Multiview Generative Adversarial Network and Its Application in Pearl Classification

Qi Xuan, *Member, IEEE*, Zhuangzhi Chen, Yi Liu, *Member, IEEE*, Huimin Huang, Guanjun Bao, *Member, IEEE*, and Dan Zhang

***Abstract*—This paper focuses on automatic pearl classification by adopting deep learning method, using multiview pearl images. Traditionally, in order to get a satisfying classification result, we need to collect a huge number of labeled pearl images, which however is expensive in industry. Fortunately, generative adversarial network (GAN) was proposed recently to effectively expand training set, so as to improve the performance of deep learning models. We thus propose a multiview GAN (MV-GAN) to automatically expand our labeled multiview pearl images, and the expanded data set is then used to train the multistream convolutional neural network (MS-CNN). The experiments show that the utilization of images generated by the MV-GAN can indeed significantly reduce the classification error of the basic MS-CNN (up to 26.71%, relatively), obtaining the state-of-the-art results. More interestingly, it can also help the MS-CNN resist the brightness disturbance, leading to more robust classification.**

***Index Terms*—Convolutional neural network (CNN), deep learning, three-dimensional (3-D) object classification, fine-grained classification, generative adversarial networks (GANs), intelligent manufacturing, pearl classification.**

I. Introduction

NOWADAYS, deep learning technologies are being widely applied to enhance the intelligence and automation of industrial manufacturing processes, achieving quite impressive results [1]–[7]. In pearl industry, most pearl producing companies rely mainly on manual classification, i.e., pearls are classified by experienced professionals, according to their size, texture, shape, luster, and other characteristics. Such manual work is not only repetitive, monotonous, and inefficient, but may also lead to ophthalmic diseases due to the strong reflection on the surface of pearls. Recently, an automatic pearl classification machine [8] was developed based on computer vision technologies. The machine mainly consists of feeding mechanism, delivering mechanism, vision-based detection device, and classification mechanism, as well as the framework and the outer casing. Although any part is important to make the machine run smoothly, its computer vision plays the key role in determining the performance of pearl classification.

Generally, multiple views of pearls should be considered simultaneously in pearl classification, since it is quite challenging as a typical fine-grained recognition task [9]–[11], where the visual differences among the categories are very small and thus can be overlooked if just one view is considered. Therefore, an high definition (HD) camera, a ring-shaped light source, and four mirrors are integrated into a single vision-based detection device to collect the images of the pearl from five different viewing angles at one shot. One can certainly use the traditional digital image processing methods to capture the multiple features of pearl images, e.g., using gray-level co-occurrence matrix [12] or local binary patterns [13] to capture the textural features, and then adopt some traditional machine learning methods, such as support vector machine [14] and back-propagation neural network [15], to realize pearl classification. However, designing appropriate hand-crafted features often relies on expert knowledge, making it expensive and inefficient.

Recently, deep learning models, especially the convolutional neural network (CNN) [16], have been widely used in image classification. CNN can automatically generate useful features and thus save huge labors, meanwhile they achieve the state-of-the-art results in many tasks. In pearl industry, according to the multiview pearl images, a multistream CNN, namely MS-CNN, was proposed to realize pearl classification of high accuracy [8]. However, to further improve the performance of MS-CNN, it requires much more labeled pearl images. In other words, more professionals are needed to manually annotate pearl images from different views based on the same standard, which is relatively expensive and may delay the real industrial application of the pearl classification machine.

An alternative way is to generate more labeled pearl images automatically. Fortunately, generative adversarial network (GAN) [17] has been developed recently to generate diverse labeled images based on the training set. Generally, a GAN contains two deep neural networks, one is the generator and the other is the discriminator. The generator tries to learn the

Manuscript received May 6, 2018; revised July 22, 2018 and September 11, 2018; accepted November 27, 2018. Date of publication December 13, 2018; date of current version May 31, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61572439 and Grant 61873241, and by the Natural Science Foundation of Zhejiang Province under Grant LR19F030001. *(Corresponding author: Yi Liu.)*

Q. Xuan, Z. Chen, H. Huang, and D. Zhang are with the Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China, and also with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: xuanqi@zjut.edu.cn; zz-chen001@qq.com; saffronhuang@163.com; jason_zhang19850626@hotmail.com).

Y. Liu and G. Bao are with the College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: yliuzju@zjut.edu.cn; gjbao@zjut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2018.2885684

distribution of the training set, and then generates fake images that match the distribution of the training set by input noise, while the discriminator tries to determine whether the generated images are fake or real. By alternately training the two networks continuously, the generator can finally generate an image that matches the distribution of the training set, perfectly satisfying the requirement to expand the pearl training set. However, the original GAN may generate pearl images too wildly. Meanwhile, it can only generate one image at a time, and these independent images cannot be directly used in the MS-CNN, since there are some restrictions among the multiview images of a single pearl.

In order to overcome these insufficiencies, we propose the multiview GAN, namely MV-GAN, based upon the deep convolutional GAN (DCGAN) [18] and the conditional GAN (CGAN) [19], to successfully generate multiview images of pearls with labels so as to efficiently expand the training set of pearl classification. Specifically, we make the following contributions.

1) First, we propose the MV-GAN as a new deep learning framework to generate multiview images of pearls. In particular, here we superimpose multiview images of each pearl into the image channel dimension in the order of top, left, right, main, and rare views, forming the multiview input data.
2) Second, the MV-GAN framework integrates the network structure of DCGAN and the way of label training in the CGAN. As a result, the discriminator has four convolution layers and the generator has four deconvolution layers. Meanwhile, the pearl labels are used to limit the training of each layer and the freedom to generate images. Then, we use the multiview images of pearls to train the MV-GAN and use the trained MV-GAN to generate multiview images so as to expand the training set.
3) Third, we retrain the MS-CNN by the expanded data set. The experiments validate that the MS-CNN can be indeed further improved, especially when the original training set is relatively small. Moreover, the MV-GAN can also help the MS-CNN resist the disturbance introduced by the environment, e.g., the brightness disturbance. These indicate that our MV-GAN framework has potential to be used in pearl industry to improve pearl classification accuracy and robustness.
4) Fourth, the MV-GAN can be used to generate the multiview images of various objects. It thus has potential to be applied in many other three-dimensional (3-D) object classification [20]–[23].

The rest of this paper is organized as follows. Section II presents the related work in the areas of pearl classification, GAN, and its applications. Section III briefly introduces the mechanism of the pearl classification machine and the collected pearl image data set. Then, the MV-GAN method is proposed in Section IV, based on which multiview pearl images can be automatically generated. These images are further used to update the classifier to improve the classification accuracy. After that, the MV-GAN is applied on pearl classification in Section V, and the experiments show that it can significantly enhance the MV-CNN, achieving the state-of-the-art performance. The paper is finally concluded in Section VI.

II. Related Work

In this section, we present the recent studies related to pearl classification, GAN, and its applications.

A. Pearl Classification

Classification is one of the most common tasks in machine learning. In industry, various classification tasks are usually based on computer vision technologies. Specifically, in pearl industry, classification of pearls based on their shape, texture, color, etc., is important and largely determines the following pearl processing, i.e., whether to put a pearl into a necklace of particular style or just use it to produce pearl powder.

Although automatic pearl classification is important and seems feasible by using computer vision technologies, there are only few studies in this area. Li *et al.* [24] proposed a pearl shape classification method, which classifies pearls mainly based on the shape features of pearl images. Meanwhile, Li *et al.* [25] also proposed a pearl classification method based on pearl luster degree, such as hue, saturation, and luminance. However, such methods largely rely on expert knowledge, making them relatively expensive and less adaptive to the different requirements from the pearl market.

Recently, deep learning methods, especially CNN, have achieved great success in computer vision, and various CNN models, such as AlexNet [26], VGG [27], GoogleNet [28], DenseNet [29], ResNet [30], and so on, were proposed. Xuan *et al.* [8] designed an automatic pearl classification device to collect multiview images of pearls and further proposed MS-CNN to automatically classify pearls. This method can make full use of the information of different viewing angles, and thus obtain impressive accuracy in pearl classification.

B. Generative Adversarial Networks

GAN is a kind of deep generation network, with the main goal to learn the distribution of the given training data and then to generate the data of similar distribution. Typically, it is an unsupervised learning model. GAN was first proposed by Goodfellow *et al.* [17], with the basic framework including a generation model and a discrimination model. The most prominent character of GAN is using adversarial training method. Due to the existence of discrimination model, the generation model can learn to approximate the given data without prior knowledge, which is difficult for many traditional generation models.

Radford *et al.* [18] summarized several important rules to design GAN structure and train CNN, and further proposed the DCGAN, by integrating convolution operation into GAN. This makes it more stable and controllable to train the GAN, and thus improve the quality of the generated images. Learning method is considered to be related to the stability of training a model. Mirza and Osindero [19] adopted the supervised learning method in GAN, instead of unsupervised learning, and proposed the CGAN. They introduced the conditional variables into the

generation model and discrimination model, in order to control the extent of generation. Denton *et al.* [31] proposed Laplacian generative adversarial networks (LAPGAN), which uses sequential learning to gradually improve the generating quality and the image resolution. Im *et al.* [32] introduced generative recurrent adversarial networks (GRAN), which combines GAN and long short-term memory (LSTM), allowing each step of learning and generation to make full use of the results produced in the previous step. Chen *et al.* [33] proposed InfoGAN, which decomposes the input noise into latent code and independent noise to learn interpretable information, so as to obtain different features of the generated images by using different encoding methods. In addition, there were also some studies trying to improve GAN by utilizing different measures of data similarity, such as Wasserstein generative adversarial networks (WGAN) [34] and Wasserstein generative adversarial networks with gradient penalty (WGAN-GP) [35].

C. Applications of GAN

As an outstanding generation method and ingenious training framework, GAN was widely applied in computer vision and natural language processing (NLP). In computer vision, it was adopted to realize single image super-resolution [36], image-to-image translation [37], style conversion [38]–[40], image inpainting [41], and so on. In NLP, it was used to realize the generation of dialogues [42], text-to-image translation [43], and machine translation [44].

Besides, GAN was also applied in data enhancement. Unlike the traditional methods that use manual rotation, deformation, and synthesis of images to expand the data set, the main issue in applying GAN is to relate the generated images with the training images of particular labels, so that these generated images can be further used in the training process to update the model. Zheng *et al.* [45] combined the unlabeled samples generated by DCGAN with the original data set using a semisupervised learning method, which slightly improved the performance of the person reidentification task. Wang *et al.* [46] adopted deep CNN and bidirectional recurrent neural networks [47] combined with LSTM [48] to perform feature learning and sequence labeling on the generated license plate, and then used real images to fine tune the models, which can improve the accuracy of license plate recognition. In addition, GAN was also used to discover drugs [49], generate new chemical formulas [50], detect cancer [51], and so on.

In this paper, we take DCGAN [18] and CGAN [19] as references. By integrating MS-CNN with GAN, we propose a new GAN framework, namely MV-GAN, to generate multiview images of pearls with labels. These generated images are further used to improve the performance of pearl classification. To the best of our knowledge, this is the first work to apply GAN to generate multiview images and further classify pearls in the industrial field.

III. Mechanism and Data Set

Here, we briefly introduce the main parts and functions of the pearl classification machine to make the readers better understand its mechanism, and also to make the paper self-contained.

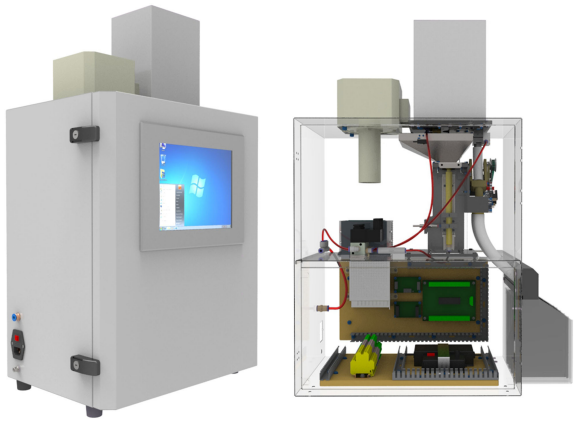

Fig. 1. Appearance (left) and the internal structure (right) of pearl classification machine.

Our pearl classification machine is shown in Fig. 1, which contains four parts: feeding mechanism; delivering mechanism; vision-based detection device; and classification mechanism. When the machine is started, the pearls are automatically delivered one by one to the vision-based detection device as the main equipment to obtain multiview pearl images. The detection device includes a ring-shaped light emitting diode (LED) light source, a camera, four mirrors, and a pearl placement platform. These materials are fixed in a well-closed container to efficiently avoid the stray light from the environment. When a pearl is sent to the placement platform by using a movable hollow rod, the ring-shaped LED light source evenly illuminates its surface. Due to the four plane mirrors, one shot can acquire five individual images of the pearl from different viewing angles, including top, left, right, main, and rear. Note that when the pearl arrives at the specific position, it is stationary in the detection device for a while, which provides enough time to detect and classify the pearl.

The segmentation, graying, and median filtering methods are used to get the five images of five viewing angles from the one-shot original image. These images are then fed into our MV-GAN to generate artificial multiview pearl images. Our data set includes 10 500 pearls, divided into seven classes with each category containing 1500 pearls. Each pearl has five images of different viewing angles. Our main goal of this study is to effectively extend the collected pearl image data set, so as to obtain a better MS-CNN model in the pearl classification machine, i.e., achieving the pearl classification of higher accuracy in practical applications.

IV. Method

Generally, GAN has no specific requirements on the structure of the generative and discriminative models [17]. In most cases, a deep neural network is adopted as the basic model due to its powerful feature extraction capability. To efficiently expand the training set of multiview pearl images, in this work, we propose MV-GAN as a new GAN framework. For convenience, we use the same network structure as the DCGAN [18], and use the same strategy as the CGAN [19] to restrict the training and the generating processes.

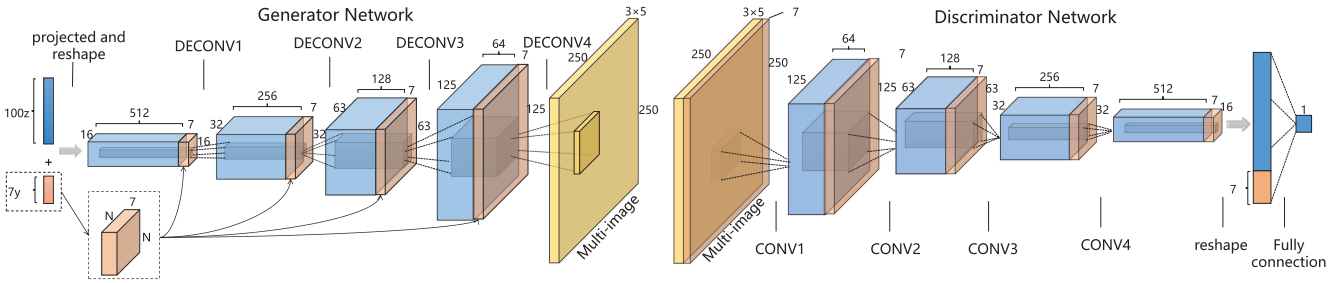

Fig. 2. Overall network structure of MV-GAN. There are four deconvolution layers in the generator network and four convolution layers in the discriminator network. The category label is added to the input layer and the feature maps of every hidden layers to limit the generation of MV-GAN so that it can generate specific category of multiview pearl images. To generate multiview images of a pearl, the input and output of the MV-GAN are set to a sequence of multiview images.

A. Basic Principle

GAN is composed of two models, i.e., the generative model G, which captures the data distribution, and the discriminative model D, which estimates the input probability from the training set. Both G and D are deep neural networks here, with the parameters denoted by θ_g and θ_d, respectively.

Let x be a training image and z be a prior noise. Then, the input of G is z, and the output of G is a fake image $G(z)$, i.e., the generator can be considered as a function $G(z;\theta_g)$ that maps the prior distribution of z to a probability distribution of $G(z)$ in image space. The input of D is x or $G(z)$, and the corresponding output of D is a single scalar $D(x;\theta_d)$ or $D(G(z);\theta_d)$, representing the probability that x or $G(z)$ came from the training data or $G(z)$. In the training process, G and D are updated simultaneously, and the ultimate goal is to get the parameter of G

$$\theta_g^* = \arg\min_{\theta_g}\max_{\theta_d} v(\theta_g,\theta_d) \tag{1}$$

where $v(\theta_g,\theta_d)$ is the loss function with the form

$$\begin{aligned} v(\theta_g,\theta_d) = {} & E_{x\sim p_{\text{data}}(x)}\left[\log D(x)\right] \\ & + E_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \end{aligned} \tag{2}$$

where $p_{\text{data}}(x)$ denotes the distribution of training data x and $p_z(z)$ denotes the distribution of the given prior noise z.

B. Multiview GAN

Without hypothesizing data distribution, GAN takes samples directly from a given data set and then approximates its distribution. However, this may lead to a quite large potential space of the generated data, making the GAN less reliable. Moreover, as an unsupervised learning method, GAN can only generate data similar to a given distribution rather than generate data with a specific label. Therefore, the traditional GAN does not meet our requirement in expanding the pearl images with labels for the classification task.

To overcome these shortages, we propose MV-GAN, with its structure shown in Fig. 2. For its good image generation ability, we use DCGAN [18] as the basic structure of the MV-GAN network. The following guidelines from [18] are used in the MV-GAN architecture: first, all the hidden layers are set to convolutional layers, and the artificially set pooling layers are eliminated [18] so that the network can learn its own upsampling (for the generator) and downsampling (for the discriminator) operators; second, fully connected layers are removed to accelerate the convergence speed [18]; and third, some tricks are used, i.e., the ReLU activation [52] is used for each hidden layer of the generator, the leak-ReLU is used for each hidden layer of the discriminator, and the batch normalization [53] is used for both generator and discriminator.

As discussed in CGAN [19], GAN can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information y. Here, y could be any kind of auxiliary information, such as class labels or data from other modalities. One can simply introduce the conditioning input and prior noise as the inputs to a single hidden layer of a multilayer perceptron, which was validated in CGAN on the Mixed National Institute of Standards and Technology (MNIST) data set [19]. Alternatively, one can also use higher order interactions for more complex generation mechanisms. For the proposed MV-GAN method, inspired by CGAN [19], a category label y as the condition information is added to the input. For the generator, the condition information y and the input noise $p_z(z)$ form a joint hidden layer representation, which can be considered as some restrictions on the potential dimensions, i.e., a constraint is imposed on the random generation of MV-GAN. Similarly, for the discriminator, the category label is also combined with the input image. Such conditional information is added to each hidden layer of the neural network to enhance the constraint, as shown in Fig. 2. The category label for the generator's input layer is one-hot coded. Sequentially, after every hidden layer outputs feature maps, the same category label is added as a part of them. Moreover, similar to the one-hot encoding, the feature map layer representing the correct category is set to one, while all the others are set to zero, as shown in Fig. 3. The same encoding method is used in the discriminator, whose task is to distinguish fake images from real ones for a certain category, due to the constraints of the category label. The main principle can be represented by

$$\begin{aligned} \min_{\theta_g}\max_{\theta_d} v(\theta_g,\theta_d) = {} & E_{x\sim p_{\text{data}}(x)}[\log D(x|y)] \\ & + E_{z\sim p_z(z)}[\log(1-D(\widetilde{x}|y))] \end{aligned} \tag{3}$$

with $\widetilde{x} = G(z|y)$.

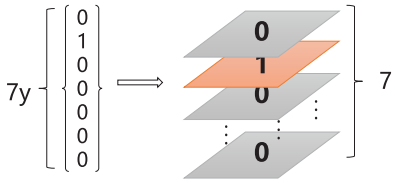

Fig. 3. Example to explain how to add conditions. The left vector represents the input label of the pearl of the second category and the right image represents the method of adding the corresponding label to the hidden layer. Its length and width are the same as the corresponding hidden layer feature map.

Since the MS-CNN requires multiview pearl images as the input, we arrange the multiview images in the order of top, left, right, main, and rear views of a pearl, and set them as the input of the MV-GAN, to generate the artificial multiview pearl images of the same order. In particular, we stack multiple views into a *cube* (or *cuboids*), and then convolves on this *cube*. As a result, each feature map is connected to these views. The multiview input of the image channel dimension allows the feature map obtained by the convolutional layer to capture the relevant information between different views of a pearl. The ith input training data thus can be represented by

$$x^i = [x^i_{\text{top}}, x^i_{\text{left}}, x^i_{\text{right}}, x^i_{\text{main}}, x^i_{\text{rear}}] \tag{4}$$

where $x^i_{\text{top}}, x^i_{\text{left}}, \ldots, x^i_{\text{rear}}$ represent the corresponding views of the ith pearl, respectively.

It should be noted that our MV-GAN is different from the multiview bidirectional generative adversarial networks (MV-BiGAN) method proposed by Chen and Denoyer [54] from the following two aspects. First, the main purposes are different, leading to two different frameworks. The purpose of MV-GAN is to generate multiview image data. It learns conditional distribution of the multiview training set and samples from the distribution by random noise. The category label is used as an input condition and both the training data and generated data are in multiview forms, i.e., multiview input and output. Differently, the MV-BiGAN [54] aims to estimate the distribution using multiview sets (or subsets) of their conditions. multiview sets are only used as input conditions for the entire framework rather than the training object. Second, the input of our MV-GAN method is more straightforward, i.e., no additional encoder is required, the category label is used as the input condition with the label dimensions set the same, and no additional mapping is needed. Consequently, compared with the network structure of MV-BiGAN, MV-GAN is simpler for practical use.

As an example, the generated pearl images, marked as *fake*, and the *real* ones are compared in Fig. 4. It can be seen that, by our method, the generated images indeed capture some features of the real images of the same view, such as the lighting direction. In addition, it also captures the correlation between the multiview images of the same pearl, i.e., the multiview images in each row have similar color and shape. For some pearls with especially complex textural and shape features, the MV-GAN may not be able to learn their features well and thus generate less desirable images, such as the multiview images in class 5 in Fig. 4. As mentioned in [45], although different from the

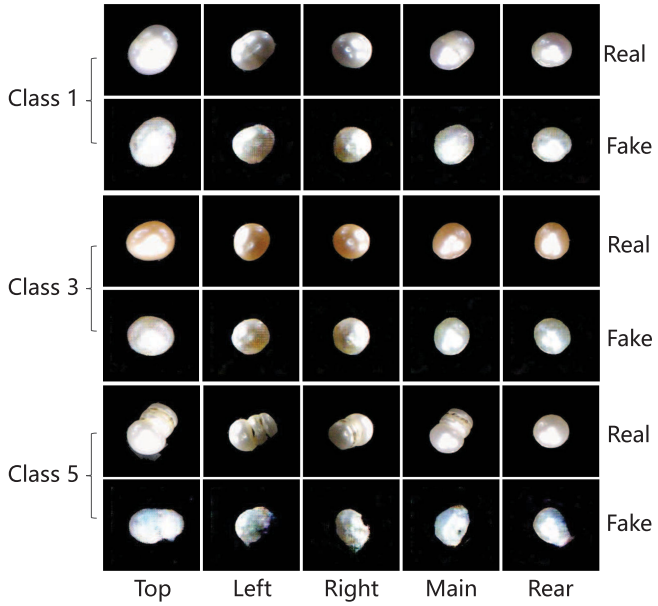

Fig. 4. Comparison of real and fake (generated by MV-GAN) images of pearls in different classes. We use the multiview images of pearls in seven categories, with 600 pearls in each category, to train the MV-GAN. Rows 1, 3, and 5 are the real multiview images of pearls in class 1, 3, and 5, respectively, in the training set. Rows 2, 4, and 6 are the corresponding fake multiview images generated by the MV-GAN.

multiview images of real pearls, these images may play a role similar to regularization, making the MS-CNN achieve better generalization performance.

In summary, the MV-GAN can be easy to modify according to the practical requirements for image generation, and the generated multiview images of pearls may help the MS-CNN have better classification performance.

C. Implementation Details

The size for each layer of MV-GAN is shown in Fig. 2. The generator network has four deconvolutional layers. Its input is a vector of 107 dimensions whose first 100 dimensions are uniformly distributed noises in $[-1, 1]$ and the last seven dimensions are one-hot codes of the pearl category label. The input vector is projected into a small spatial extent convolutional representation with many feature maps. After four deconvolution layers, it is mapped into a 15-channel image, which can be used as five 3-channel pearl images in the subsequent use. The discriminator network has four convolutional layers and its input is a 22-channel image whose first 15 channels are stacked from the five views of a pearl. The elements in one of the last seven channels are all one, representing the category label of the pearl. The network outputs a probability after four convolution and one fully connection layers.

We use the center-crop pearl image as the training input to reduce the complexity of the network. The image size is set to 250×250, the network batch size is set to 30, and the learning rate of the Adam optimization algorithm is set to 0.0002. The two networks are alternately trained until convergence: the discriminator is updated first; the generator follows; and so forth,

Algorithm 1: Training MV-GAN.

Input: training images $\mathbf{x} = \{x^1, x^2, \ldots, x^n\}$, training labels $\mathbf{y} = \{y^1, y^2, \ldots, y^n\}$, uniform distribution $[-1, 1]$.

Output: generated data $\{\widetilde{x}^1|y^1, \widetilde{x}^2|y^2, \ldots\}$

1: **repeat**
2: Sample m examples $\{x^1|y^1, x^2|y^2, \ldots, x^m|y^m\}$ from $\mathbf{x}$ and $\mathbf{y}$, correspondingly.
3: Sample m noisy samples $\{z^1, z^2, \ldots, z^m\}$ from the uniform distribution $[-1, 1]$.
4: Obtain $\{z^1|y^1, z^2|y^2, \ldots, z^m|y^m\}$ by adding labels to the noise samples.
5: Obtain generated data $\{\widetilde{x}^1|y^1, \widetilde{x}^2|y^2, \ldots, \widetilde{x}^m|y^m\}$, where $\widetilde{x}^i|y^i = G(z^i|y^i)|y^i$.
6: Update discriminator parameters θ_d to maximize: $v = \frac{1}{m}\sum_{i=1}^{m} \log D(x^i|y^i) + \frac{1}{m}\sum_{i=1}^{m} \log(1 - D(\widetilde{x}^i|y^i))$. $\theta_d \leftarrow \theta_d + \eta\nabla v(\theta_d)$.
7: Sample another m noisy samples $\{z^1, z^2, \ldots, z^m\}$ from the uniform distribution $[-1, 1]$.
8: Obtain $\{z^1|y^1, z^2|y^2, \ldots, z^m|y^m\}$ by adding labels to the noise samples.
9: Update generator parameters θ_g to minimize: $v = \frac{1}{m}\sum_{i=1}^{m} \log[1 - D(G(z^i|y^i)|y^i)]$. $\theta_g \leftarrow \theta_g - \eta\nabla v(\theta_g)$.
10: **until** convergence.
11: Sample $\{z^1, z^2, \ldots\}$ from the uniform distribution $[-1, 1]$.
12: Obtain generated data $\{\widetilde{x}^1|y^1, \widetilde{x}^2|y^2, \ldots\}$ by putting labels to generator: $\widetilde{x}^i|y^i = G(z^i|y^i)|y^i$.
13: Return $\{\widetilde{x}^1|y^1, \widetilde{x}^2|y^2, \ldots\}$.

with the details shown in Algorithm 1. We use TensorFlow to build the network and the program is run on GeForce GTX TITAN X.

V. Experiments

Here, MV-GAN is used to generate artificial multiview pearl images. These images, as well as those real pearl images, are used to train the MS-CNN [8] for pearl classification. Three experiments are performed. First, in order to study the impact of data volume on MV-GAN models, we use three sets, containing 1500, 1000, and 500 pearls in each category, respectively. The method is thus implemented on the three pearl image data sets of varied size, containing 10 500, 7000, and 3500 pearls, namely $Pearl10500$, $Pearl7000$, and $Pearl3500$, respectively. Second, the generated data of MV-GAN are further investigated on more CNN structures using five single-views separately. Third, the scenarios of pearl images under different lighting conditions are investigated to explore the effect of environment on the performance of MV-GAN.

For each data set, we divide it into a training set, a validation set, and a test set, with the ratio 3:1:1. Then, we use the MV-GAN to expand the training set, and further use the expanded training set and the validation sets to train the MS-CNN. Finally, we report the classification results on the test set.

TABLE I
Classification Results Obtained by the MS-CNN on the Original and Expanded Data Sets

Dataset	Training Set	Error (%)	Relative IMP (%)
Pearl10500	6300	8.76	-
Pearl10500	6300+2100	8.29	+5.37
Pearl10500	6300+4200	8.29	+5.37
Pearl10500	6300+6300	**8.00**	**+8.68**
Pearl7000	4200	15.50	-
Pearl7000	4200+1400	12.34	+20.39
Pearl7000	4200+2800	**11.36**	**+26.71**
Pearl7000	4200+4200	12.79	+17.48
Pearl3500	2100	18.14	-
Pearl3500	2100+700	17.71	+2.37
Pearl3500	2100+1400	**15.72**	**+13.34**
Pearl3500	2100+2100	16.00	+11.80

A. Classification Results

We first directly use the original training and validation data sets to train the MS-CNN [8]. This is our baseline model. Then, we expand the training set to different scales by the MV-GAN and further use the expanded training set, as well as the original validation set, to train the MS-CNN, in order to see the effectiveness of our MV-GAN method.

The classification results are shown in Table I, where the sign "+" in the second column indicates that the training set is expanded by the MV-GAN to certain extent. Overall, we can find that the multiview pearl images generated by our MV-GAN can indeed help the MS-CNN achieve better performance, i.e., for any case, the MS-CNN based on the expanded training set has lower classification error than that purely based on the original training set.

By comparison, the MV-GAN behaves best to improve the performance of the MS-CNN on the $Pearl7000$ data set, i.e., the classification error decreases most in this case when a number of multiview images are generated by the MV-GAN and added to collaboratively train the MS-CNN. The relative improvement (IMP) is even up to 26.71% when 2800 multiview pearl images are generated and added into the classification model. However, it seems that the MS-CNN benefits less from the MV-GAN for both larger and smaller data sets, i.e., we can only get smaller relative improvements on $Pearl10500$ and $Pearl3500$.

Such results can be explained as follows. When the data set is large enough, the MS-CNN itself can achieve quite a high classification accuracy. Therefore, the potential for further improvement is relatively low even when the MV-GAN is well-trained so that the generated multiview pearl images are of high quality. On the other hand, when the data set is too small, the MV-GAN may not be well-trained and thus produce poor multiview pearl images, this may hurt the classification model to a certain extent. Some of the poor multiview pearl images produced by the MV-GAN using the $Pearl3500$ data set are shown in Fig. 5, which appear less on the other two data sets. As one can see, these images look very coarse and are quite different from the real pearl images.

When the data set is slightly insufficient, the MS-CNN itself may not produce a satisfied classification result. The multiview pearl images generated by the MV-GAN thus can play a role of regularization in this situation, i.e., it tends to sample the images

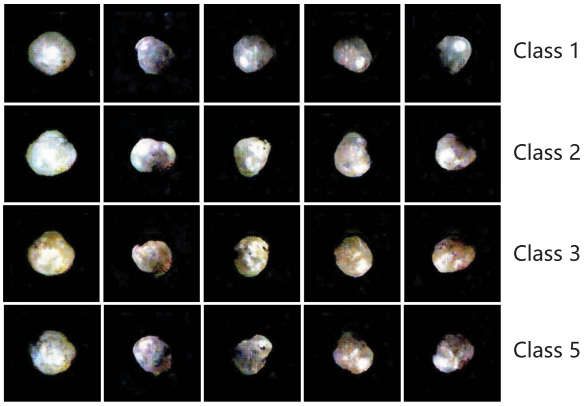

Fig. 5. Some poor multiview pearl images generated by the MV-GAN using the Pearl3500 data set.

TABLE II
CLASSIFICATION RESULTS USING SINGLE-VIEW PEARL IMAGES BY THE CNN OF DIFFERENT STRUCTURES, WITH OR WITHOUT MV-GAN

Method	Error(%)				
	Top	Left	Right	Main	Rear
AlexNet	23.71	15.29	12.52	24.52	17.90
AlexNet+MV-GAN	20.48	14.43	12.38	21.71	19.00
Relative IMP (%)	+13.62	+5.62	+1.12	+11.46	-6.15
GoogleNet	14.90	13.67	10.71	16.86	17.71
GoogleNet+MV-GAN	16.19	11.38	10.19	16.19	17.05
Relative IMP (%)	-8.66	+16.75	+4.86	+3.97	+3.73
ResNet34	16.14	11.24	12.05	18.91	18.00
ResNet34+MV-GAN	15.95	12.38	10.81	19.51	15.95
Relative IMP (%)	+1.18	-10.14	+10.29	-3.17	+11.39

that are identical but somewhat different in certain characteristics from the real images in the original training set, which may prevent the overfit, so as to improve the performance, of the MS-CNN.

B. Classification Using Single-View Images

For comparison, we also use the single-view pearl images generated by the MV-GAN to train the CNN of different structures. This is mainly to investigate whether multiview images generated by MV-GAN can represent more information than single-view ones. We performed the experiments on the data set $pearl10500$ and generate the same number of pearl images as the training set, i.e., $6300 + 6300$. The experimental results are shown in Table II, from which we conclude the following.

1) First, different views can provide different amounts of information, resulting in different classification performances, i.e., we find that the CNN models, with or without MV-GAN, behave better when using the pearl images of left and right views.
2) Second, compared with the multiview images, single-view images seem to provide much less information to distinguish different classes of pearls, i.e., we find that the AlexNet, with or without MV-GAN, behaves much better when using multiview pearl images than using single-view pearl images. This result, consistent with [8], indicates the advantage of MS-CNN.
3) Third, taking the single-view images generated by MV-GAN to expand the training set may, or may not, improve the classification accuracy, depending on the network structure and viewing angle. Therefore, it is essential to use MV-GAN to generate multiview pearl images to significantly improve the performance of MS-CNN.

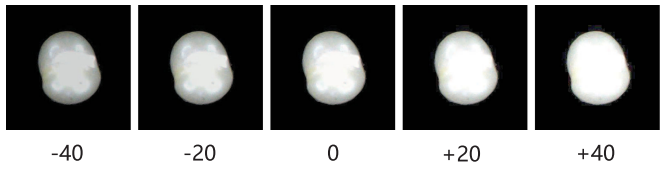

Fig. 6. Pearl images of different levels of brightness.

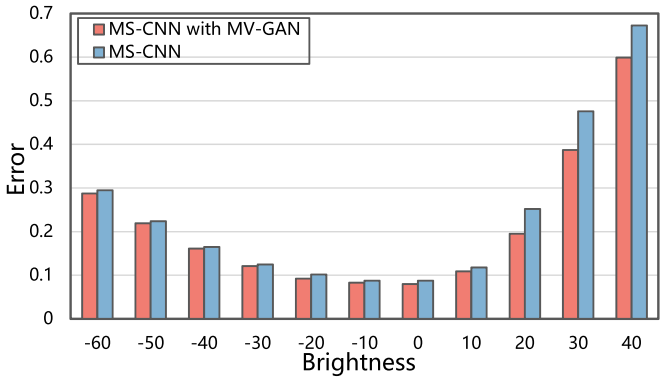

Fig. 7. Classification results under the brightness disturbance.

C. Classification Under Different Brightness

In real industrial application, there might be many environmental factors that can influence the performance of vision-based machines. One typical environmental factor is lighting. Take our pearl classification machine, for example, different machines may have different lighting conditions, even when we use the exactly same model of light sources.

We synthesize the pearl images under different lighting conditions as follows: for the regions where the pixel value is nonzero, we uniformly change the pixel value to a different level of brightness. For instance, Fig. 6 presents five pearl images of different levels of brightness. The middle one is the original image. The left two are of lower brightness, i.e., their pixel values are reduced by 20 and 40, respectively, and the right two are of higher brightness, i.e., their pixel values are increased by 20 and 40, respectively. We use these pearl images of different levels of brightness as our new test sets.

As an illustrated case, we still use the data set $pearl10500$ to test the classification performance under different levels of brightness. The classification results on the new test sets are shown in Fig. 7. MS-CNN with MV-GAN means that the MS-CNN model is trained by the 6300 original images and the 6300 images generated by MV-GAN, while MS-CNN means that the MS-CNN model is trained only by the 6300 original images. Experimental results show that the multiview pearl images generated by MV-GAN can indeed make the MS-CNN model more robust under the brightness disturbance, especially in the situation when the brightness increases. This result suggests that our MV-GAN method can make the classification

model less influenced by environment, and thus make it more practical in industrial applications.

VI. Conclusion

In this paper, we proposed a new framework of GAN, namely MV-GAN, to generate multiview pearl images for the first time. Then, we used this technology to successfully generate multiview pearl images, which can be used to significantly improve the performance of MS-CNN in pearl classification. The experimental results validated that the multiview pearl images generated by our MV-GAN can indeed help to reduce the classification error, and the combination of MV-GAN and MS-CNN achieve the state-of-the-art results on pearl classification. Moreover, it seemed that MV-GAN can also help MS-CNN resist the disturbance introduced by the environment, e.g., the brightness disturbance, leading to more robust classification. These may accelerate the industrial application of the automatic pearl classification machine. Note that our MV-GAN can also be used to generate the multiview images of other objects. It thus has the potential to be applied in many other 3-D object classification tasks [20]–[23].

Considering that the inner environment may be slightly varied for different pearl classification machines, in the future, we will consider such differences in our MV-GAN, to make sure that the MS-CNN models established on a few machines can also perform well in many others, preparing for the large-scale industrial application.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.

[2] O. Costilla-Reyes, P. Scully, and K. B. Ozanyan, "Deep neural networks for learning spatio-temporal features from tomography sensors," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 645–653, Jan. 2018.

[3] F.-C. Chen and M. R. Jahanshahi, "Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018.

[4] X. Gao, C. Shang, Y. Jiang, D. Huang, and T. Chen, "Refinery scheduling with varying crude: A deep belief network classification and multimodel approach," *AIChE J.*, vol. 60, no. 7, pp. 2525–2532, 2014.

[5] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 812–820, Jun. 2015.

[6] Y. Liu, Y. Fan, and J. Chen, "Flame images for oxygen content prediction of combustion systems using DBN," *Energy Fuels*, vol. 31, no. 8, pp. 8776–8783, 2017.

[7] Y. Liu, C. Yang, Z. Gao, and Y. Yao, "Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes," *Chemometrics Intell. Laboratory Syst.*, vol. 174, pp. 15–21, 2018.

[8] Q. Xuan *et al.*, "Automatic pearl classification machine based on a multistream convolutional neural network," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6538–6547, Aug. 2018.

[9] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1134–1142.

[10] H. Zhang *et al.*, "SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1143–1152.

[11] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked CNN for fine-grained visual categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1173–1182.

[12] R. M. Haralick *et al.*, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[13] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[15] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception*. New York, NY, USA: Elsevier, 1992, pp. 65–93.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[17] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[18] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015. [Online]. Available: http://arxiv.org/abs/1511.06434

[19] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

[20] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multiview CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5648–5656.

[21] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multiview convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.

[22] Z. Wu *et al.*, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[23] P. Zanuttigh and L. Minto, "Deep learning for 3D shape classification from multiple depth maps," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3615–3619.

[24] G. Li *et al.*, "Pearl shape recognition based on computer vision," *Trans. Chinese Soc. Agricultural Mach.*, vol. 39, no. 7, pp. 129–132, 2008.

[25] G. Li, B. Li, Y. Wang, and J. Li, "Classification method of pearl luster degree based on HSL," *Trans. Chin. Soc. Agricultural Mach.*, vol. 39, no. 6, pp. 113–117, 2008.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[28] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.

[29] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[31] E. L. Denton *et al.*, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.

[32] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generating images with recurrent adversarial networks," *CoRR*, vol. abs/1602.05110, 2016. [Online]. Available: http://arxiv.org/abs/1602.05110

[33] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *CoRR*, vol. abs/1701.07875, 2017. [Online]. Available: http://arxiv.org/abs/1701.07875

[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5769–5779.

[36] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016. [Online]. Available: http://arxiv.org/abs/1609.04802

[37] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[38] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR*, vol. abs/1703.10593, 2017. [Online]. Available: http://arxiv.org/abs/1703.10593
[39] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image to-image translation," *CoRR*, vol. abs/1704.02510, 2017. [Online]. Available: http://arxiv.org/abs/1704.02510
[40] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *CoRR*, vol. abs/1703.05192, 2017. [Online]. Available: http://arxiv.org/abs/1703.05192
[41] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
[42] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *CoRR*, vol. abs/1701.06547, 2017. [Online]. Available: http://arxiv.org/abs/1701.06547
[43] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *CoRR*, vol. abs/1605.05396, 2016. [Online]. Available: http://arxiv.org/abs/1605.05396
[44] L. Wu *et al.*, "Adversarial neural machine translation," *CoRR*, vol. abs/1704.06933, 2017. [Online]. Available: http://arxiv.org/abs/1704.06933
[45] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by GAN improve the person re-identification baseline in vitro," *CoRR*, vol. abs/1701.07717, 2017. [Online]. Available: http://arxiv.org/abs/1701.07717
[46] X. Wang, M. You, and C. Shen, "Adversarial generation of training examples for vehicle license plate recognition," *CoRR*, vol. abs/1707.03124, 2017. [Online]. Available: http://arxiv.org/abs/1707.03124
[47] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[49] A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, and A. Zhavoronkov, "drugan: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico," *Molecular Pharmaceutics*, vol. 14, no. 9, pp. 3098–3104, 2017.
[50] G. L. Guimaraes, B. Sanchez-Lengeling, P. L. C. Farias, and A. Aspuru-Guzik, "Objective-reinforced generative adversarial networks (organ) for sequence generation models," *CoRR*, vol. abs/1705.10843, 2017. [Online]. Available: http://arxiv.org/abs/1705.10843
[51] S. Kohl *et al.*, "Adversarial networks for the detection of aggressive prostate cancer," *CoRR*, vol. abs/1702.08014, 2017. [Online]. Available: http://arxiv.org/abs/1702.08014
[52] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[53] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167
[54] M. Chen and L. Denoyer, "Multiview generative adversarial networks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2017, pp. 175–188.

Qi Xuan (M'18) received the B.S. and Ph.D. degrees in control theory and engineering from Zhejiang University, Hangzhou, China, in 2003 and 2008, respectively.

He was a Postdoctoral Researcher with the Department of Information Science and Electronic Engineering, Zhejiang University from 2008 to 2010, and a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China, in 2010 and 2017. From 2012 to 2014, he was a Postdoctoral Fellow with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently a Professor with the College of Information Engineering, Zhejiang University of Technology. His current research interests include network-based algorithms design, social networks, data mining, cyberspace security, machine learning, and computer vision.

Zhuangzhi Chen received the B.S. degree in electrical engineering and automation from the Zhejiang University of Technology, Hangzhou, China, in 2017. He is currently working toward the M.S. degree in control theory and engineering from the Zhejiang University of Technology.

His current research interests include computer vision, deep learning method including deep convolutional neural networks, and generative adversarial networks.

Yi Liu (M'11) received the Ph.D. degree in control theory and engineering from Zhejiang University, Hangzhou, China, in 2009.

He was a Postdoctoral Researcher with the Department of Chemical Engineering, Chung-Yuan Christian University from 2012 to 2013. He is currently an Associate Professor with the Zhejiang University of Technology, Hangzhou. He has published about 30 international journal papers. His research interests include data intelligence with applications to modeling, control, and optimization of industrial processes.

Huimin Huang was born in Shenzhen, China, in 1997. She is currently working toward the B.S. degree in automation with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China.

Her current research interests include computer vision, machine learning, and deep learning.

Guanjun Bao (M'18) received the B.S. degree in mechanical engineering from North China Electric Power University, Beijing, China, in 2001, and the Ph.D. degree in mechatronics from the Zhejiang University of Technology, Hangzhou, China, in 2006.

In 2006, he joined the College of Mechanical Engineering, Zhejiang University of Technology as an Assistant Professor, where he was appointed as an Associate Professor in 2009. His research interests includes robotics, soft robotics, machine vision, and smart manufacturing. He has authored or coauthored about 70 papers.

Dan Zhang received the Ph.D. degree in control theory and control engineering from the Zhejiang University of Technology, Hangzhou, China, in 2013.

He is currently a Research Fellow with the Department of Biomedical Engineering, City University of Hong Kong, Hong Kong, China. His current research interests include hybrid systems, cyber-physical systems, and artificial intelligence.