

Diversified and Multi-Class Controllable Industrial Defect Synthesis for Data Augmentation and Transfer

Jing Wei^{1,2} Fei Shen^{1,2,3} Chengkan Lv^{1,2,3} Zhengtao Zhang^{1,2,3} Feng Zhang^{1,2,3} Huabin Yang^{1,2,3}

¹Institute of Automation, Chinese Academy of Sciences

²The School of Artificial Intelligence, University of Chinese Academy of Sciences

³CASI Vision Technology CO.

{weijing2020, fei.shen, chengkan.lv, zhengtao.zhang, feng.zhang, huabin.yang}@ia.ac.cn

Abstract

Data augmentation is crucial to solve few-sample issues in industrial inspection based on deep learning. However, current industrial data augmentation methods have not yet demonstrated on-par ability in the synthesis of complex defects with pixel-level annotations. This paper proposes a new defect synthesis framework to fill the gap. Firstly, DCDGANc (Diversified and multi-class Controllable Defect Generation Adversarial Networks based on constant source images) is proposed to employ class labels to construct source inputs to control the category and random codes to generate diversified styles of defects. DCDGANc can generate defect content images with pure backgrounds, which avoids the influence of non-defect information and makes it easy to obtain binary masks by segmentation. Secondly, the Poisson blending is improved to avoid content loss when blending generated defect contents to the normal backgrounds. Finally, the complete defect samples and accurate pixel-level annotations are obtained by fine image processing. Experiments are conducted to verify the effectiveness of our work in wood, fabric, metal, and marble. The results show that our methods yield significant improvement in the segmentation performance of industrial products. Moreover, our work enables zero-shot inspection by facilitating defect transfer between datasets with different backgrounds but similar defects, which can greatly reduce the cost of data collection in industrial inspection.

1. Introduction

Supervised deep learning methods have achieved excellent performance in industrial defect inspection, which greatly promotes the development of intelligent manufacturing [1]. However, data issues, such as the small number of defect samples, the poor diversity of datasets, and the huge time cost of creating datasets, limit the application of these industrial inspection methods [2]. The

continuous optimization of modern industrial processes has led to fewer and fewer defective samples, that is, the number of defect images is very limited [3]. Few training samples cannot meet the needs of inspection based on supervised deep-learning methods [4]. Insufficient data will lead to overfitting and poor generalization ability of the deep-learning methods [5]. Furthermore, when the type of the product changes, it is inevitable to recollect and relabel defect samples to retrain the inspection system. Therefore, data issues are major challenges to improve industrial inspection performance and save costs.

Image data augmentation is a key way to solve data issues and improve the performance of deep-learning methods. It involves methods based on traditional image manipulations and deep learning [3]. The former consists of color space conversion, noise injection, rotation, and so on, which can't edit the defects and are insufficient to enrich defects essentially. Using generative networks to synthesize images is one of the incredible deep-learning augment methods [6]. Recently, drawn by the broad success of generative models in image synthesis, a few researchers attempt to generate industrial defect samples by GANs [7-13]. These methods have achieved data augmentation in corresponding datasets and improved their inspection performance. According to generated results, these defect synthesis methods can be roughly divided into two types, *i.e.*, those with image-level annotations [7-9] and those with pixel-level annotations [10-13].

Defect synthesis with image-level annotations can only serve for the performance of the classification models. Based on CycleGAN [14], SDGAN [7] generates high-quality and diversified images of the commutator cylinder surface defect dataset and improves the classification performance. Defect-GAN [8] can generate defects on normal images and remove defects in defective images in the meantime based on CycleGAN and StarGAN [15]. DTD-GAN [9] diversifies the tire-defect shape by mutual information maximization and adds an auxiliary classifier to control the tire defect category. Although these methods can generate complex defects, they cannot improve the performance of segmentation models, and their range of

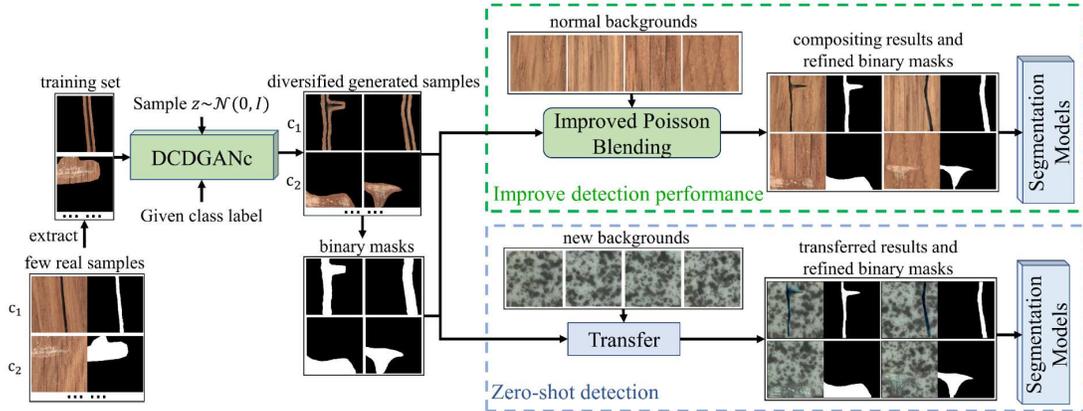


Figure 1: Overall schema of the proposed methods.

application is limited.

Defect synthesis with pixel-level annotations can serve for the performance of both the classification and the segmentation models. Multistage GAN [10] uses defect patch generation and defect-fusing networks to fine-tune the segmentation networks. SIGAN [11] uses CycleGAN and L1 loss to achieve defect augmentation and segmentation of solar cells. D.-M. Tsai et al. [12] utilize 8 networks to automatically synthesize and annotate defect pixels in an image. S. Niu et al. [13] propose a defect image generation method with controllable defect regions and strength which successfully generates weak gray defects on the Kolektor surface-defect and the metal hook defect datasets.

However, there is room for improvement in defect synthesis, particularly in the robustness of generation methods and the generation of high-quality and diversified complex defects with accurate pixel-level annotations. Firstly, current defect synthesis methods may be disturbed by the non-defect background information. In the generation of the whole defect image, the non-defect information may be generated as defects. Secondly, these methods cannot totally preserve real normal backgrounds and provide pixel-level annotations for defects with complex textures. Besides, most of the existing methods that can generate high-quality defects rely on CycleGAN [14] to avoid paired training inputs [7, 8, 11, 12]. There are fewer random factors in their generation process, that is, the diversity is insufficient. They can only get one defective image from a normal image by a well-trained model.

To resolve the interference of non-defective information, get rid of the constraints of paired training inputs, and generate more diversified defects with pixel-level annotations, this paper proposes a new multi-class controllable defect synthesis framework. We decouple the defect sample synthesis into two stages, defect generation and compositing with normal backgrounds, so that users can choose the normal backgrounds and category of defects

separately to construct wanted defect samples. We propose DCDGANc (Diversified and multi-class Controllable Defect Generation Adversarial Networks based on constant source image) to generate defect content images with pure backgrounds which makes it easy to obtain binary masks for defects. Then the defect is fused into normal backgrounds by proposed improved Poisson blending, where real normal contents are totally retained. The accurate binary masks of defect samples are obtained by fine image processing after compositing. Different from other defect sample generation methods, our main contributions are as follows,

(1) Only defect contents that are extracted from whole defect samples are used to train DCDGANc. As a result, our method totally removes non-defect background information and can obtain accurate binary masks of defect samples by fine image processing.

(2) By constructing constant input for our generator, DCDGANc can control the category of generated defects and gets rid of paired training inputs.

(3) DCDGANc generates diversified defects by modulating random codes into the generator with proposed OD-SPADE (One-Dim Spatially Adaptive Normalization).

Experimental results suggest that DCDGANc can control generated defect categories, and obtain richer diversity and better quality than other methods. Defect inspection experiments demonstrate the superiority of our synthetic samples against other augmented samples. Besides, our work achieves zero-shot tile defect inspection by defect transferring. In the following parts, we will introduce our method in detail and show relevant experiments.

2. Method

2.1. Motivation

The flowchart of our work is shown in Figure 1. Firstly, the normal backgrounds of the real defect images are

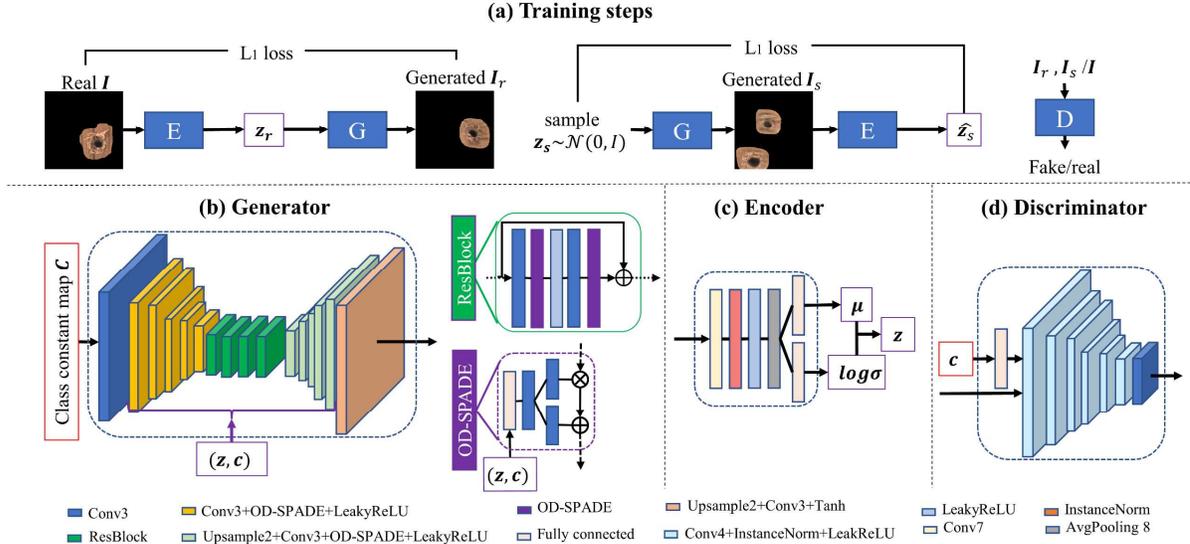


Figure 2: (a) Training process of DCDGANc, where E, G, and D in blue boxes are the encoder, generator, and discriminator. (b)-(d) Architectures of generator, encoder, and discriminator of DCDGANc. *Conv* n means the kernel size of the convolution is $n \times n$, *Upsample2* means the image is up-sampled by 2 times, *AvgPooling* n means the kernel size of the average pool filter is $n \times n$.

subtracted by binary masks, then only the defect contents are used to train *DCDGANc*. Afterward, raw binary masks can be obtained from generated defect content images with pure backgrounds. Then the *Improved Poisson Blending* is used to fuse the generated defects to the normal backgrounds. Finally, the synthetic datasets with refined masks are constructed after image processing, which are utilized to train the *Segmentation Models* to improve inspection performance. Moreover, our defect *Transfer* extends the generated wood defects to the tile backgrounds to create a segmentation training set without any real tile defect samples. The transferred datasets can reach zero-shot inspection. This means that the detection networks can be trained without collecting real tile defects, which greatly saves the cost of intelligent manufacturing.

2.2. DCDGANc

Our DCDGANc is proposed to generate diversified complex defect contents for multiple categories in the same item. Firstly, to control the category and get rid of the constraint of paired training inputs, we repeatedly pad the class label to obtain a class constant map. This constant map is the same size as the output image of the generator and is used as the source input for the generator. Secondly, DCDGANc modulates the random codes from latent space into the generator to control the styles of generated defects by our OD-SPADE (One Dim-SPADE) which is modified from SPADE (Spatially-Adaptive Normalization) [16]. OD-SPADE modulates the connection of one-hot class code and random noise into the generator to stabilize the training process and generate diversified defects.

As shown in Figure 2, DCDGANc includes an encoder, a generator, and a discriminator. The input image is encoded into latent space to obtain mean μ and logarithmic variance $\log \sigma$. Sampled random noise z and one-hot class code c are connected and modulated into the generator by OD-SPADE, which achieves the mapping from latent space to image space. The generator generates a defect content image of the category specified by the input class constant map. An image and c are inputted into the discriminator to distinguish whether the input image is real or not under the given class c .

Multi-class controllable. The generator employs class constant maps [17, 18] as input to control the categories. Different class labels are standardized into $[-1, 1]$, then input maps are constructed by simple pixel-level repetition of standardized results. Each class corresponds to a specific class constant map \mathcal{C} . For the n th class of N classes, the value $\mathcal{C}_{n,k}^{i,j}$ at the coordinate (i, j) of the k th channel of the constant map \mathcal{C} is

$$\mathcal{C}_{n,k}^{i,j} = n \frac{2}{N} - 1 \quad (1)$$

where n is an integer between 0 and N , i, j , and k depend on the image size in the training set. Since the category n and the one-hot class code c are different representations of the defect category, the following parts omit n and use c to represent the category.

Modulation of style. SPADE can modulate the style information into a high-dimension tensor to achieve element-level modulation. It can capture the difference between pixels and is more suitable for few-shot tasks. Therefore, as shown in the OD-SPADE in Figure 2, we add

Table 1. Algorithm flow chart of DCDGANc

Algorithm Training of DCDGANc
Require: real dataset DI , L1 loss function
for epoch in epochs
for \mathbf{c}, \mathbf{I} in DI
$\boldsymbol{\mu}, \log \boldsymbol{\sigma} = E(\mathbf{I}), \text{compute: } \mathcal{L}_{KL}$
$\mathbf{z}_r = \text{reparameterization}(\boldsymbol{\mu}, \boldsymbol{\sigma}),$
$\mathbf{I}_r = G(\mathbf{z}_r \mathbf{c}), \text{compute: } \mathcal{L}_p$
sample $\mathbf{z}_s \sim \mathcal{N}(0, I), \text{compute: } \mathbf{I}_s = G(\mathbf{z}_s \mathbf{c})$
compute: $\mathcal{L}_{adv} + \lambda_p \mathcal{L}_p + \lambda_{KL} \mathcal{L}_{KL}, \text{update } E$
$\hat{\mathbf{z}}_{s,-} = E(\mathbf{I}_s), \text{compute: } \lambda_l \mathcal{L}_l, \text{update } G$
maximize $\mathcal{L}_{adv}, \text{update } D$
end for
end for

a fully connected layer to adapt SPADE [16] to the one-dimension input and cut the number of convolutional channels in the first layer to simplify the architecture. In the generation process, OD-SPADE modulates the connection of one-hot class code and random noise into the normalized middle feature maps. Different noises correspond to different styles.

Training objectives. The VAE idea [19] is used for reference in this work to generate diversified defects. Firstly, the real image \mathbf{I} is encoded into latent space, $\boldsymbol{\mu}, \log \boldsymbol{\sigma} = E(\mathbf{I})$. Using KL loss [20] to constrain encoded outputs to follow Gaussian distribution,

$$\mathcal{L}_{KL} = \mathbb{E}_{\mathbf{I} \sim p(\mathbf{I})} [\mathcal{D}_{KL}(E(\mathbf{I}) || \mathcal{N}(0, I))] \quad (2)$$

where $\mathcal{D}_{KL}(p_1 || p_2) = - \int p_1(z) \log \frac{p_1(z)}{p_2(z)} dz$. Sample \mathbf{z}' from $\mathcal{N}(0, I)$, \mathbf{z}_r is obtained by reparameterization,

$$\mathbf{z}_r = \mathbf{z}' \cdot \boldsymbol{\sigma} + \boldsymbol{\mu} \quad (3)$$

then we input \mathbf{z}_r into generator to obtain reconstructed image $\mathbf{I}_r = G(\mathbf{z}_r | \mathbf{c})$, and calculate the reconstruction loss to construct mapping from Gaussian latent space to image space,

$$\mathcal{L}_p = \mathbb{E}_{\mathbf{I} \sim p(\mathbf{I}), \mathbf{z}_r \sim E(\mathbf{I})} \|\mathbf{I} - \mathbf{I}_r\|_1 \quad (4)$$

where $\|\cdot\|_1$ is the L1 loss. Next, we sample latent code \mathbf{z}_s from $\mathcal{N}(0, I)$ to get $\mathbf{I}_s = G(\mathbf{z}_s | \mathbf{c})$. The encoded mean $\hat{\mathbf{z}}_{s,-} = E(\mathbf{I}_s)$ should be equal to \mathbf{z}_s so as to construct mapping from image space to Gaussian latent space, which is constrained by the latent loss

$$\mathcal{L}_l = \|\mathbf{z}_s - \hat{\mathbf{z}}_{s,-}\|_1 \quad (5)$$

By cooperating E and G twice above, DCDGANc constitutes the constraint in two loops of $\mathbf{I} - \mathbf{z}_r - \mathbf{I}_r$ and $\mathbf{z}_s - \mathbf{I}_s - \hat{\mathbf{z}}_{s,-}$. Furthermore, the adversarial loss is calculated to ensure the realism of the generated results

$$\begin{aligned} \mathcal{L}_{adv} = & 2\mathbb{E}_{\mathbf{I} \sim p(\mathbf{I})} [\log(D(\mathbf{I} | \mathbf{c}))] \\ & + \mathbb{E}_{\mathbf{z}_r \sim E(\mathbf{I})} [\log(1 - D(G(\mathbf{z}_r | \mathbf{c}) | \mathbf{c}))] \\ & + \mathbb{E}[\log(1 - D(\mathbf{I}_s | \mathbf{c}))] \end{aligned} \quad (6)$$

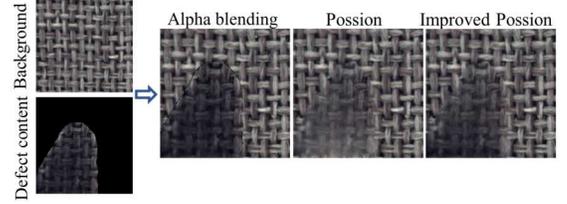


Figure 3: The edge between the *Defect content* (left-lower) and *Background* (left-upper) of the *Alpha blending* is fractured. In the *Poisson* blending result, the background excessively penetrates into the defect. Our *Improved Poisson* achieves a good transition at the edge while retaining the defect content well.

Finally, the objective function is

$$\begin{aligned} G^*, E^*, D^* = & \arg \min_{G, E} \max_D (\mathcal{L}_{adv}(D, G, E) + \\ & \lambda_p \mathcal{L}_p(G, E) + \lambda_l \mathcal{L}_l(G, E) + \lambda_{KL} \mathcal{L}_{KL}(E)) \end{aligned} \quad (7)$$

where λ_p, λ_l and λ_{KL} are hyperparameters to control the contribution of each loss to the overall objective. The algorithm flow is shown in Table 1.

Testing process. DCDGANc generates defect content images based on class labels and sampled random noises, then when testing, we only need to sample \mathbf{z} from Gaussian distribution and specify a class label to obtain various defect content images under the given class.

2.3. Defects compositing

Defect contents generated by DCDGANc need to be blended into normal backgrounds to acquire whole defect samples. Firstly, the binary mask M of the generated defect content images are obtained by watershed segmentation. Then improved Poisson blending is proposed to avoid the problems of foreground content loss in Poisson blending and edge-cutting between foreground and background in alpha compositing. First, the binary mask M of the generated defect image I_d is progressively transformed by distance transformation [21]. Then we select 5 pixels at the boundary of the distance transformed mask to construct the boundary mask M_w . Next, alpha compositing is adopted to get the background input I_{an} of Poisson blending

$$I_{an} = M_w \odot I_d + (1 - M_w) \odot I_n \quad (8)$$

\odot is the pixel-level multiplication. The region to be blended (I_{dc}) is extracted by binary mask M from I_d . The area occupied by I_{dc} in I_{an} is Ω , and $\partial\Omega$ represents the boundary of Ω . $\bar{\Omega}$ denotes the region except Ω in I_{an} . p_d and p_α are pixel values in Ω and $\bar{\Omega}$ respectively. Then Poisson blending formula is used to calculate the whole defect image I_{IPB}

$$\min_{p_d} \iint_{\Omega} |\nabla p_d - \nabla I_{dc}|^2 \text{ with } p_d | \partial\Omega = p_\alpha | \partial\Omega \quad (9)$$

$$I_{IPB} = p_d + p_\alpha \quad (10)$$

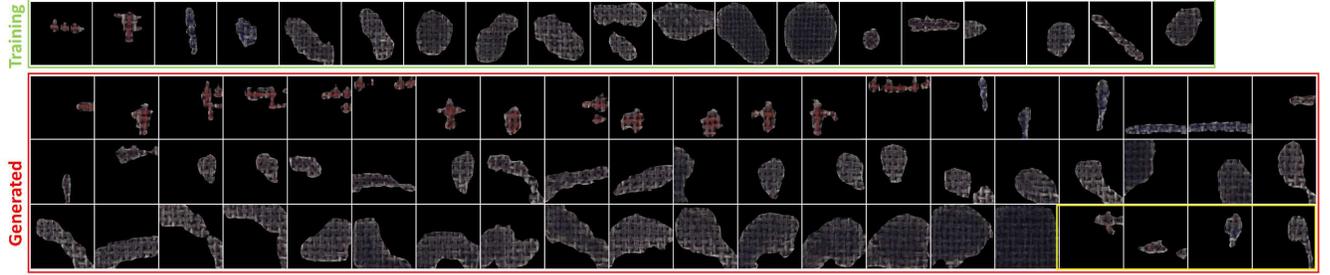


Figure 4: Diversity comparison (256^2) of carpet-color. The *Training* contains all defect contents in the training sets, and the *Generated* contains 60 images generated by DCDGANc. The images in yellow boxes are mixing color defects that don't exist in the training sets.

where \iint is the double integral and ∇ is gradient calculation. As the boundary condition, the pixel values of I_{IPB} are consistent with I_{an} at the boundary. The compositing results of these three methods are shown in Figure 3. It can be seen that our improved Poisson blending can not only achieve seamless compositing but also preserve the defect contents. To further obtain the complete defect samples with accurate pixel-level annotations, we refine binary masks according to the characteristics of compositing results by simple image processing.

3. Experiments

3.1. Experiment setup

Datasets. We perform experiments on defect images of 6 classes in the carpet and 5 classes in the wood from MVTEC-AD [22] separately. In order to preserve the similarity among kinds of defects in the same item, the original binary masks are expanded for some defect types whose labels don't contain any background texture. The numbers of defect samples are shown in Table 2. There are multiple categories of defects in one image of the *Wood-Combined*, whose defect contents are dispersed to corresponding defect datasets. For the convenience of description, metal contamination is denoted by mc below.

Implementations. We use Adam optimizer with $\beta_1=0.5$ and $\beta_2=0.999$ to train DCDGANc with the batch size=20 and the learning rate=0.0005. We assign $\lambda_p=10$, $\lambda_l=1$, $\lambda_{KL}=0.01$, and add WGAN-GP loss [23] with the weight of 10 into the equation (7) to stabilize training. We trained the DCDGANc for 500 iterations on one NVIDIA GeForce RTX 3090 GPU of a server with Intel(R) Xeon(R) Gold 622306R CPU @ 2.90GHz.

3.2. Qualitative results

Diversification. Figure 4 shows the comparison between all real defect contents and partially generated contents of carpet-color. It can be seen that DCDGANc generated a large number of defects with different shapes

Table 2. The number of defect images in each category (original /constructed)

Defects	Datasets		Wood
	Carpet	Defects	
Color	19/1055	Color	8/976
Cut	17/1409	Hole	10/1033
Hole	17/1178	Liquid	10/620
Metal contamination	17/1016	Scratch	21/1115
Thread	19/1580	Combined	11/-

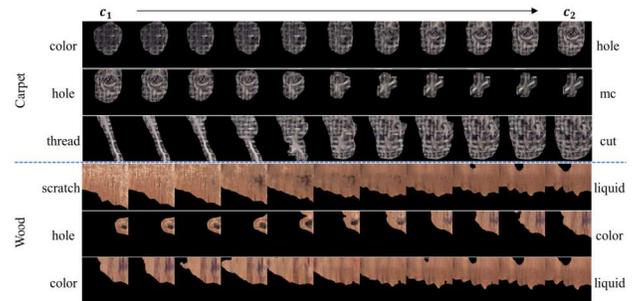


Figure 5: Results (256^2) of classes interpolation under the same style code. Three pairs of results are shown in *Carpet* and *Wood* respectively. Each row from left to right represents the results of the transition from c_1 to c_2 in the step of $0.1c_1$. With the change of class label, the category of generated defects gradually changes.

and contents for datasets with only 19 images. Generated results are different from training sets because random noises are introduced into generation by OD-SPADE. The images in yellow boxes indicate that DCDGANc is able to mix information from different real images into a single generated image. Overall, introducing randomness into the generation by our OD-SPADE greatly enhances the diversity of generated results, which assists DCDGANc to map different style codes into various defect contents and achieve a rich expansion for few-sample datasets.

Classes controlling. As shown in Figure 5, all defects in the same item derive from the same one z , and the continuous change of class labels causes the category of generated results gradually transits from the original c_1 to the target c_2 . It indicates that DCDGANc models the class space continuously and accurately by our constructed class

Table 3. Contents of the segmentation datasets (*Raw* and *Test* show the number of real samples (1024²)/cropped images (256²))

Dataset	Defect	Carpet					Wood			
		color	cut	hole	mc	thread	color	hole	liquid	scratch
Raw(real/cropped)		12/114	12/73	12/102	13/330	13/126	5/116	6/173	7/97	11/316
Augment/Enlarge		456	365	408	660	504	481	493	484	650
Test(real/cropped)		7/51	5/49	5/39	4/85	6/28	3/65	4/68	3/38	10/77

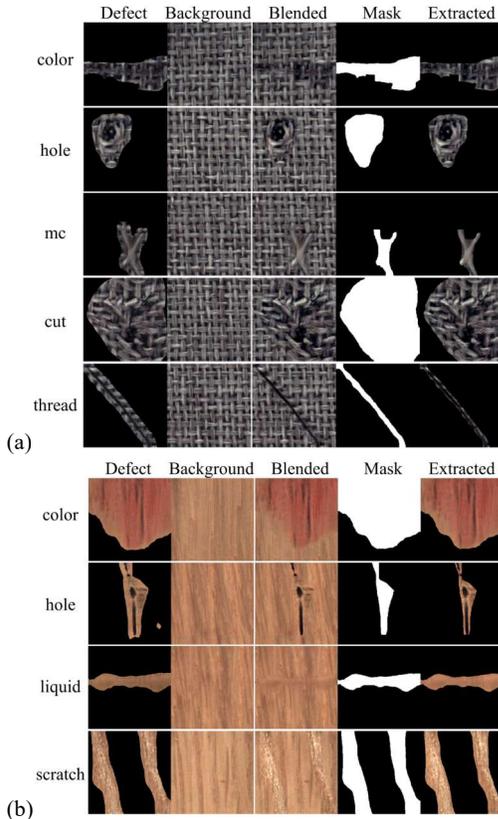


Figure 6: Synthetic results. Each row includes five images (256²). *Defect* is generated by DCDGANc. *Background* is the normal background used for compositing. *Blended* is the result of improved Poisson blending. *Mask* is the refined binary mask. *Extracted* is defect content extracted by *Mask*. (a) Synthetic results of carpet. (b) Synthetic results of wood.

constant maps. Besides, the results of the same row which comes from the same \mathbf{z} indicate that DCDGANc can preserve the style similarity between different classes. In summary, DCDGANc achieves correspondence of class space and latent style space and can control categories and styles independently by the cooperation between OD-SPADE and class constant maps.

Compositing results. Figure 6 shows that the defects generated by DCDGANc and real normal backgrounds are well preserved in the compositing defect images, and the refined binary masks accurately indicate the defect locations. Overall, our methods can construct synthetic datasets with accurate pixel-level annotations for various complex texture defects, which has rarely been considered in previous defect synthesis works.

Table 4. Test results (AUC/F1) under two models and three training sets

Dataset		Wood	Carpet
Res	Raw	0.9891/0.8462	0.9812/0.7497
	Augment	0.9891/0.8227	0.9862/0.7833
	Enlarge	0.9903/0.8486	0.9901/0.7961
U-Net	Raw	0.9936/0.8784	0.9846/0.7776
	Augment	0.9943/0.8778	0.9851/0.7967
	Enlarge	0.9945/0.8896	0.9903/0.8135

3.3. Application evaluation

To verify the role of synthetic samples in inspection, we construct three different segmentation training sets. Synthetic samples of our method and augmented samples of traditional methods of brightness adjustment, rotation, and noise injection are added to the real dataset “*Raw*” to get “*Enlarge*” and “*Augment*” respectively. The numbers of images are shown in Table 3. Three training sets are used to train segmentation networks based on the backbones of U-Net [24] and ResNet [25] respectively. The AUC (Area Under Curve) and F1 coefficient calculated from the same test sets are shown in Table 4. Optimal results are indicated by bolding. Table 4 show that compared with the *Raw* and the *Augment*, *Enlarge* greatly improves the inspection performance where AUC improves by up to 1% and F1 improves by up to 4%. Figure 7 shows the inspection results of six groups of hard-to-detect samples based on U-Net. Compared with the testing results of *Raw* and *Augment*, *Enlarge* has the most accurate segmentation results. There is overkill and escape inspection in *Raw* and *Augment* while *Enlarge* accurately detects the defects.

The segmentation experiments show that models trained by the *Enlarge* are more conducive to detecting unseen and hard-to-detect samples. Diversified and realistic defect contents generated by DCDGANc provide more different textures and shapes of defects to the inspection models. Then inspection models can see a richer set of defects in training by *Enlarge* and obtain better generalization. In summary, our synthetic samples are more beneficial to develop the potential of the inspection model against traditional augmented samples, help the model to clear decision boundaries, and improve inspection performance.

3.4. Comparison results

To verify the advantages of DCDGANc and its key modulation block OD-SPADE, we compare DCDGANc with other state-of-the-art methods and replace OD-SPADE with AdaIN [26] and InstanceNorm respectively.

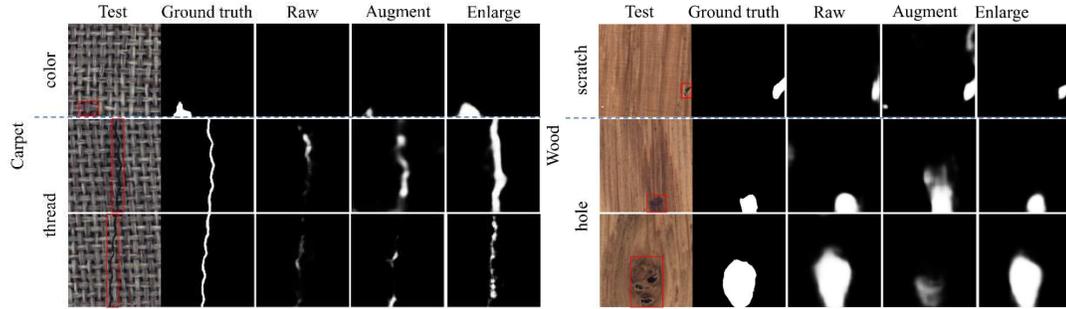


Figure 7: Inspection results (256^2) of hard-to-detect samples. *Test* is the test image, where the red boxes indicate the defects. *Ground truth* is the real defect annotations. *Raw*, *Augment*, and *Enlarge* are the segmentation results from three training sets, respectively. In the two columns, there are one carpet-color, two carpet-thread defects, one wood-scratch, and two wood-hole defects, from top to bottom.

Table 5. Segmentation results (AUC/F1) of all testing samples based on different augmented training sets

Dataset	Model	DCDGANc	CycleGAN	AdaIN	InstanceNorm	StyleGAN2	StarGAN
Wood	Res	0.9903/0.8486	0.9731/0.8439	0.9740/0.8002	0.9819/0.8476	0.9715/0.8386	0.9749/0.7995
	U-Net	0.9945/0.8896	0.9815/0.8677	0.9841/0.8513	0.9838/0.8746	0.9761/0.8765	0.9800/0.8605
Carpet	Res	0.9901/0.7961	0.9477/0.6653	0.8860/0.6561	0.9582/0.7138	0.9601/0.7338	0.9451/0.6790
	U-Net	0.9903/0.8135	0.9638/0.7527	0.9804/0.7987	0.9511/0.7137	0.9567/0.7382	0.9572/0.7103

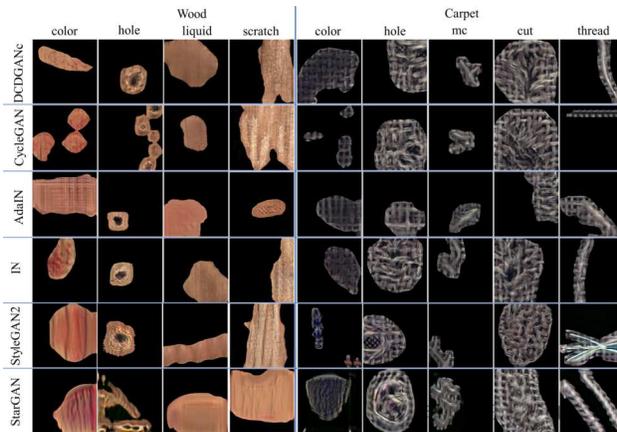


Figure 8: Comparison (256^2) results. Each row shows the generated results of one method on nine defects.

As shown in Figure 8, other methods cannot generate realistic-like defects on black backgrounds (StyleGAN v2-carpet-cut [18], StarGAN [15], AdaIN, InstanceNorm-wood-color). There is mode collapse in the generation of carpet-thread by CycleGAN where all source images only get the same output. In contrast, our OD-SPADE successfully assists DCDGANc to obtain diversified and realistic-like defect samples in few-shot tasks. DCDGANc can control category based on the class constant map and generate complex defects without paired training inputs, whose generated results are more realistic, and training is more stable.

Furthermore, to verify the effectiveness of DCDGANc on inspection over these methods, we add the synthetic results of these methods to *Raw* to train segmentation models. The test results are shown in Table 5, where

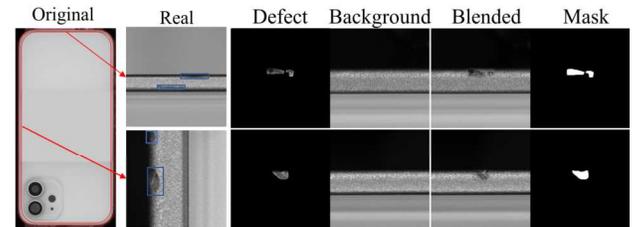


Figure 9: *Real* is cropped from the *Original* of size 5472×10980 where red boxes indicate band regions. Blue boxes in *Real* indicate defects. Columns 3-6 are contents related to synthesis. Generated defects of size 64×256 are placed in black backgrounds of size 256^2 (*Defect*).

Table 6. The numbers and test results of three training sets

Dataset		Raw	Augment	Enlarge
		Number	1653	3306
AUC/F1	Res	0.978/0.544	0.988/0.633	0.993/0.660
	U-Net	0.908/0.678	0.917/0.653	0.944/0.701

DCDGANc achieves the best performance. In summary, our method is most beneficial to improve inspection performance due to the high-quality and diversified generated results.

3.5. Engineering application

To test the practical application effectiveness of this work in real industrial scenes, the above experiments are conducted on the metal phone band dataset shown in Figure 9 (*Original* and *Real*). We crop the extracted defect content images into a total of 2852 images with size

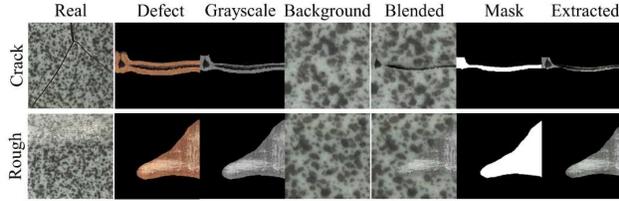


Figure 10: *Real* is defect images (840^2) of *Crack* and *Rough* in tile. *Defect* (256^2) is generated by DCDGANc. *Grayscale* (256^2) is the grayscale *Defect*, where defective edges of wood-hole are partially cut. *Background* (256^2) is used for compositing. *Blended* (256^2) is the result of improved Poisson blending. *Mask* (256^2) is the refined binary mask. *Extracted* (256^2) is defect content.

Table 7. The numbers of training sets and test sets (Test shows the number of real images (840^2)/cropped images (256^2))

Defect Name	Crack	Rough
Train	418	913
Test(real/cropped)	17/82	15/105

64×256 to train DCDGANc. Synthetic results are shown in Figure 9. The numbers and the segmentation results of the three training sets are shown in Table 6, the size of all images is 128×128 . It can be seen that compared with the traditional augmented samples, our synthetic samples improve F1 by up to **9.5%** (Res) and AUC by up to **3%** (U-Net). The experimental results of generation, synthesis, and inspection show that our work has achieved excellent results in the augmentation of metal phone band defect samples and has practical application value in intelligent manufacturing.

3.6. Defects transfer

In this part, we will show that supervised segmentation models can be trained without any real defect samples by our defect transfer.

The hole and the scratch defects in wood are fused to the normal backgrounds of MVTec-tile as the transferred defect samples of the crack and the rough in marble tile respectively. As shown in Figure 10, the transferred defect images are similar to the real tile defect images. Then only the transferred defect images are used to train segmentation models, and all the real tile images are used as the test set. The numbers of the datasets are shown in Table 7. The test results of two segmentation models with backbones of ResNet and U-Net are shown in Table 8. It can be seen that models trained only by transferred samples achieve excellent test performance for real tile defects, where AUC is up to **0.9919**, and F1 is up to **0.8376**. Figure 11 shows some segmented results which are similar to the ground truth.

In summary, by dividing defect synthesis into two steps, defect contents generation and compositing with normal backgrounds, we can independently operate defect content

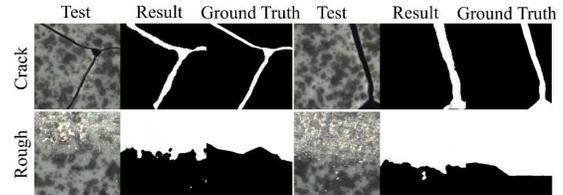


Figure 11: Zero-shot segmentation results. *Test* is the real tile defect image. *Result* is the segmented result from models trained by transferred datasets. *GroundTruth* is real defect annotations.

Table 8. Test results of zero-shot segmentation (AUC/F1)

Defect \ Model	Crack	Rough
Res	0.9896/0.6876	0.9688/0.8376
U-Net	0.9919/0.7279	0.9790/0.8129

and normal background, and easily achieve wonderful zero-shot inspection performance by transferring generated defects of other datasets into new target backgrounds.

4. Conclusion

This paper proposes a method to solve the problems of lacking accurate pixel-level annotations, poor diversity, and interference from non-defective information in industrial complex defect synthesis. Firstly, based on constructed source input and proposed modulation block OD-SPADE, DCDGANc can generate diversified and multi-category defect contents where binary masks can be easily obtained. Secondly, we improve Poisson blending to avoid the loss of defect contents in compositing with normal images. Finally, experimental results show that our method synthesizes high-quality and diversified samples which are more conducive to improve the inspection performance compared with traditional and other synthetic samples. Besides, our defect transfer can reach to zero-shot detection with AUC up to **0.9919**. However, the effectiveness of our method on weak defects with inconspicuous texture changes still needs to be improved. In future works, we will continue to explore the application value of our methods in other scenarios.

References

- [1] Chen, Y.; Ding, Y.; Zhao, F.; Zhang, E.; Wu, Z.; Shao, L.. Surface Defect Detection Methods for Industrial Products: A Review., *Appl. Sci.* 11, 7657, 2021. doi: 10.3390/app11167657.
- [2] Yanqi Bao, Kechen Song, Jie Liu, et al., Triplet-Graph Reasoning Network for Few-Shot Metal Generic Surface Defect Segmentation, *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-11, Art no. 5011111, 2021, doi: 10.1109/TIM.2021.3083561.

- [3] Shorten C., Khoshgoftaar T. M., A survey on Image Data Augmentation for Deep Learning., *J Big Data* 6. vol.60, July, 2019, doi: 10.1186/s40537-019-0197-0.
- [4] Tabernik D., Šela S., Skvarč J., and Skočaj D., Segmentation-based deep-learning approach for surface-defect detection, *Journal of Intelligent Manufacturing*, vol.31, pp. 759-776, May., 2020. doi:10.1007/s10845-019-01476-x.
- [5] J. Xu and Y. Liu, Multiple Guidance Network for Industrial Product Surface Inspection With One Labeled Target Sample, *IEEE Transactions on Neural Networks and Learning Systems*, 2022, doi: 10.1109/TNNLS.2022.3165575.
- [6] Jain M., Meegan C. and Dev S.. Using Gans to Augment Data for Cloud Image Segmentation Task. *IEEE International Geoscience and Remote Sensing Symposium IGARSS*. pp. 3452-3455, 2021, doi: 10.1109/IGARSS47720.2021.9554993.
- [7] Niu S., Li B., Wang X. and Lin H. Defect Image Sample Generation With GAN for Improving Defect Recognition. *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1611-1622, July, 2020. doi: 10.1109/TASE.2020.2967415.
- [8] Zhang G., Cui K., Hung T. Y. and Lu S. Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2523-2533, doi: 10.1109/WACV48630.2021.00257.
- [9] Y. Zhang et al., Diversifying Tire-Defect Image Generation Based on Generative Adversarial Network, in *IEEE Transactions on Instrumentation and Measurement*, 2022, vol. 71, pp. 1-12, 2022, Art no. 5007312, doi: 10.1109/TIM.2022.3160542.
- [10] Liu J., Wang C., Su H., Du B., and Tao D.. Multistage GAN for Fabric Defect Detection. *IEEE Transactions on Image Processing*, vol. 29, pp. 3388-3400, 2020, doi: 10.1109/TIP.2019.2959741.
- [11] Su B., Zhou Z., Chen H. X., and Cao. SIGAN: A Novel Image Generation Method for Solar Cell Defect Segmentation and Augmentation. *ArXiv preprint*. arXiv:2104.04953, 2021.
- [12] D. -M. Tsai, S. -K. S. Fan and Y. -H. Chou, Auto-Annotated Deep Segmentation for Surface Defect Detection, in *IEEE Transactions on Instrumentation and Measurement*, 2021, vol. 70, pp. 1-10, 2021, Art no. 5011410, doi: 10.1109/TIM.2021.3087826.
- [13] S. Niu, B. Li, X. Wang and Y. Peng, Region- and Strength-Controllable GAN for Defect Generation and Segmentation in Industrial Images, in *IEEE Transactions on Industrial Informatics*, 2022, vol. 18, no. 7, pp. 4531-4541, July 2022, doi: 10.1109/TII.2021.3127188.
- [14] Zhu J., Park T., Isola P., and Efros A. A.. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2242-2251, 2017, doi: 10.1109/ICCV.2017.244.
- [15] Choi Y., Choi M., Kim M., Ha J. W., Kim S. and Choo J.. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 8789-8797, 2018.
- [16] Park T., Liu M. Y., Wang T. C., and Zhu J. Y., Semantic Image Synthesis With Spatially-Adaptive Normalization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2332-2341, doi: 10.1109/CVPR.2019.00244.
- [17] Karras T., Laine S., and Aila T.. A Style-Based Generator Architecture for Generative Adversarial Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4396-4405, doi: 10.1109/CVPR.2019.00453.
- [18] Karras T., Laine S., Aittala M., Hellsten J., Lehtinen J., and Aila T.. Analyzing and Improving the Image Quality of StyleGAN. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8107-8116, doi: 10.1109/CVPR42600.2020.00813.
- [19] Liu M. Y H., Breuel T., Kautz J.. Unsupervised image-to-image translation networks. *Advances in neural information processing systems (NeurIPS)*, vol 30, pp. 700-708, Dec., 2017, doi: 10.5555/3294771.3294838.
- [20] Kingma D. P., Welling M., Auto-encoding variational bayes. *ArXiv preprint*. 2013, arXiv:1312.6114, 2013.
- [21] Rosenfeld, Pfaltz J. L.. Sequential operations in digital picture processing. in *Journal of ACM*, 1966, vol. 13(4), pp. 471-494, Oct. 1966, doi: 10.1145/321356.321357.
- [22] Bergmann P., Fauser M., Sattlegger D., and Steger C.. MVTEC AD-A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9584-9592, doi: 10.1109/CVPR.2019.00982.
- [23] Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., Courville A. C. Improved training of wasserstein gans. *Advances in neural information processing systems*, 2017, vol.30.
- [24] Ronneberger, Fischer P., Brox T.. U-Net: Convolutional Networks for Biomedical Image Segmentation. *International Conference on Medical image computing and computer-assisted Intervention, Lecture Notes in Computer Science*, Springer, Cham. 2015, vol 9351. https://doi.org/10.1007/978-3-319-24574-4_28.
- [25] He K., Zhang X., Ren S., and Sun J.. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [26] Huang X., Belongie S.. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1510-1519, doi: 10.1109/ICCV.2017.167.