Defect-GAN : high-fidelity defect synthesis for automated defect inspection

Zhang, Gongjie; Cui, Kaiwen; Hung, Tzu-Yi; Lu, Shijian

2021

https://hdl.handle.net/10356/146285

Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection

Gongjie Zhang[1]　　　Kaiwen Cui[1]　　　Tzu-Yi Hung[2]　　　Shijian Lu[*1]

[1]Nanyang Technological University　　　[2]Delta Research Center, Singapore

{gongjiezhang,shijian.lu}@ntu.edu.sg　　kaiwen001@e.ntu.edu.sg　　tzuyi.hung@deltaww.com

Abstract

Automated defect inspection is critical for effective and efficient maintenance, repair, and operations in advanced manufacturing. On the other hand, automated defect inspection is often constrained by the lack of defect samples, especially when we adopt deep neural networks for this task. This paper presents Defect-GAN, an automated defect synthesis network that generates realistic and diverse defect samples for training accurate and robust defect inspection networks. Defect-GAN learns through defacement and restoration processes, where the defacement generates defects on normal surface images while the restoration removes defects to generate normal images. It employs a novel compositional layer-based architecture for generating realistic defects within various image backgrounds with different textures and appearances. It can also mimic the stochastic variations of defects and offer flexible control over the locations and categories of the generated defects within the image background. Extensive experiments show that Defect-GAN is capable of synthesizing various defects with superior diversity and fidelity. In addition, the synthesized defect samples demonstrate their effectiveness in training better defect inspection networks.

1. Introduction

Automated visual defect inspection aims to automatically detect and recognize various image defects, which is highly demanded in different industrial sectors, such as manufacturing and construction. In manufacturing, it is one key component in maintenance, repair, and operations (MRO) that aims to minimize the machinery breakdown and maximize production. It is also important for quality control for spotting anomalies at different stages of the production pipeline. In construction, it is critical to public safety by identifying potential dangers in various infrastructures such as buildings, bridges, etc. Although automated visual defect inspection has been studied for years, it remains a challeng-
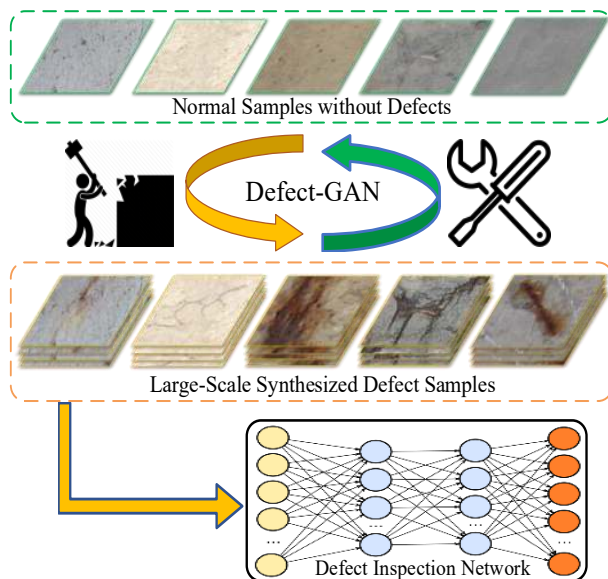

Figure 1. Mimicking the defacement and restoration processes over the easily collected normal samples, Defect-GAN generates large-scale defect samples with superior fidelity and diversity. The generated defect samples demonstrate great effectiveness in training accurate and robust defect inspection network models.

ing task with a number of open research problems.

One key challenge in automated visual defect inspection lies with the training data, which usually manifests in two different manners. First, collecting a large number of labeled defect samples are often expensive and time-consuming. The situations become much worse due to the poor reusability and transferability of defect samples, i.e., we often have to re-collect and re-label defect samples while dealing with various new defect inspect tasks. Second, collecting defect samples is not just about efforts and costs. In many situations, the defect samples are simply rare, and the amount available is far from what is required, especially when training deep neural network models. The availability of large-scale defect samples has become one bottleneck for effective and efficient design and development of various automated defect inspection systems.

---

*Corresponding author.

An intuitive way to mitigate the defect insufficiency issue is to synthesize defect samples. Though Generative Adversarial Networks (GANs) have achieved superior image synthesis in recent years, synthesizing defect samples using GANs is still facing several challenges. *First*, existing GANs usually require large-scale training data, but large-scale defect samples are not available in many situations. *Second*, GANs tend to generate simpler structures and patterns by nature [3] and so are not good at synthesizing defects that often have complex and irregular patterns with large stochastic variations. *Third*, defect samples with different backgrounds are very difficult to collect, and GANs thus tend to generate defect samples with similar backgrounds as the collected reference samples. As a result, the GANs synthesized defect samples often have similar feature representation and distribution as those reference samples and offer little help while facing various new defect samples on various different backgrounds.

Inspired by [36] that collects defect samples by manually damaging the surface of normal work-pieces, we design a Defect-GAN that aims for automated generation of high-fidelity defect samples for training accurate and robust defect inspection networks. Defect-GAN simulates the defacement and restoration processes, which greatly mitigates the defect-insufficiency constraint by leveraging large-scale normal samples that are often readily available. We design novel control mechanisms that enable Defect-GAN to generate different types of defects at different locations of background images flexibly and realistically. We also introduce randomness to the defacement process to capture the stochastic variation of defects, which improves the diversity of the generated defect samples significantly. Additionally, we design a compositional layer-based network architecture that allows for generating defects over different normal samples but with minimal change of normal samples' background appearance. As a result, the model trained with such generated defect samples is more capable of handling new defect samples with variously different backgrounds. Extensive experiments show that Defect-GAN can generate large-scale defect samples with superior fidelity and diversity as well as effectiveness while applied to train deep defect inspection networks.

The contributions of this work can be summarized in three aspects. *First*, we design a compositional layer-based network architecture to generate defects from normal samples while preserving the appearance of normal samples, which improves the defect diversity by simulating how defects look like on various normal samples. *Second*, we propose a Defect-GAN that synthesizes defects by simulating defacement and restoration processes. It offers superior flexibility and control over the category and spatial locations of the generated defects in the image background, achieves great defect diversity by introducing stochastic

variations into the generation process, and is capable of generating high-fidelity defects via defacement and restoration of normal samples. *Third*, extensive experiments show that the Defect-GAN generated defect samples help to train more accurate defect inspection networks effectively.

2. Related Works

Image Synthesis. GANs [13] are a powerful generative model that simultaneously trains a generator to produce realistic faked images and a discriminator to distinguish between real and faked images. Early attempts [13, 46, 2, 23, 4] focus on synthesizing images unconditionally. Recently, more and more works emerge to perform image synthesis conditioned on input images, which has wide applications including style translation [33, 20, 26, 75, 35, 28], facial expression editing [7, 45, 6, 60, 59], super-resolution [30, 58, 48], image inpainting [66, 67, 44, 63], etc. Another trend is multi-modal image synthesis [24, 25, 19, 8, 76]. However, existing methods fail to generalize well on defect synthesis. Our Defect-GAN is designed to generate defect samples by simulating the defacement and restoration processes and incorporating randomness to mimic the stochastic variations within defects. Besides, inspired by [64, 50, 42, 69], it deems defects as a special foreground and adopts a layer-based architecture to compose defects on normal samples, thus reserve the normal samples' style and appearance and achieving superior synthesis realism and diversity.

Learning From Limited Data. Deep learning based techniques [47, 71, 68] usually require a large amount of annotated training samples, which are not always available. Recent researches have proposed many attempts to mitigate the data-insufficiency issue. They can be broadly categorized as few-shot learning and data augmentation.

Few-shot learning [51, 52, 12, 5, 31, 22, 62, 57, 70, 10, 55] refers to learning from extremely limited training samples (e.g., 1 or 3) for an unseen class. However, their performances are quite limited and thus far from practical application. Besides, few-shot learning techniques usually require large amounts of samples from the same domain, which does not lift the data-insufficiency constraint. Data augmentation aims to enrich the training datasets in terms of quantity and diversity such that better deep learning models can be trained. Several recent research attempts [1, 56, 61, 40] adopt GANs as data augmentation methods to synthesis realistic training samples. The proposed Defect-GAN also works as a data augmentation method to train better defect inspection networks by synthesizing various defect samples with superior diversity and fidelity.

Defect Inspection. Surface defect inspection refers to the process of identifying and localize surface defects based on machine vision, which is an important task with extensive real-life applications in industrial manufacturing,
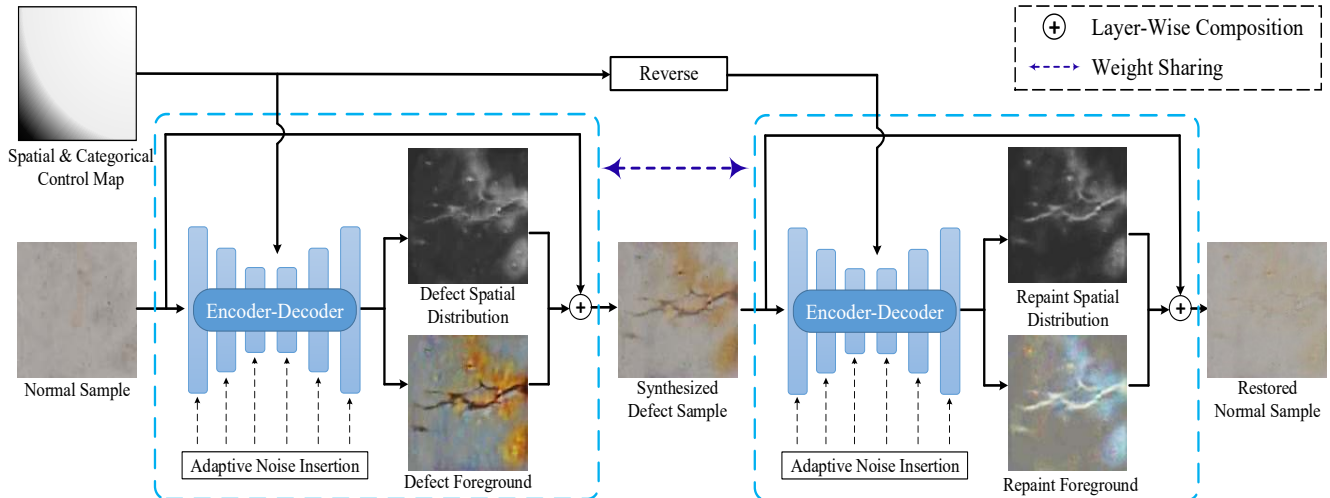
Figure 2. Generation pipeline of the proposed Defect-GAN: It adopts an encoder-decoder structure to synthesize defects by mimicking defacement and restoration processes. The *Spatial & Categorical Control Map* generated from category vectors controls where and what kind of defects to generate within the provided normal sample. The *Adaptive Noise Injection* introduces stochastic variations into the generated defects to improve the diversity of the generated defects. In addition, Defect-GAN adopts a *Layer-Wise Composition* strategy that produces defect and repaint foregrounds according to the corresponding spatial distribution maps. This helps preserve the style and appearance of the normal samples and achieve superior realism in defect synthesis.

safety inspection, building construction, etc. Before deep learning era, traditional methods [39, 54, 29, 74, 53] design hand-crafted feature extractors and heuristic pipelines, which needs specialized expertise and not robust. In deep learning era, many works [32, 9, 41, 38] adopt Convolutional Neural Networks (CNNs) based models for defect inspection and achieve remarkable performances.

However, in practical scenarios, limited number of defect samples has always been a bottleneck issue. To mitigate such defect-insufficiency issue, [36] manually destroys work-pieces to collect defect samples; [37, 18] further adopt Computer-Aided Drawing (CAD) to synthesis defect samples. However, such methods can only handle simple cases. The recently proposed SDGAN [40] adopts GANs to perform defect sample synthesis for data augmentation. We also propose to synthesis defect samples with GANs for training better defect inspection networks. By simulating the defacement and restoration processes with a layer-wise composition strategy, our proposed Defect-GAN can generate defect samples with superior realism, diversity, and flexibility. It can further provide better transferability by imposing learnt defect patterns on unseen surfaces.

3. Methodology

In this section, we discuss the proposed method in details. As illustrated in Fig. 1, our proposed method consists of two parts: (1) Defect-GAN design for automated synthesis of defect samples, and (2) defect inspection by using the synthesized defect samples.

3.1. Defect-GAN for Defect Synthesis

We hypothesis that there are sufficient amount of normal samples, and only a limited number of defect samples since defects are usually rare and difficult to capture. Based on this hypothesis of data availability, we propose to perform defect synthesis following the paradigm of unpaired image-to-image translation [75, 7], which usually requires less training data and can produce better synthesis fidelity. Our proposed Defect-GAN is based on the intuition that defects do not exist out of thin air, i.e., there is always a defacement process to generate defects over those normal samples, and there also exists a restoration process to restore the defect samples back to normal samples. By mimicking the defacement and restoration processes as mentioned above, we are able to leverage the large number of normal samples to generate required defect samples.

The Defect-GAN architecture consists of a generator G and a discriminator D. During the training stage, Defect-GAN performs image translation using G in two cycles: $n \rightarrow d \rightarrow \hat{n}$ and $d \rightarrow n \rightarrow \hat{d}$, where $n \in R^{H \times W \times 3}$ denotes a normal sample, $d \in R^{H \times W \times 3}$ denotes a defect sample, and $\hat{n}, \hat{d} \in R^{H \times W \times 3}$ denote restored normal and defect sample, respectively. Since the two cycles are identical and simultaneously conducted, we only describe the cycle $n \rightarrow d \rightarrow \hat{n}$ in the following sections for simplicity.

The generator G is illustrated in Fig. 2. It employs an encoder-decoder architecture. The major architecture of G mainly follows the commonly used image-to-image translation networks [20, 75, 7], which first encodes the input im-

age by a stride of 4, and then decodes it to its original size. To improve synthesis realism and diversity for defect generation, we specifically design spatial and categorical control, stochastic variation and layer-based composition for G. The network architecture of D is the same as StarGAN [7], which includes a D_{src} to distinguish faked samples from real ones using PatchGAN [21] and a D_{cls} to predict the categories of generated defects.

Spatial and Categorical Control for Defect Generation. Different types of defects can exist on different locations of normal samples. To provide better attribute (spatial and categorical) control over the generated defects, we feed an attribute controlling map $A \in R^{H \times W \times C}$ into G to add specific kind of defects to specific location, where $A_{x,y} \in R^C$ represents the presence of defects at the corresponding location, and C denotes the number of defect categories. A is imposed into the network via SPADE normalization [43] and is fed into every block in the decoder part of G.

Note that since we only assume image-level annotations available, during training stage, the attribute controlling map A should be constant for all locations of the image, i.e., A is acquired by spatial-wisely repeating the target defect label $c \in R^C$. This restriction can be lifted during inference stage, which enables Defect-GAN to add defects at different location(s) in a context-compatible manner.

Stochastic Variation of Defects. Unlike general objects, defects are known to possess complex and irregular patterns with extremely high stochastic variations that are extremely challenging to model using GANs. To mitigate this issue, we employ an adaptive noise insertion module in each block of the encoder-decoder architecture, which explicitly injects Gaussian noise into the feature maps after each convolutional block. For each noise injection, it learns a exclusive scalar to adjust the intensity of the injected noise. By explicitly mirroring the stochastic variations within defects, Defect-GAN can generate more realistic defect samples with much higher diversity.

Layer-Wise Composition. As illustrated in Fig. 2, Defect-GAN is also different from existing image-to-image translation GANs [20, 75, 7, 40] in that we consider the final generation as composition of two layers. Specifically, in the defacement process, final defects samples are generated by adding a defect foreground layer on top of the provided normal samples. Similarly, in the restoration process, final restored normal samples are generated by adding a repaint foreground layer on top of the defect samples.

The defacement process can be formulated as:

$$f_d, m_d = G(n, A_{n \to d}) \tag{1}$$

$$d = n \odot (1 - m_d) + f_d \odot m_d \tag{2}$$

where $\odot$ denotes spatial-wise multiplication, f_d denotes generated defect foreground, and $m_d \in [0, 1]$ denotes the

corresponding spatial distribution map of f_d. Similarly, the restoration process can be formulated as:

$$f_{\hat{n}}, m_{\hat{n}} = G(d, A_{d \to \hat{n}}) \tag{3}$$

$$\hat{n} = d \odot (1 - m_{\hat{n}}) + f_{\hat{n}} \odot m_{\hat{n}} \tag{4}$$

where $\hat{n}$ denotes restored normal sample without defects.

The intuition behind this layer-wise composition strategy is that defects can be deemed a special kind of foreground composed on the background (normal samples). Similarly, the restoration process that removes defects from the background can also be considered a 'repainting' process to cover the defect areas. Instead of generating synthesized images directly, Defect-GAN separately generates defect foregrounds along with the corresponding spatial distribution maps, and then performs an layer-wise composition to produce the synthesized defect samples.

The novel compositional layer-based synthesis can significantly improve defect synthesis in terms of both realism and diversity. This is mainly because by taking normal samples as background, our model implicitly focuses on generation of defects, without considering the generation of backgrounds. This feature provides our model with more capability to generate more realistic defect samples. Furthermore, defects can potentially exists on various backgrounds. Due to the rarity of defect samples, we can only collect specific defects on a very limited number of backgrounds. As a result, typical image synthesis methods lack defect transferablity, i.e., they can only synthesize defect samples under a constrained number of contexts. Our proposed layer-wise composition strategy can mitigate this issue. This is because it is able to sufficiently preserve the identities (appearances, styles, etc.) of backgrounds, which forces the model to simulate how defects would interact with the exact provided backgrounds. This significantly improves the defect transferability, which means our model is capable of generating new defect samples within variously different backgrounds.

Training Objective. To generate visually realistic images, we adopt adversarial loss to make the generated defect d indistinguishable from real defect sample d_{real}.

$$\mathcal{L}_{adv} = \min_G \max_{D_{src}} \mathbb{E}_{d_{real}}[log D_{src}(d_{real})] \\ + \mathbb{E}_d[log(1 - D_{src}(d))] \tag{5}$$

Our Defect-GAN aims to generate defects conditioned on target defect label $c \in R^C$. To make the generated defects align with the target category, we impose a category classification loss, which consists of two terms: $\mathcal{L}_{cls}^r$ to optimize D by classifying real defect sample d_{real} to the corresponding category c', and $\mathcal{L}_{cls}^f$ to optimize G to generate defect sample of target category c.

$$\mathcal{L}_{cls}^r = \mathbb{E}_{d_{real}, c'}[-log(D_{cls}(c'|d_{real}))] \tag{6}$$

$$\mathcal{L}_{cls}^f = \mathbb{E}_{d,c}[-log(D_{cls}(c|d))] \quad (7)$$

Additionally, we impose a reconstruction loss $\mathcal{L}_{rec}$ that helps preserve the content of input images as much as possible. We adopt L1 loss for the reconstruction loss.

$$\mathcal{L}_{rec} = \mathbb{E}_{n,\hat{n}}[||n - \hat{n}||_1] \quad (8)$$

The layer-wise composition strategy will generate spatial distribution maps in both defacement and restoration process to guide the final compositions. We further improve composition by introducing two additional spatial constraints (beyond spatial distribution maps), namely, a cycle-consistency loss and a region constrain loss.

To precisely restore the generated defect samples to normal samples, the repaint spatial distribution map shall be ideally the same as the defect spatial distribution map. Thus, we design a spatial distribution cycle-consistency loss $\mathcal{L}_{sd-cyc}$ between the defect spatial distribution map and the repaint spatial distribution map.

$$\mathcal{L}_{sd-cyc} = \mathbb{E}_{m_n, m_{\hat{n}}}[||m_n - m_{\hat{n}}||_1] \quad (9)$$

Meanwhile, to avoid the defect foreground and the repaint foreground to take over the whole image area, we introduce a region constrain loss $\mathcal{L}_{sd-con}$ to penalize excessively large defect and foreground distribution maps:

$$\mathcal{L}_{sd-con} = \mathbb{E}_{m_n, m_{\hat{n}}}[||m_n - 0||_1 + ||m_{\hat{n}} - 0||_1] \quad (10)$$

The overall training objectives for G and D are:

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}^r \mathcal{L}_{cls}^r \quad (11)$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}^f \mathcal{L}_{cls}^f + \lambda_{rec} \mathcal{L}_{rec} \\ + \lambda_{con} \mathcal{L}_{sd-cyc} + \lambda_c \mathcal{L}_{sd-con} \quad (12)$$

where λ_{cls}^r, λ_{cls}^f, λ_{rec}, λ_{sd-cyc}, λ_{sd-con} are hyperparameters that are empirically set as 2.0, 5.0, 5.0, 5.0, 1.0, respectively.

3.2. Boosting Defect Inspection Performance

The large amounts of defect samples generated by the aforementioned Defect-GAN can be further used to train the state-of-the-art visual recognition models for defect inspection. We adopt the most commonly used image recognition models ResNet [15] and DenseNet [17] to perform defect inspection. The generated defect samples are mixed with the original dataset to train the recognition models.

However, we notice that although Defect-GAN can synthesize realistic defect samples, there still exists a domain gap between the generated samples and the original samples. Naively training a recognition model over the augmented data will lead the model to learn to distinguish
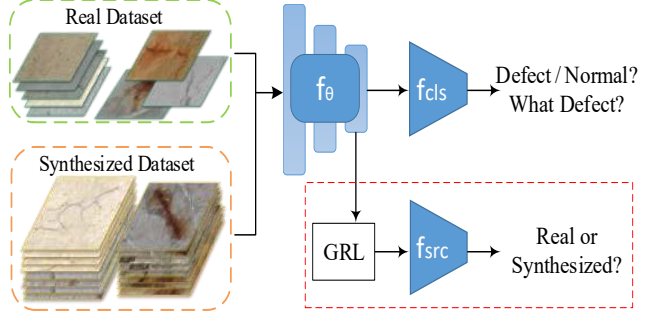

Figure 3. We introduce a source classifier f_{src} (connected to the network backbone f_θ through a Gradient Reversal Layer) for explicitly distinguishing synthesized and real samples. With this the defect inspection network will not learn for such task undesirably.

these two domains undesirably. We attach an additional source classifier f_{src} to distinguish synthesized samples from real ones explicitly, and connect this domain classifier to the network backbone through a Gradient Reversal Layer (GRL) [11] as illustrated in Fig.3. Therefore, there will be no distinguishable difference between the features extracted by f_θ for the synthesized samples and the real samples, which ensures all training data are effectively learnt.

4. Experiments

This section presents experimentation of our methods. We first evaluate Defect-GAN's defect synthesis performance, and then demonstrate its capacity in boosting defect inspection performance as a data augmentation method.

Dataset. We evaluate Defect-GAN on CODEBRIM[1] [38] – a defect inspection dataset in context of concrete bridges, which features six mutually non-exclusive classes: crack, spallation, efflorescence, exposed bars, corrosion and normal samples. It provides image patches for multi-label classification as well as the full-resolution images from which image patches are cropped. Compared with existing open datasets for defect inspection [49, 65, 34], CODEBRIM is the most challenging and complex one to the best of our knowledge, which can better reflect the practical scenarios.

4.1. Defect Synthesis

Implementation Details. We use all images from the classification dataset to train Defect-GAN. Besides, we collect extra 50,000 normal image patches by simply cropping from the original full-resolution images. All images are resized to 128×128 for training. To stabilize the training and generate better images, we replace Eq. 5 with Wasserstein GAN objective with gradient penalty [2, 14] and perform one generator update every five discriminator updates. We use Adam optimizer [27] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$

---

[1]Dataset available at https://doi.org/10.5281/zenodo.2620293

Methods	FID Scores ↓
StackGAN++ [72]	111.1
Conditional StackGAN++ [72]	132.1
StyleGAN v2 [25]	148.2
StyleGAN v2 [25] + DiffAug [73]	142.4
CycleGAN [75]	94.5
StarGAN [7]	295.1
StarGAN [7] + SPADE [43]	103.0
Defect-GAN (Ours)	**65.6**
Ideal Defect Synthesizer	25.0

Table 1. Quantitative comparison of Defect-GAN with existing image synthesis methods in Fréchet Inception Distance (FID).

Design Choices				FID Scores ↓
SCC	ANI	LWC	SC	
×	×	×	×	295.1
✓	×	×	×	103.0
✓	✓	×	×	99.7
✓	×	✓	×	76.8
✓	×	✓	✓	69.5
✓	✓	✓	✓	**65.6**

Table 2. Ablation studies of the proposed Defect-GAN: Our designed Spatial and Categorical Control (SCC), Adaptive Noise injection (ANI), Layer-Wise Composition (LWC), and additional Spatial Constraints (SC) are complementary and jointly beneficial to the quality of the synthesized defects.

to train Defect-GAN with the learning rate starting from 2×10^{-4} and reducing to 1×10^{-6}. We set batch size at 4 and the total training iteration at 500,000. The training takes about one day on a single NVIDIA 2080Ti GPU.

Evaluation Metric. We adopt the commonly used Fréchet Inception Distance (FID) [16] to evaluate the realism of synthesized defect samples. Lower FID scores indicate better synthesis realism.

Quantitative Experimental Results. Table 1 show quantitative experimental results regarding defect synthesis fidelity, in which the first block includes direct synthesis methods (image synthesis from a randomly sampled latent code), and the second block includes image-to-image translation methods. We also present the FID score of a perfect defect synthesizer in the third block by randomly separating the real defect samples into two sets and computing the FID score between them. As Table 1 shows, the direct synthesis methods generally have unsatisfactory performances due to the lack of defect training samples as well as their limited capacities to capture the complex and irregular patterns of defects. As a comparison, by mimicking the defacement and restoration processes following Defect-GAN, existing image-to-image translation methods can generate defects with significantly better quality. This is because, with more information as input, such methods are generally more data-efficient. Besides, they can utilize the large amount of normal samples in training. On the other hand, Defect-GAN achieves significantly better synthesis FID, which demonstrate its superiority in defect synthesis. Interestingly, models with categorical control tend to perform worse in terms of FID scores than models without. We believe introducing additional categorical control can limit model's synthesis realism. However, even with such constraint, Defect-GAN still achieves the best performance.

We further demonstrate the effectiveness of our proposed designs in Defect-GAN by presenting quantitative ablative experiments in Table 2. Without our designed components,

Defect-GAN degrades to StarGAN [7] – a widely used multi-domain image-to-image translation model. However, it fails to converge on this task and cannot synthesize any defect-like patterns. By incorporating Spatial and Categorical Control (SCC), it can converge and generate defect samples with comparable quality with existing methods. Based on this, the Layer-Wise Composition (LWC) can significantly improve the synthesis realism. We believe the reason is twofold: (1) it lifts the defect-insufficiency constraint by allowing the networks to fully focus on defect generation; (2) it can generate contextually more natural defects. Furthermore, Adaptive Noise Injection (ANI) and additional Spatial Constraints (SC) for training can also boost defect synthesis performance. These proposed components are proved to be complementary to each other, enabling Defect-GAN to achieve state-of-the-art defect synthesis quality.

Qualitative Experimental Results. Fig. 4 shows qualitative results of Defect-GAN and comparisons with other synthesis methods. Rows 1-2 show the synthesis by state-of-the-art direct synthesis methods: StackGAN++ [72] and StyleGAN v2 [25] with DiffAug [73]. We can see that many generated samples do not contain clear defects, and some samples are not visually natural. This verifies the aforementioned limitation of GANs for defect synthesis. For image-to-image translation methods, we choose StarGAN [7] with SPADE [43] as the competing method since it offers categorical control as Defect-GAN. And other methods like CycleGAN [75] and SDGAN [40] produce visually similar results. As shown in Row 4-5, StarGAN w/ SPADE and Defect-GAN can produce visually realistic and diverse defect samples conditioned on normal samples. Defect samples by StarGAN w/ SPADE look comparable with Defect-GAN, except that it tends to alter the background identity, while Defect-GAN can preserve the appearance and style of normal samples thanks to the layer-wise composition strategy. On the other hand, StarGAN w/ SPADE completely fails to transfer the learnt defect patterns to novel
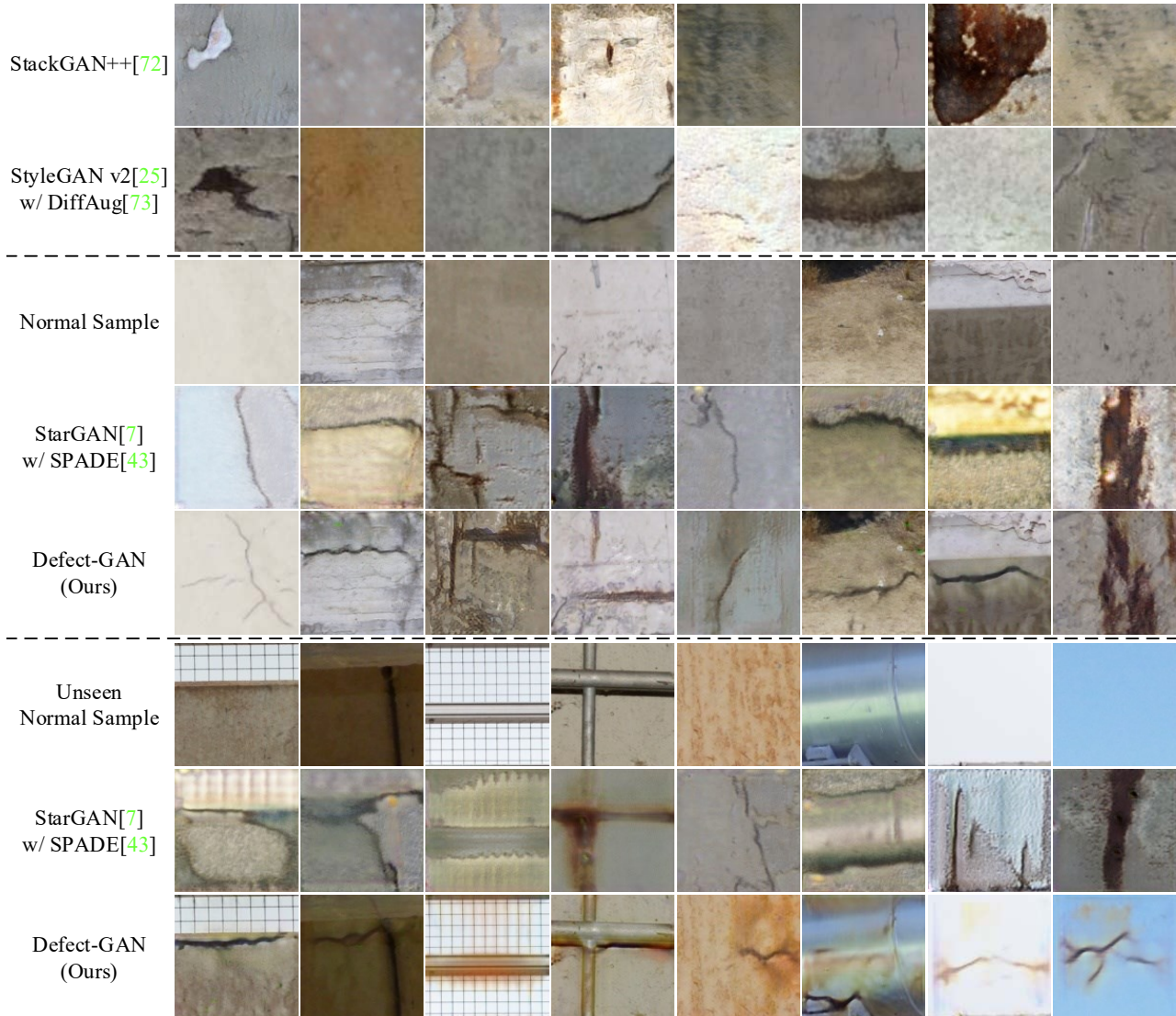
Figure 4. Qualitative comparison of Defect-GAN with state-of-the-art image synthesis methods: Rows 1-2 show direct defect synthesis from random noises by two latest image synthesis methods. Rows 4-5 compare defect generation over *Normal Samples* in Row 3 (used in network training) by StarGAN with SPADE and our Defect-GAN, while Rows 7-8 compare defect generation over *Unseen Normal Samples* in Row 6 (not used in network training) by StarGAN with SPADE and our Defect-GAN.

backgrounds that are not seen during training, while Defect-GAN shows superb defect transferability and synthesis realism as shown in Rows 7-8. This property is essential for introducing new information into the training data.

In addition, we show Defect-GAN's categorical control in defect generation in Fig. 5, where different types of defects can be generated conditioned on the same normal image. Fig. 6 also shows Defect-GAN's spatial control in defect generation, where red boxes denote the intended places to generate defects. Defect-GAN can generate defects on specific locations while maintaining contexts natural.

4.2. Defect Inspection

Implementation Details. We use the training set and additional 50,000 normal images to train Defect-GAN. Then, Defect-GAN expands the training samples by synthesizing 50,000 defect samples. The generated defect samples are mixed with the original training data to train the defect inspection networks, with the extra restored normal samples also included to avoid data imbalance. All images are resized to 224×224. We adopt SGD with a learning rate of 1×10^{-3} to train the network until convergence. Batch size is set to 16. We use the validation set to select the best performing model and report the performance on the test set.
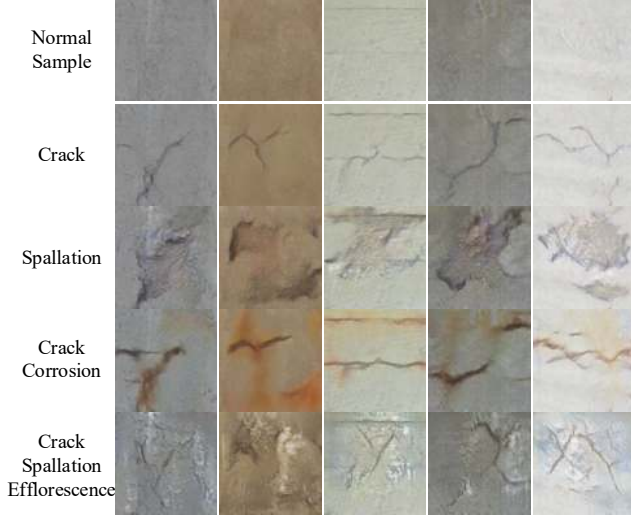
Figure 5. Illustration of categorical control in defect generation by Defect-GAN: For each normal sample in Row 1, Rows 2-3 and 4-5 show the generated defect samples conditioned on a single and multiple target categories, respectively.

Figure 6. Illustration of spatial control in defect generation by Defect-GAN: Each row shows defect samples generated with different normal samples but the same spatial control, while each column shows defect samples generated with the same normal sample but different spatial controls.

Quantitative Experimental Results. As CODEBRIM features a multi-label classification task, we can only adopt methods with categorical control to expand the training samples. Results for defect inspection is shown in Table 3. The first row of each block shows defect inspection performance only with original training data, and the rest three rows of each block present defect inspection performance with original training data and the augmented samples generated by different synthesis methods. For fair comparison,

Networks	Augmentation Methods	Accuracy(%)
ResNet34 [15]	None	70.25
	Conditional StackGAN++[72]	62.59
	StarGAN[7]+SPADE[43]	71.90
	Defect-GAN (Ours)	**75.48**
DenseNet121 [17]	None	70.77
	Conditional StackGAN++[72]	58.68
	StarGAN[7]+SPADE[43]	72.61
	Defect-GAN (Ours)	**75.79**

Table 3. Quantitative experimental results for defect inspection.

50,000 synthesised defect samples are augmented for all synthesis methods. As the results shows, the synthesized defect samples from Conditional StackGAN++ [72] greatly drop the defect inspection performance. This is because that StackGAN++ is not even able to generate realistic defect samples due to its limited capacity in defect modeling. StackGAN++ generated defect samples are harmful to network training. On the other hand, StarGAN[7]+SPADE[43] generated samples can slightly boost the inspection performance. And our proposed Defect-GAN can further significantly improve the accuracy of trained defect inspection networks. Although both methods can generate defect samples with good visual realism, our proposed Defect-GAN is capable of simulating the learnt defects on backgrounds that are not seen during training. This feature makes Defect-GAN generated samples much more diverse, thus can introducing new information into the training data, significantly improving the performance of trained models. The results also demonstrate the superiority of Defect-GAN for defect synthesis in terms of fidelity, diversity and transferability.

5. Conclusion

This paper presents a novel Defect-GAN for defect sample generation by mimicking the defacement and restoration processes. It can capture the stochastic variations within defects and can offer flexible control over the locations and categories of the generated defects. Furthermore, with a novel compositional layer-based architecture, it is able to generate defects while preserving the style and appearance of the provided backgrounds. The proposed Defect-GAN is capable of generating defect samples with superior fidelity and diversity, which can further significantly boost the performances of defect inspection networks.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[3] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *ICCV*, pages 4502–4511, 2019.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[5] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[6] Ying-Cong Chen, Xiaohui Shen, Zhe Lin, Xin Lu, I Pao, Jiaya Jia, et al. Semantic component decomposition for face attribute manipulation. In *CVPR*, pages 9859–9867, 2019.

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018.

[8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020.

[9] Shahrzad Faghih-Roohi, Siamak Hajizadeh, Alfredo Núñez, Robert Babuska, and Bart De Schutter. Deep convolutional neural networks for detection of rail surface defects. In *International joint conference on neural networks (IJCNN)*, pages 2584–2589. IEEE, 2016.

[10] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *CVPR*, 2020.

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[12] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR 2018*, pages 4367–4375, 2018.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein GANs. In *NeurIPS*, pages 5767–5777, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, pages 6626–6637, 2017.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[18] Qian Huang, Yuan Wu, John Baruch, Ping Jiang, and Yonghong Peng. A template model for defect simulation for evaluating nondestructive testing in x-radiography. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(2):466–475, 2009.

[19] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[22] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, pages 8420–8429, 2019.

[23] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, pages 8110–8119, 2020.

[26] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[28] Ali Koksal and Shijian Lu. Rf-gan: A light and reconfigurable network for unpaired image-to-image translation. In *ACCV*, 2020.

[29] Chung-Feng Jeffrey Kuo, Chien-Tung Max Hsu, Zong-Xian Liu, and Han-Cheng Wu. Automatic inspection system of led chip using two-stages back-propagation neural network. *Journal of Intelligent Manufacturing*, 25(6):1235–1243, 2014.

[30] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017.

[31] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

[32] Yundong Li, Weigang Zhao, and Jiahao Pan. Deformable patterned fabric defect detection with fisher criterion-based deep learning. *IEEE Transactions on Automation Science and Engineering*, 14(2):1256–1264, 2016.

[33] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, pages 700–708, 2017.

[34] Marc Maguire, Sattar Dorafshan, and Robert J. Thomas. SDNET2018: A concrete crack image dataset for machine learning applications. 2018.

[35] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*, pages 3693–3703, 2018.

[36] Domingo Mery and Dieter Filbert. Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence. *IEEE Transactions on Robotics and Automation*, 18(6):890–901, 2002.

[37] D Mery, D Hahn, and N Hitschfeld. Simulation of defects in aluminium castings using cad models of flaws and real x-ray images. *Insight-Non-Destructive Testing and Condition Monitoring*, 47(10):618–624, 2005.

[38] Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the COncrete DEfect BRidge IMage dataset. In *CVPR*, pages 11196–11205, 2019.

[39] Henry YT Ngan, Grantham KH Pang, and Nelson HC Yung. Automated fabric defect detection—a review. *Image and vision computing*, 29(7):442–458, 2011.

[40] Shuanlong Niu, Bin Li, Xinggang Wang, and Hui Lin. Defect image sample generation with GAN for improving defect recognition. *IEEE Transactions on Automation Science and Engineering*, 2020.

[41] Shuanlong Niu, Hui Lin, Tongzhi Niu, Bin Li, and Xinggang Wang. DefectGAN: Weakly-supervised defect detection using generative adversarial network. In *CASE*, pages 127–132, 2019.

[42] Dim P Papadopoulos, Youssef Tamaazousti, Ferda Ofli, Ingmar Weber, and Antonio Torralba. How to make a pizza: Learning a compositional layer-based gan model. In *CVPR*, pages 8002–8011, 2019.

[43] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019.

[44] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016.

[45] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. GANimation: Anatomically-aware facial animation from a single image. In *ECCV*, pages 818–833, 2018.

[46] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[48] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, pages 4491–4500, 2017.

[49] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[50] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. FineGAN: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*, pages 6490–6499, 2019.

[51] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS 2017*, pages 4077–4087, 2017.

[52] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR 2018*, pages 1199–1208, 2018.

[53] AS Tolba and Hazem M Raafat. Multiscale image quality measures for defect detection in thin films. *The International Journal of Advanced Manufacturing Technology*, 79(1-4):113–122, 2015.

[54] D-M Tsai and C-Y Hsieh. Automated surface inspection for directional textures. *Image and Vision computing*, 18(1):49–62, 1999.

[55] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. 2020.

[56] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, pages 7278–7286, 2018.

[57] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. *ICCV*, pages 9924–9933, 2019.

[58] Zheng Wang, Mang Ye, Fan Yang, Xiang Bai, and Shin'ichi Satoh. Cascaded sr-gan for scale-adaptive low resolution person re-identification. In *IJCAI*, pages 3891–3897, 2018.

[59] Rongliang Wu and Shijian Lu. LEED: Label-free expression editing via disentanglement. In *ECCV*, pages 781–798, 2020.

[60] Rongliang Wu, Gongjie Zhang, Shijian Lu, and Tao Chen. Cascade EF-GAN: Progressive facial expression editing with local focuses. In *CVPR*, pages 5021–5030, 2020.

[61] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-VAEGAN-D2: A feature generating framework for any-shot learning. In *CVPR*, pages 10275–10284, 2019.

[62] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta R-CNN: Towards general solver for instance-level low-shot learning. In *ICCV*, 2019.

[63] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *ECCV*, pages 1–17, 2018.

[64] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. LR-GAN: Layered recursive generative adversarial networks for image generation. In *ICLR*, 2017.

[65] Liang Yang, Bing Li, Wei Li, Zhaoming Liu, Guoyong Yang, and Jizhong Xiao. Deep concrete inspection using unmanned aerial vehicle towards CSSC database. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017.

[66] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, pages 5485–5493, 2017.

[67] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018.

[68] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. 2018.

[69] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion GAN for image synthesis. In *CVPR*, pages 3653–3662, 2019.

[70] Gongjie Zhang, Kaiwen Cui, Rongliang Wu, Shijian Lu, and Yonghong Tian. PNPDet: Efficient few-shot detection without forgetting via plug-and-play sub-networks. In *WACV*, 2021.

[71] Gongjie Zhang, Shijian Lu, and Wei Zhang. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):10015–10024, 2019.

[72] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.

[73] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020.

[74] Wenju Zhou, Minrui Fei, Huiyu Zhou, and Kang Li. A sparse representation based fast detection method for surface defect detection of bottle caps. *Neurocomputing*, 123:406–414, 2014.

[75] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017.

[76] Zhen Zhu, Zhiliang Xu, Ansheng You, and Xiang Bai. Semantically multi-modal image synthesis. In *CVPR*, pages 5467–5476, 2020.