Generative-adversarial-network–based ghost recognition

Yuchen He,[1] Yibing Chen,[1] Yuan Yuan,[1] Hui Chen,[1,*] Huaibin Zheng,[1] Jianxing Li,[2] and Zhuo Xu[1]

[1]*Electronic Materials Research Laboratory, Key Laboratory of the Ministry of Education and International Center for Dielectric Research, School of Electronic Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

[2]*School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

(Received 6 November 2021; accepted 11 July 2022; published 15 August 2022)

Target recognition technique plays an important role in many fields. The conventional methods perform recognition from images of scenes. However, noise or insufficient obtainable information would cause difficulty in image reconstruction, highly impeding the recognition. Ghost imaging (GI) can accomplish a recognition far below Nyquist limit under which a feasible image would not be reconstructed. However, recent GI recognition methods worked with targets of only 32×32 in resolution, and recognition for high-resolution targets with an extremely low sampling rate is still challenging. We here propose a target recognition method combining GI and generative adversarial network (GAN), which implements a recognition for a target of a 1024×1024 resolution at a sampling rate of 0.07%. GI samples spatial frequencies by illuminating a target with a set of random speckle patterns. The detected bucket signals (the intensities of the echo light from the target) are input into the GAN. The generator of the GAN is exploited to redraw the input bucket signals (detected in the high-resolution setup) to a low-dimension data (as if detected in a low-resolution setup). The discriminator can quickly classify the dimension-reduced data at a very low sampling rate. The method not only brings a way to high-resolution target recognition before image reconstruction, but also proposes an application of GI to circumvent its major challenge: a slow imaging speed due to the requirement of a large number of illumination patterns.

DOI: 10.1103/PhysRevA.106.023710

I. INTRODUCTION

Ghost imaging (GI) exploits the second-order correlation to reconstruct images, which is quite different than the traditional imaging methods [1–9]. GI utilizes a series of random speckle patterns to illuminate a target, and uses a bucket detector without spatially resolving the ability to collect echo intensity from the target. The correlation between the bucket signals and the illumination patterns recovers the image of the target. Due to its lensless imaging capability, turbulence-free imaging, and high detection sensitivity, GI has received lots of attention in recent years [10–18]. However, the poor trade-off between the image quality and imaging speed limits its application. There has been a large body of literature on how to solve this problem, or look for a compromise such as suitable application scenarios. Target recognition technique is an important approach in many fields [19–37]. Conventional recognition techniques retrieve the information from the image of a target, which means that the quality of the image decides the recognition accuracy. In some scenarios where the detection of signal-to-noise-ratios are very low or there exist strong turbulence or scattering, image reconstruction would suffer from low imaging quality, slow imaging speed or even failure of acquiring a feasible and faithful result. It consequently causes a slow recognition process or unreliable outcomes. Recently, Li *et al.* implemented object recognition with GI and a convolutional neural network [38]. Hadamard matrices were used as illumination patterns. Under a 32×32 resolution, the recognition worked well at a 9.77% sampling rate but was limited at 0.98%. He *et al.* performed handwritten digit recognition with GI and a deep neural network [39], where discrete cosine transform bases were used for illumination patterns. The method worked for targets at a 28×28 resolution under a sampling rate of 12.76%. These works demonstrated that GI can realize target recognition at a low sampling rate where images may not be reconstructed with a feasible quality. However, the low resolutions limit their application. High resolution at low sampling rate, for example 1024×1024, may be a challenge to the above methods. For instance, the resolution of Hadamard-based patterns is limited to 128×128 [40,41].

We here propose a recognition method based on GI and a generative adversarial network (GAN) [42]. This method implements a recognition for a target of a 1024×1024 resolution at a sampling rate of 0.07%. Under such a low sampling rate, GI fails to recover a feasible image. To perform detections at the high-resolution (the number of speckle cells on the object plane is also 1024×1024), we employed random speckle patterns. The bucket signals were input into a conditional generative adversarial network (CGAN) [43] for recognition. Although GAN is usually used for creating photos (to mix the spurious with the genuine) rather than recognition, we here exploit the generator in the GAN to convert the bucket signals to another data as if the "redrawn" data were obtained from a GI system at a resolution of 28×28 (both the target and detecting patterns are of 28×28). To fulfill the goal, we trained the discriminator of the CGAN with

*chenhui@xjtu.edu.cn

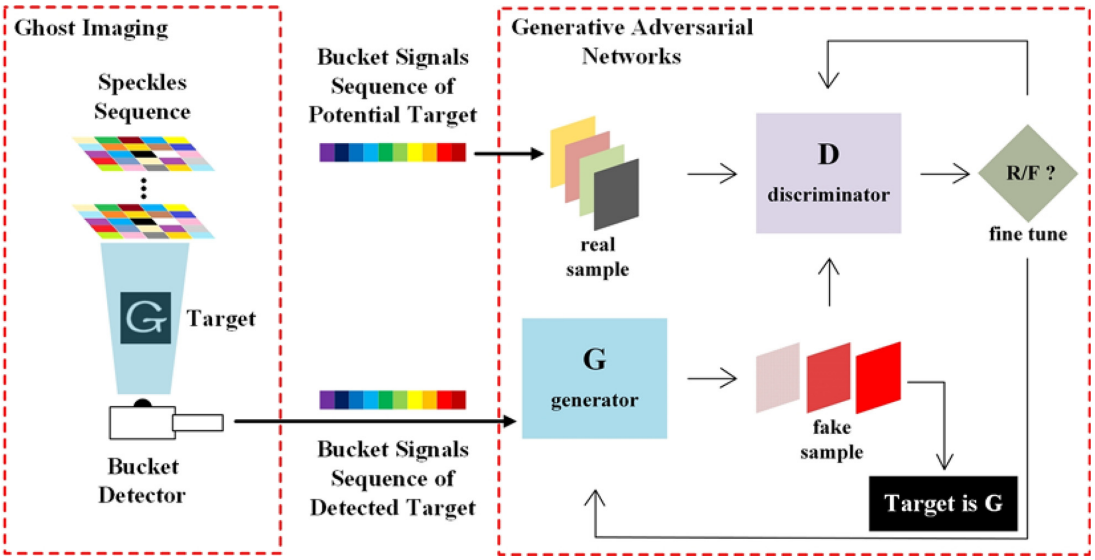

FIG. 1. Schematic diagram of the proposed method.

the data from the GI detection at 28×28. Because CGAN inherently has the classification ability, the CGAN becomes a suitable tool that works with GI for recognition. Handwritten targets recognition for numbers and letters are demonstrated at a 0.07% sampling rate for targets of 1024×1024. Further validation experiments that target recognition under different orientations and translational position are demonstrated. Finally, a physical experiment at distance of 20 meters is carried out on different targets. The resolution of the experimental GI system is 700×700 in a number of the speckle cells on the detection plane. Under a 0.02% sampling rate, the targets were well recognized as well. The rest of this paper is organized as follows. Section II presents a comprehensive introduction to generative-adversarial-networks-based ghost recognition. In Sec. III, the proposed method is demonstrated by extensive simulations and experiments. The paper is concluded in Sec. IV.

II. GAN-BASED GHOST RECOGNITION

A. Related method

GI illuminates an object with light whose intensity fluctuates temporally and spatially, which can be called time-varying speckle patterns. Using a bucket detector, one can measure the total light intensity reflected from the object, which represents how the object responses to a illumination pattern. After testing an object with thousands of different light patterns, GI recover the image by calculating the second-order correlation between the patterns and the corresponding bucket detectors, formulated as

$$g^{(2)} = \frac{\langle I_{\text{bucket}} \cdot I(x,y)\rangle}{\langle I_{\text{bucket}}\rangle \langle I(x,y)\rangle}, \tag{1}$$

where $g^{(2)}$ denotes the second-order correlation function, $\langle\cdot\rangle$ is the ensemble average, I_{bucket} is the bucket signal and $I(x,y)$ is the reference signal. Note that, either the traditional GI or computational ghost imaging (CGI) [10] requires that the illumination patterns on the object must be known. The difference is CGI uses preset patterns avoiding using an imager (such as CCD) to real-time observing the time-varying speckle patterns. The reconstruction based on the formula above needs to illuminate a large number of speckles to obtain an ideal result.

Recently, artificial intelligence methods have been increasingly introduced into GI. GAN, which has attracted much attention in the field of deep learning in recent years, is also considered to develop the application scenarios of GI. The basic idea of GAN can be expressed as: The generator network G generates the forged image according to the input, and the discriminator network D judges whether the image is true or false. When discriminator D cannot judge whether the image is a real image or generated by generator G, the model training is completed. The objective function of GAN can be expressed as

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log_{10} D(x)] + E_{z\sim P_z(x)}[\log_{10}(1 - D(G(z)))], \tag{2}$$

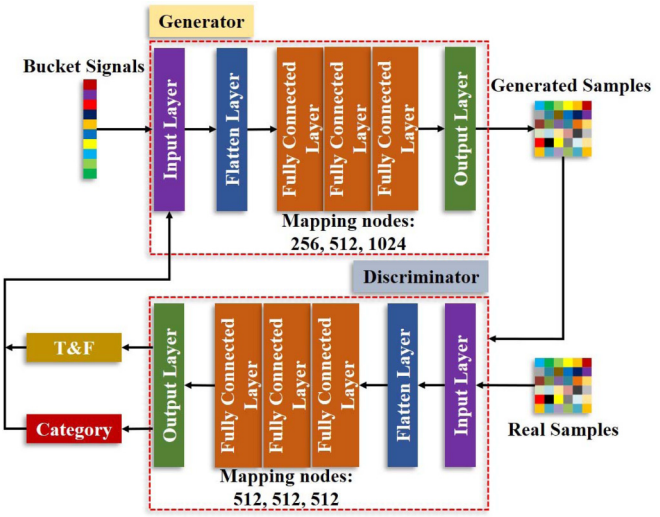

FIG. 2. Network architecture of the proposed method.

FIG. 3. Handwritten targets in our experiment.

where x represents the real image, and z represents the input of the generator G. V denotes the objective function, E denotes the expectation, P_{data} denotes the distribution of the training samples, and P_z denotes the distribution of the generator inputs. The right side of the above formula is divided into two terms. The first term indicates the probability of discriminator D judging the real image as true, the closer to 1, the better. In the second term, $G(z)$ represents the image generated by generator G, the task of generator G is to make the image generated close to the real image, the better the performance of G is, the greater the $D(G(z))$. The second term represents the probability of discriminator D judging whether the image generated by generator G is a real image, the better the performance of D, the smaller the $D(G(z))$, and there is a process of game competition. Although GAN is effective, it is too free to control the output of the generator. To solve this problem, Montreal proposes generative adversarial networks with constraints, named CGAN. In CGAN, conditional variable is added to generator and discriminator to guide the data generation process. In this paper, CGAN is employed to match bucket signal sequence and target category.

B. Principle

Combining CGI and CGAN, we propose an improved target recognition technique based on bucket signal arrays. Figure 1 shows the architecture of the proposed method.

Different speckle patterns are utilized to illuminate target at different positions in the scene, respectively, and the echo signal is continuously received by a bucket detector. After multiple samplings, a bucket signal sequence is formed. For the generator, the bucket signal sequence is regarded as one sample. For the discriminator, the bucket signal sequences of various targets at different positions are regarded as real samples. Obviously, different bucket signal sequences represent different labels. Therefore, the target in testing will be recognized quickly if the speckle sequence in the training stage is still utilized to illuminate the target. The objective function of this work consists of two parts: log likelihood of the real bucket signal sequence samples L_s and log likelihood of real bucket signal sequence class is labeled L_c. The goal of discriminator is to find a suitable value to maximize $L_s + L_c$, and the goal of generator is to find a suitable value to maximize $L_s - L_c$. When the discriminator cannot distinguish whether the input data is real or not, the network reaches equilibrium, and the probability distribution of output is 0.5. By introducing the bucket signal sequence class label as a condition variable, the uncontrollable shortcoming of the traditional GAN training process is improved. Finally, the label control generator of the bucket signal sequence can generate the corresponding category of bucket signal sequence data. Simultaneously, the discriminator also learns the data feature distribution of real samples in the process of confrontation to judge whether the input is real or generated.

C. Network architecture

In this paper, the construction of CGAN is based on multilayer perceptron (MLP), which is a deep artificial neural network. A MLP neural network generally contains multiple perceptrons, including an input layer to receive input data and an output layer to classify or predict input data. There are usually many hidden layers between the input layer and the output layer. The calculation process of MLP is completed in the hidden layer. The schematic diagram of the network architecture of the proposed method is shown in Fig. 2.

In this work, the generator network includes six layers (one input layer, one flatten layer, three fully connected layers, and one output layer), which is used to generate images of the same type as the target. According to the number of bucket signals, the input is tiled into 784 nodes in the flattened layer. The number of nodes in a fully connected layer third to fifth is 256, 512, and 1024, respectively. The output layer reconstructs the output of the previous layer to generate an image, which is the same as the input. Similarly, the discriminator is also based on MLP, and the network includes six layers (one input layer, one flattened layer, three fully connected layers, and one output layer), which used to

TABLE I. Recognition results for letter targets.

Samples	5000				10000				20000				60000			
Epoch	500	1000	2000	5000	500	1000	2000	5000	500	1000	2000	5000	500	1000	2000	5000
Target_A	90%	100%	100%	100%	100%	100%	100%	100%	100%	70%	90%	100%	100%	100%	80%	100%
Target_B	80%	100%	100%	100%	100%	100%	100%	100%	80%	100%	100%	100%	90%	100%	100%	100%
Target_C	100%	100%	100%	80%	70%	90%	100%	100%	100%	90%	90%	100%	80%	100%	100%	100%
Target_D	100%	100%	90%	100%	100%	100%	100%	100%	90%	100%	100%	100%	100%	100%	90%	100%
Target_E	100%	100%	100%	100%	90%	100%	100%	90%	100%	90%	100%	100%	100%	100%	80%	100%
Target_F	100%	100%	100%	80%	100%	100%	100%	100%	80%	100%	100%	100%	90%	100%	100%	100%
Target_G	100%	100%	90%	100%	100%	100%	100%	100%	80%	100%	90%	100%	100%	90%	100%	100%
Target_H	100%	90%	100%	100%	100%	100%	100%	100%	80%	90%	100%	100%	70%	100%	100%	100%
Target_I	100%	80%	100%	100%	100%	100%	100%	100%	90%	100%	100%	100%	100%	100%	100%	100%
Target_J	90%	90%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	80%	100%	100%

TABLE II. Recognition results for number targets.

Samples	5000				10000				20000				60000			
Epoch	500	1000	2000	5000	500	1000	2000	5000	500	1000	2000	5000	500	1000	2000	5000
Target_0	90%	100%	100%	100%	60%	100%	100%	100%	100%	70%	100%	100%	100%	100%	100%	100%
Target_1	100%	70%	100%	100%	100%	80%	100%	100%	100%	100%	70%	100%	90%	100%	100%	90%
Target_2	90%	100%	100%	100%	100%	100%	100%	90%	100%	100%	100%	100%	70%	100%	100%	100%
Target_3	100%	80%	100%	100%	100%	100%	80%	100%	100%	100%	100%	100%	100%	90%	100%	100%
Target_4	100%	100%	100%	100%	100%	100%	90%	90%	100%	80%	100%	100%	60%	100%	80%	100%
Target_5	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Target_6	100%	90%	100%	100%	100%	100%	100%	100%	100%	70%	100%	100%	100%	100%	100%	100%
Target_7	90%	100%	80%	90%	90%	100%	100%	100%	90%	80%	100%	100%	100%	70%	80%	100%
Target_8	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Target_9	90%	100%	100%	100%	100%	100%	100%	100%	100%	90%	100%	100%	100%	100%	90%	100%

identify the category of the input image. The third to fifth layer is a fully connected layer, and the number of nodes is 512. The sixth layer uses a sigmoid function to normalize the output of the previous layer. The output of the sixth layer consists of two parts, the authenticity of the training sample and the category of the target. The cross entropy between the output of the training samples and the label is taken as the training loss function. The output of the training samples is compared with the label, and the optimizer Adam is used to control the change of the learning rate. Finally, the operation to reduce the loss is defined for the confrontation training.

The steps can be summarized as:

(1) Prepare training and testing sets. According to CGI mechanism, using a set of random speckle sequence to illuminate potential targets, and the bucket signal sequence collected as a training set.

(2) Build network. Based on tensorflow-gpu 1.13 (keras version 2.1.5), three models are built: generator model, discriminator model, and confrontation training model.

(3) Train network. In the loop, the images in the training set and their corresponding categories are input, and the generator and discriminator model are trained simultaneously.

(4) Test network. The same speckle sequence in training is used to illuminate, and the bucket signal sequence is sent to the network to recognize the target.

III. EXPERIMENTAL RESULTS

A. Training settings

This work originated from a realistic scenario: recognition for license plates or certain types of objects in the viewfinder. It requires detection under a certain high resolution and a fast recognition speed. Since the license numbers and letters are of a fixed shape, we did not intend to have the method recognize different fonts. Instead, we are concerned about if the recognition is insensitive to the position of a detected text. Under the architecture of CGI, the training data comes from bucket signal sequence after multiple samplings of target based on a set of fixed random speckle sequence. In this work, the size of speckle and target is both 1024*1024. 784 echoes are used to construct a bucket signal sequence, and the sampling rate is 0.07%. Note that, with such a low sampling rate, GI failed to recover a feasible image. The output size of the generator and the input size of the discriminator is 28*28. The targets in our experiment are handwritten letters and numbers. Letter targets include 10 categories of "A,B,C,D,E,F,G,H,I,J", and number targets include 10 categories of "0,1,2,3,4,5,6,7,8,9". We trained four networks using 5000, 10 000, 20 000, and 60 000 samples, and each category of target contains 500, 1000, 2000, and 6000 in four networks, respectively. Meanwhile, 500, 1000, 2000, and 5000 epochs are performed in each network. Each object in our training set contains different

TABLE III. Testing results on letter and number targets with same speckles sequence.

Samples	5000		10000		Samples	5000		10000	
Epoch	2000	5000	2000	5000	Epoch	2000	5000	2000	5000
Target_A	100%	100%	100%	100%	Target_0	90%	90%	100%	100%
Target_B	100%	100%	80%	100%	Target_1	100%	50%	100%	100%
Target_C	80%	100%	60%	100%	Target_2	100%	100%	100%	100%
Target_D	100%	90%	100%	100%	Target_3	100%	100%	100%	100%
Target_E	90%	100%	100%	90%	Target_4	100%	100%	100%	100%
Target_F	100%	80%	100%	100%	Target_5	70%	100%	100%	100%
Target_G	60%	100%	100%	100%	Target_6	100%	100%	70%	100%
Target_H	100%	100%	100%	100%	Target_7	100%	60%	100%	90%
Target_I	100%	100%	100%	100%	Target_8	100%	100%	100%	100%
Target_J	100%	80%	100%	100%	Target_9	100%	100%	100%	100%

TABLE IV. Testing results on noise effects.

	Environmental			System		
	6 dB	0 dB	−6 dB	8 dB	−5 dB	−10 dB
Target_A	100%	90%	80%	100%	90%	80%
Target_B	90%	80%	70%	90%	80%	70%
Target_C	100%	100%	100%	100%	100%	100%
Target_D	100%	100%	90%	100%	100%	90%
Target_E	80%	80%	70%	100%	90%	80%
Target_F	90%	80%	70%	100%	100%	80%
Target_G	90%	90%	80%	100%	100%	90%
Target_H	100%	80%	80%	100%	100%	90%
Target_I	100%	100%	100%	100%	100%	100%
Target_J	100%	90%	90%	90%	80%	70%

positions, through training to simulate the different positions of objects in the scene in practical application. For example, in the case of 5000 samples for letter targets recognition in this work, we make each object in 100 different positions for training, and each position contains five samples. In other cases, we expand the number of samples at each position. In testing, we place objects in other positions in the scene. The GPU used for training in this work was a NVIDIA GeForce GT 710. The shortest training cost (5000 samples and 500 epochs) was 1.56 min and the longest training cost (60000 samples and 5000 epochs) was 14.9 min. Figure 3 shows the handwritten targets for testing in our experiment.

B. Results on letters and numbers targets

Firstly, we test the proposed method on the letter handwriting targets set. Ten categories of objects, respectively, represent the bucket signal sequence formed by 10 letter targets after 784 times of illumination. Then, we randomly select 10 samples from the testing set (containing 100 samples that positions not in the training set). It took 3.5 s to identify 10 targets, with an average of 0.35 s for one target. The recognition results are shown in Table I.

Table I shows that with the increase of the number of samples, the recognition accuracy eventually reaches an ideal level. Similarly with letter targets, we test the proposed method on four networks with different sizes and epochs, and Table II shows the results.

C. Same random speckles sequence

In the experiments above, two different speckle sequences are used for training and testing on letter and number targets. In this section, we try to train letters and numbers targets with one set of random speckle sequence, and so do the test. Similarly, the size of speckle is 1024*1024, and these 784 random speckles are different and spatially independent. Then, the bucket signal sequence formed by 784 illuminations is taken as one sample. We use 5000 and 10 000 samples to train two networks, each with 2000 and 5000 epochs, respectively. Table III lists the testing results, it can be seen that the proposed method achieves promising performance.

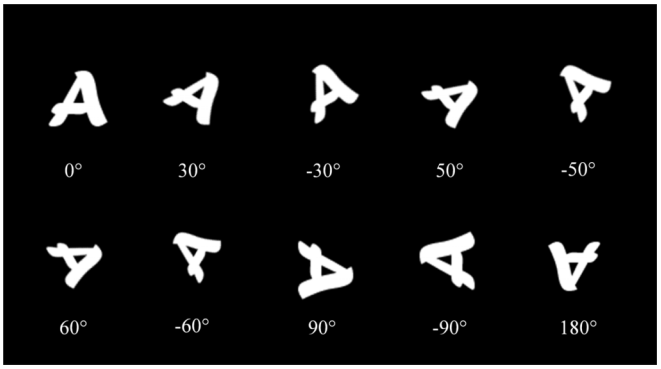

FIG. 4. Different orientations of "A".

D. Results on noise effects

Furthermore, we demonstrate noise effects (including environmental noise and system noise) on the proposed method. The targets are handwritten letters "A,B,C,D,E,F,G,H,I,J". The training set contains 60 000 samples and the epoch is 5000. We add different levels of noises to the illumination path as the environmental noise. In addition, we add different levels of noises to the bucket signals sequence as the system noise. We demonstrate the proposed method in these two noise effects. Demonstration results are shown in Table IV.

It can be seen that the recognition rate decreases in the presence of environmental noise and system noise, except for simple and intuitive objects such as "C" and "I".

E. Results on different orientations

In real scene, the orientation of objects are often changing. Target needs to be successfully recognized in different orientations. Therefore, we try this task with the proposed method. Figure 4 shows the different orientations of letter target "A".

We test the recognition performance of 10 different orientations, including 0°, 30°, −30°, 50°, −50°, 60°, −60°, 90°, −90°, 180°. Table V shows the results of different orientations of letter target A; it can be seen that the recognition rate of the proposed method is considerable regardless of the target orientation.

TABLE V. Testing results on different orientations of target A.

Samples	10000		20000	
Epoch	2000	5000	2000	5000
Orientation_0	100%	90%	100%	100%
Orientation_30	100%	100%	100%	100%
Orientation_−30	100%	100%	80%	100%
Orientation_50	100%	100%	100%	100%
Orientation_−50	100%	100%	100%	100%
Orientation_60	100%	100%	100%	100%
Orientation_−60	100%	100%	100%	100%
Orientation_90	100%	100%	100%	100%
Orientation_−90	100%	100%	100%	90%
Orientation_180	100%	100%	80%	100%

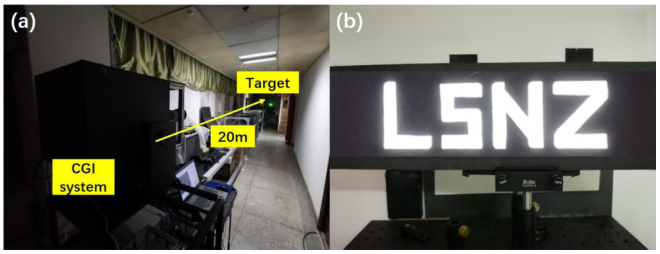

FIG. 5. Physical experiment. (a) Experimental light path. (b) Target LSNZ.

F. Physical experiment

Finally, we demonstrated a physical experiment at a distance of 20 meters. Figure 5 shows the physical experimental light path. The physical experiment is based on the CGI architecture, and the targets are reflective letters LSNZ. Each letter is of $\sim 7 \times 7$ cm, which is placed 20 meters away from the CGI system. A laser beam with the wavelength of 532 nm is modulated by a rotation ground glass and then illuminated onto the target. The laser beam is 1 mm in diameter when it hit on the rotating ground glass. We utilize field programmable gate array (FPGA) to control the rotation position of ground glass to obtain a specific speckle sequence. The scattered light is then collimated by a telescope system, which is equivalent to a lens with a focus length of 200 mm, forming an approximately parallel beam of 10 cm wide. The average size of speckle grains on the object plane is approximately 532 nm*200 mm/1 mm = 0.1 mm, so the resolution of a so-generated speckle pattern is (7 cm/0.01 cm)*(7 cm/0.01 cm) $= 700 \times 700$. The four targets are illuminated independently, and the echo signals are also recorded independently. The echo signal is recorded by a lensless CCD for the light intensity collection.

In training, the bucket signal sequence contains 100 echoes for each target (the sampling rate is 0.02%). The training set contains 24 000 samples (each target contains 6000 samples), and we trained 1000, 6000, and 8000 epochs, whose training times were 3.08, 17.75, and 23.55 min, respectively. In testing, the speckle sequence in training step is employed to illuminate targets. The bucket signal sequences are input to the trained CGAN for recognition at the same time. The recognition results are shown in the Table VI.

Table VI shows that the proposed method can also recognize different targets in the physical experiment, and the recognition rate becomes better with the increase of epoch. It took 0.12 s to identify four targets, with an average of 0.03 s for one target.

TABLE VI. Physical experimental results.

	Epoch		
Target	1000	6000	8000
Target_L	60%	70%	100%
Target_S	90%	100%	100%
Target_N	50%	80%	100%
Target_Z	90%	100%	100%

IV. CONCLUSIONS

In this paper, we proposed a target recognition method combining GI and GAN, which implements a recognition for a target of a 1024×1024 resolution at a sampling rate of 0.07%. Under the framework of GI, we utilize a set of random speckle sequence to illuminate targets, and the bucket detector is employed to received echo signals continuously. Then, the bucket signals are formed and considered as a sample of CGAN. The generator of the CGAN is exploited to redraw the input bucket signals (detected in the high-resolution setup) to a low-dimension data (as if detected in a low-resolution setup). After training, the discriminator can quickly classify the dimension-reduced data at a very low sampling rate. Extensive experiments show that the proposed method achieves promising performance at very low sampling rate on different data sets. Finally, a physical experiment at a distance of 20 m is carried out to demonstrate the proposed method on different targets. The method not only brings a way to high-resolution target recognition before image reconstruction, but also proposes an application of GI to circumvent its major challenge: a slow imaging speed due to the requirement of a large number of illumination patterns.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 61901353), 111 Project of China (Grant No. B14040), Shaanxi Key Research and Development Project (Grant No. 2021GXLH-Z-012) and the Fundamental Research Funds for the Central Universities (Grant No. xhj032021005).

[1] T. B. Pittman, Y. H. Shih, D. V. Strekalov, and A. V. Sergienko, Phys. Rev. A **52**, R3429(R) (1995).
[2] A. F. Abouraddy, B. E. A. Saleh, A. V. Sergienko, and M. C. Teich, Phys. Rev. Lett. **87**, 123602 (2001).
[3] R. S. Bennink, S. J. Bentley, and R. W. Boyd, Phys. Rev. Lett. **89**, 113601 (2002).
[4] A. Gatti, E. Brambilla, M. Bache, and L. A. Lugiato, Phys. Rev. Lett. **93**, 093602 (2004).
[5] R. S. Bennink, S. J. Bentley, R. W. Boyd, and J. C. Howell, Phys. Rev. Lett. **92**, 033601 (2004).
[6] A. Valencia, G. Scarcelli, M. D'Angelo, and Y. Shih, Phys. Rev. Lett. **94**, 063601 (2005).
[7] F. Ferri, D. Magatti, A. Gatti, M. Bache, E. Brambilla, and L. A. Lugiato, Phys. Rev. Lett. **94**, 183602 (2005).
[8] L. Basano and P. Ottonello, Appl. Phys. Lett. **89**, 091109 (2006).
[9] G. Scarcelli, V. Berardi, and Y. Shih, Phys. Rev. Lett. **96**, 063602 (2006).
[10] J. H. Shapiro, Phys. Rev. A **78**, 061802(R) (2008).

[11] R. Meyers, K. S. Deacon, and Y. Shih, Phys. Rev. A **77**, 041801(R) (2008).
[12] O. Katz, Y. Bromberg, and Y. Silberberg, Appl. Phys. Lett. **95**, 131110 (2009).
[13] R. E. Meyers, K. S. Deacon, and Y. Shih, Appl. Phys. Lett. **98**, 111115 (2011).
[14] D. Pelliccia, A. Rack, M. Scheel, V. Cantelli, and D. M. Paganin, Phys. Rev. Lett. **117**, 113902 (2016).
[15] Y. He, G. Wang, G. Dong, S. Zhu, H. Chen, A. Zhang, and Z. Xu, Sci. Rep. **8**, 6469 (2018).
[16] T. Bian, Y. Yi, J. Hu, Y. Zhang, Y. Wang, and L. Gao, Sci. Rep. **10**, 12149 (2020).
[17] H. Wu, R. Wang, G. Zhao, H. Xiao, D. Wang, J. Liang, X. Tian, L. Cheng, and X. Zhang, Opt. Express **28**, 3846 (2020).
[18] Z. Zhang, C. Wang, W. Gong, and D. Zhang, Appl. Opt. **60**, 3732 (2021).
[19] S. Xiao, G. Guo, Z. Zhuang, and M. Luo, in *Proceedings of National Aerospace and Electronics Conference (NAECON'94)* (IEEE, Dayton, 1994), pp. 80–85.
[20] S. Der and R. Chellappa, IEEE Trans. Image Processing **6**, 92 (1997).
[21] H. Ren, Q. Du, J. Wang, C.-I. Chang, J. Jensen, and J. Jensen, IEEE Trans. Aerosp. Electron. Syst. **42**, 1372 (2006).
[22] C. Gronwall, F. Gustafsson, and M. Millnert, IEEE Trans. Image Processing **15**, 3400 (2006).
[23] D.-Y. Chen, K. Cannons, H.-R. Tyan, S.-W. Shih, and H.-Y. Liao, IEEE Trans. Multimedia **10**, 1578 (2008).
[24] K. M. Iftekharuddin, IEEE Trans. Neural Networks **22**, 906 (2011).
[25] Z. Guo and Z. J. Wang, IEEE Trans. Multimedia **15**, 621 (2013).
[26] J. A. Garzon-Guerrero, D. P. Ruiz, and M. C. Carrion, IEEE Trans. Antennas Propag. **61**, 4881 (2013).
[27] X. Li, X. Meng, X. Yang, Y. Yin, Y. Wang, X. Peng, W. He, G. Dong, and H. Chen, IEEE Photonics J. **8**, 1 (2016).
[28] Z. Yang, J. Deng, and A. Nallanathan, IEEE Sens. J. **16**, 5671 (2016).
[29] A. I. Karjalainen, R. Mitchell, and J. Vazquez, in *2019 Sensor Signal Processing for Defence Conference (SSPD)* (IEEE, Brighton, 2019).
[30] X. Bai, X. Zhou, F. Zhang, L. Wang, R. Xue, and F. Zhou, IEEE Trans. Geosci. Electron. **57**, 9912 (2019).
[31] N. M. Nasrabadi, IEEE Trans. Aerosp. Electron. Syst. **55**, 2687 (2019).
[32] X. Jiang, Y. Lai, S. Wang, and Y. Song, IEEE Access **8**, 143033 (2020).
[33] L. Tao, Y. Zhou, X. Jiang, X. Liu, and Z. Zhou, IEEE Geosci. Remote Sens. Lett. **18**, 1776 (2020).
[34] X. Yang, X. Nan, and B. Song, IEEE Trans. Geosci. Electron. **58**, 3667 (2020).
[35] C. Mao, L. Huang, Y. Xiao, F. He, and Y. Liu, IEEE Access **9**, 39608 (2021).
[36] Y. Guo, L. Du, D. Wei, and C. Li, IEEE J. Select. Topics Appl. Earth Observation. Remote Sens. **14**, 716 (2021).
[37] L. M. Ngô, S. Karaoğlu, and T. Gevers, IEEE Trans. Multimedia **24**, 377 (2022).
[38] J. Li, M. Le, J. Wang, W. Zhang, B. Li, and J. Peng, Appl. Phys. B **126**, 166 (2020).
[39] X. He, S. Zhao, and L. Wang, Chin. Phys. B **30**, 054201 (2021).
[40] M. Sun, M. Tong, L. Wang, E. Matthew, P. Miles, and R. Neal, Sci. Rep. **7**, 3464 (2017).
[41] W. K. Yu, Sensors **19**, 4122 (2019).
[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *Advances in Neural Information Processing Systems*, Vol. 27 (Curran Associates, Inc., Montreal, 2014), pp. 2672–2680.
[43] M. Mirza and S. Osindero, arXiv:1411.1784 (2014).