nomura, E. Pérez-Cabré, M. S. Millán, N. K. Nishchal, R. Torroba, J. F. Barrera, W. He, X. Peng, A. Stern, Y. Rivenson, A. Alfalou, C. Brosseau, C. Guo, J. T. Sheridan, G. Situ, M. Naruse,

Optics EXPRESS

T. Matsumoto, I. Juvells, E. Tajahuerce, J. Lancis, W. Chen, X. Chen, P. W. H. Pinkse, A. P. Mosk, and A. Markman, "Roadmap on optical security," J. Opt. **18**(8), 083001 (2016).

34. B. I. Erkmen, "Computational ghost imaging for remote sensing applications," *IPN Prog. Rep.* 42–185 (2011).
35. W. Gong, C. Zhao, H. Yu, M. Chen, W. Xu, and S. Han, "Three-dimensional ghost imaging lidar via sparsity constraint," Sci. Rep. **6**(1), 26133 (2016).
36. B. I. Erkmen and J. H. Shapiro, "Signal-to-noise ratio of gaussian-state ghost imaging," Phys. Rev. A **79**(2), 023833 (2009).
37. M. P. Edgar, G. M. Gibson, and M. J. Padgett, "Principles and prospects for single-pixel imaging," Nat. Photonics **13**(1), 13–20 (2019).
38. Y. Wang, Y. Liu, J. Suo, G. Situ, C. Qiao, and Q. Dai, "High speed computational ghost imaging via spatial sweeping," Sci. Rep. **7**(1), 45325 (2017).
39. Z.-H. Xu, W. Chen, J. Penuelas, M. J. Padgett, and M.-J. Sun, "1000 fps computational ghost imaging using LED-based structured illumination," Opt. Express **26**(3), 2427–2434 (2018).
40. O. Katz, Y. Bromberg, and Y. Silberberg, "Compressive ghost imaging," Appl. Phys. Lett. **95**(13), 131110 (2009).
41. C. Zhao, W. Gong, M. Chen, E. Li, H. Wang, W. Xu, and S. Han, "Ghost imaging lidar via sparsity constraints," Appl. Phys. Lett. **101**(14), 141123 (2012).
42. W. Wang, X. Hu, J. Liu, S. Zhang, J. Suo, and G. Situ, "Gerchberg-saxton-like ghost imaging," Opt. Express **23**(22), 28416–28422 (2015).
43. W. Wang, Y. P. Wang, J. Li, X. Yang, and Y. Wu, "Iterative ghost imaging," Opt. Lett. **39**(17), 5150–5153 (2014).
44. D. Jin, W. Gong, and S. Han, "The influence of sparsity property of images on ghost imaging with thermal light," Opt. Lett. **37**(6), 1067–1069 (2012).
45. J. H. Shapiro and R. W. Boyd, "The physics of ghost imaging," Quantum Inf. Process. **11**(4), 949–993 (2012).
46. M. T. Mccann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," IEEE Sig. Process. Mag. **34**(6), 85–95 (2017).
47. K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Trans. Image Process. **26**(7), 3142–3155 (2017).
48. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR* 770–778 (2016).
49. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Proc. NIPS **1**, 1097–1105 (2012).
50. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Proc. ICML **37**, 448–456 (2015).
51. T. S. Ferguson, "An inconsistent maximum likelihood estimate," J. Am. Stat. Assoc. **77**(380), 831 (1982).
52. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv e-prints 1412.6980 (2014).
53. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**(11), 2278–2324 (1998).
54. C. Li, W. Yin, H. Jiang, and Y. Zhang, "An efficient augmented lagrangian method with applications to total variation minimization," Comput. Optim. Appl. **56**(3), 507–530 (2013).
55. https://github.com/abhi9716/handwritten-MNIST-digit-recognition.
56. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Process. **13**(4), 600–612 (2004).
57. E. Tajahuerce, V. Durán, P. Clemente, E. Irles, F. Soldevila, P. Andrés, and J. Lancis, "Image transmission through dynamic scattering media by single-pixel photodetection," Opt. Express **22**(14), 16945–16955 (2014).