IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 August 2023, accepted 5 October 2023, date of publication 9 October 2023, date of current version 13 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322927

RESEARCH ARTICLE

Synthesizing Industrial Defect Images Under Data Imbalance

EUNHEE CHO[1], (Student Member, IEEE), BYEONGHWAN JEON[2], (Member, IEEE), AND IN KYU PARK[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea
[2]Artificial Intelligence Convergence Research Center, Inha University, Incheon 22212, South Korea

Corresponding author: In Kyu Park (pik@inha.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean Government (MSIT) through the Artificial Intelligence Convergence Innovation Human Resources Development, Inha University under Grant RS-2022-00155915; and in part by Inha University Research Grant.

ABSTRACT Defect detection is a crucial technology in the industry that enhances production efficiency within the manufacturing sector. However, obtaining a balanced dataset with sufficient samples of both normal and defect is often challenging and time-consuming. Constructing an unbalanced dataset skewed toward normal samples results in decreased performance and reduced generalization of trained models. Therefore, building an appropriate dataset is essential for effectively training deep models. In this study, we propose a defect image augmentation technique based on generative adversarial networks (GANs), dubbed SyNDGAN, to address the challenges of unbalanced datasets encountered in real-world manufacturing scenarios. Specifically, our SyNDGAN synthesizes defect samples from normal images with given segmentation maps which contain the defect types and location of the defect. We validate our method by utilizing manufacturer data which considers the industrial scenario, with limited data. In our experiments, the proposed method shows superior quality compared to other methods both quantitatively and qualitatively. Furthermore, we demonstrate that synthesized data helps to improve the defect recognition performance, which can be utilized in real-world scenarios.

INDEX TERMS Defect synthesis, generative adversarial networks, augmentation, classification.

I. INTRODUCTION

Deep learning is currently being actively pursued, with a focus on its practical applications in various industries. Defect inspection is especially crucial in manufacturing, as it directly affects the productivity and profits. Convolutional neural networks (CNNs) [1] have shown potential in automation technologies. They can identify defects that may be difficult for humans to detect, thereby helping improve the production efficiency and accuracy.

However, deep learning requires a large amount of data for training, and imbalance in defect data is often a significant issue. Defects in the manufacturing processes typically occur rather intermittently; therefore, compared to normal data, securing a sufficient quantity and variety of labeled defect datasets is challenging and costly. To address this issue, previous works [2], [3] have proposed using Generative Adversarial Networks (GANs) to synthesize defect images in industrial manufacturing. SDGAN [2] is designed to make comprehensive use of defect-free industrial images for defect sample generation from commutator cylinder surface image data sets. Defect-GAN [3] also provides an automated defect synthesis network that generates realistic and diverse concrete bridge defect samples to train an accurate and robust defect inspection network. However, generating high-resolution defect images and guaranteeing training stability in limited or unbalanced data situations still remains challenging.

In this paper, we propose a novel method for synthesizing defect images from normal images by using user-provided masks. Our method considers real-world scenarios, which are applicable to limited data situations. First, we fine-tune a pre-trained GAN using normal and defect data. Second, by utilizing the pre-trained representations, we train the

The associate editor coordinating the review of this manuscript and approving it for publication was Miaohui Wang.

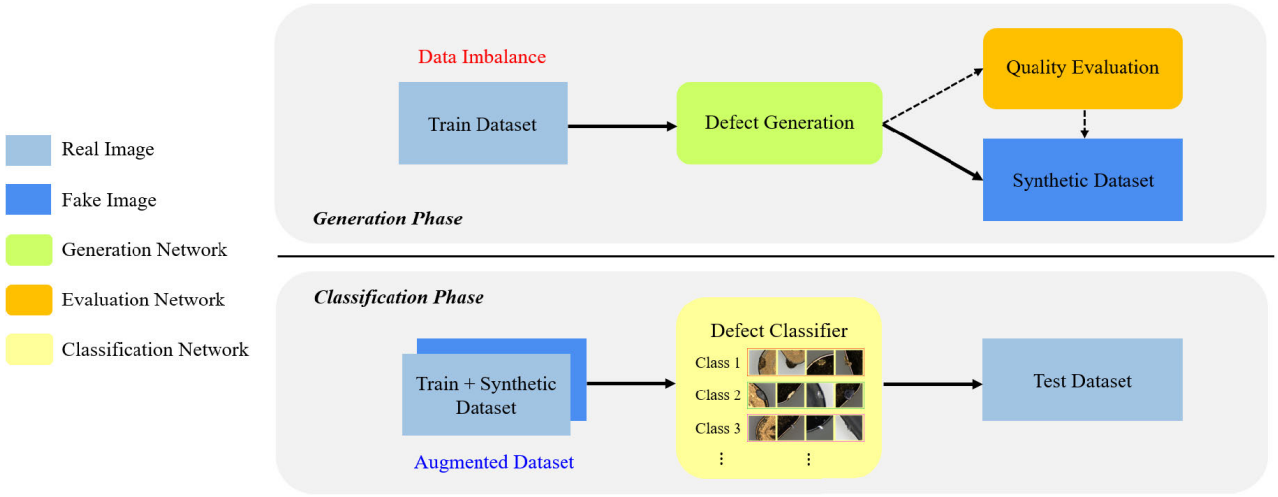


FIGURE 1. Proposed model to address the issue of class imbalance between normal and defect data. The model synthesizes defect samples by utilizing normal samples and defect masks. Augmenting the dataset with the synthesized data enhances the performance of the classification model.

normal-to-defect synthesizing model using a mask that contains the location and type of defect. We empirically demonstrate that the proposed method is scalable for synthesizing defects, and produces better results with both high quality and diversity than the previous methods. Next, we combine the synthesized defect images to train the dataset for classifying the defects.

Figure 1 provides an overview of the proposed methodology. Our method leverages synthetic data to enhance the defect classification performance. Furthermore, our approach considers limited data situations that occur in many real-world applications.

Our contributions can be summarized as follows.

- We propose an effective data augmentation method that can overcome the imbalance in the training data in manufacturing sector by using a GAN-based model.
- We augment the training dataset with the generated synthetic data for learning with a deep learning model as well as to establish the improved performance of the classification model.
- We demonstrate the superiority of the proposed technique via quantitative and qualitative comparisons with existing data synthesis techniques.

This paper is organized as follows. Section II introduces the related work. The proposed deep model is described in Section III. In Section IV and Section V, experimental details and results are provided extensively. We give a conclusive remark in Section VI.

II. RELATED WORK

A. DEFECT INSPECTION

In manufacturing industries, defect inspection is crucial for improving the production process through detecting and classifying defects. There are two main approaches to defect inspection: machine vision-based and deep learning-based methods. The machine vision-based methods involve several steps, such as image pre-processing, feature extraction and selection, and image recognition. These methods require expert knowledge and are not very robust. In recent years, deep learning-based methods have received significant attention owing to their strong learning capability and ability to learn higher-order features from data. However, these methods demand large amounts of training data; this requirement poses practical hindrance to introducing deep learning inspection systems in actual production lines; moreover, the occurrence rate of defective products is lower than that of normal products, which results in a data imbalance problem.

Class imbalance [4], [5] is a common issue in machine learning-based research. Cost-sensitive learning [6], [7] was proposed to address this problem. It balanced the learning by weighting a small number of classes during the training rather than generating new data. One way to achieve this was by applying focal loss in a one-stage detector that handled both localization and classification simultaneously. Focal loss [8] can give more weight to difficult or misclassified cases and reduce the loss. However, a limitation of this methodology is that its performance may not be consistent across different datasets and imbalance situations.

Anomaly detection [9] technology has been proposed to address the class imbalance problem in deep learning-based defect inspection. Anomaly detection estimates the distribution of normal data during the learning phase and classifies distributions that deviate from this as abnormal data. However, as abnormal data generated in actual industrial sites have diverse characteristics, and it is difficult to improve the process by identifying the accuracy and defect types. To address these limitations, studies have proposed augmenting defect data required for training deep learning models. For example, Niu et al. [2] proposed an approach that simultaneously generated defect images from actual normal

images and generated normal images from actual defect images through SDGAN. Zhang et al. [3] further controlled the location and type of defects by injecting a spatial and categorical control map into the decoder part for the normal image input. These approaches generated various realistic defects; however, they had limitations in creating complex defects at high resolutions.

B. IMAGE SYNTHESIS

GAN is an unsupervised learning-based model that involves two different networks, a generator and a discriminator, engaging in adversarial learning to create data similar to the reality. GAN was first proposed by Goodfellow et al. [10] and has performed excellently in various fields, including text and video generation [11], [12]. However, the training process for GAN is unstable and suffers from phenomena such as mode collapse [13]. Various methodologies have been proposed to increase the training stability and performance of GANs, including deep convolutional GAN [14], conditional GAN [15], least square GAN [16], Wasserstein GAN [17], progressive GAN [18], StyleGAN [19], and StyleGAN2 [20].

More recently, GANs have been developed further to manipulate the semantic properties of real images to obtain more realistic synthesized images. Zhu et al. [21] proposed a method which inverted a given real image into the latent space, reconstructed the input image from it and generated semantic manipulation. Kim et al. [22] created a style map with spatial information from latent vectors through a mapping network composed of fully-connected layers, and adjusted the style of the resulting image in the synthesis network through affine transforms.

However, when the data size is limited, the performance of GAN tends to significantly decrease. To address this problem, StyleGAN2-ADA [23], a model designed to enable stable training with less data, was proposed. In conclusion, GANs have shown significant potential in generating high-quality data while securing diversity; however, further development is needed for training with limited data.

C. GANS WITH LIMITED DATA

GAN models have shown excellent performance in image generation tasks. However, they require a large amount of training data, ranging from tens of thousands to hundreds of thousands of elements, to avoid overfitting and ensure stable learning. Traditional data augmentation techniques are not suitable for GANs as they can cause information leakage to the generator.

To address this, the consistency regularization GAN (CRGAN) [24] was proposed, which applied a consistency regularization term to the discriminator during learning, thereby ensuring that it applied the same discrimination for the images. This method was effective, stable, and required less computation. The balanced consistency regularization GAN (bCRGAN) [25] applied consistency regularization to both real and generated images; however, the model still suffered from reinforcement leakage.

StyleGAN2-ADA overcomes the limitations of existing methods by training the discriminator and generator using only augmented images, enabling the model to learn from a small number of images while significantly improving the image quality. FUNIT [26] and FastGAN [27] are two other methods that can generate high-quality images with very little data. Attention-based modules and a self-supervised discriminator can mitigate mode collapse and improve the robustness and generalization performance of the model.

ProjectedGAN [28] combines the FastGAN generator structure with random projection and a multi-resolution discriminator to more effectively utilize the deep layer features in pre-trained models. By mixing feature maps with channel and resolution, the model demonstrated superior performance in terms of image quality and convergence speed compared to other models.

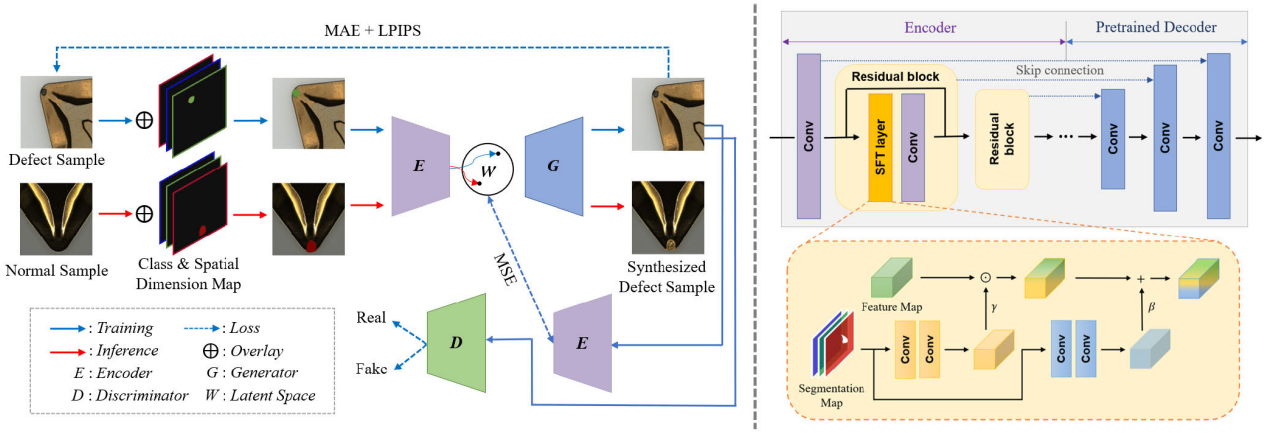
The previously mentioned studies demonstrate notable improvement in training GANs within data limitations. However, their effectiveness has primarily been observed in constrained domains like facial and landscape imagery, making their direct application to complex, high-resolution industrial datasets less straightforward. To overcome this constraint, our study employs domain-specialized GANs, expanding their utility into real-world industries. Our method enables the generation of diverse defect data for training deep learning-based inspection models, effectively addressing the challenge of data scarcity.

III. PROPOSED METHOD

A. OVERVIEW

In this paper, we introduce a GAN-based defect image augmentation method called the Synthesizing Normal-to-Defect GAN (SyNDGAN) framework, designed to tackle the challenge of imbalanced datasets. Our approach involves synthesizing a wide range of realistic defect images from normal data.

Overall architecture of the proposed synthesis network is presented in Figure 2. Auto-encoder model is employed to learn extracting and expressing defect features. During training, a real defect image masked by a corresponding segmentation map is fed to the model. The segmentation map has RGB channels, and each defect class is assigned with a specific color as shown in Figure 2 (a). By overlaying the defect segmentation map onto a defect image, it would be easier to synthesize defects for normal images during the inference. When the overlaid image is fed as input, encoder projects it into latent space. Then, defect image is synthesized by decoder. Real samples are encoded into latent vectors by the encoder E, which are then mapped onto the latent space. By leveraging a pre-trained decoder D, the model generates highly realistic images by decoding these latent representations. This encoder-based approach enables the synthesis of images that closely align with the distribution of real samples, resulting in visually compelling outputs.



(a) Overall Architecture (b) Synthesis Network Detail

FIGURE 2. Architecture of proposed defect synthesis network, *i.e.* SyNDGAN. During the training (blue solid arrows), our proposed method synthesizes a defect image by inputting a defect sample and a corresponding segmentation map based on auto-encoder learning. In the inference (red solid arrows), a defect image is synthesized by inputting a normal sample and a segmentation map that specifies the type and location of the defect. Each defect class is assigned a specific color mask.

B. FRAMEWORK DETAILS

1) NETWORK ARCHITECTURE

Our framework consists of a generator and a discriminator. The generator has an encoder-decoder architecture with skip-connections [29], synthesizing defect data from normal samples. Specifically, the SyNDGAN's encoder network is structured with one convolution layer and seven residual blocks. On the other hand, the decoder is a pretrained generator, strategically leveraging its high-quality generation capabilities to effectively overcome data limitation challenges.

2) PRE-TRAINED GENERATOR

The backbone model is chosen to efficiently learn from limited defect data. We utilize ProjectedGAN [28], a data-efficient model capable of robust performance even with a small dataset. After comparing two generators, StyleGAN2 [20] and FastGAN [27] pre-trained on FFHQ [19] dataset, we select the former for its superior performance. Our model is trained using a pre-trained generator on the entire dataset, including normal and defect data.

3) DEFECT SYNTHESIS

In the synthesis process shown in Figure 2 (b), the input image is convolved once and passed through several residual blocks. Each block includes a layer that performs spatial-feature transform (SFT) [30] using the segmentation probability map to modulate the spatial unit feature with affine transformation. Within each layer, the segmentation map is fed to two separate convolutional layers. The first convolution layer includes a Sigmoid at the end and outputted 'scaling' (γ). The other layer has a simple structure producing 'shifting' (β). The transformation is carried out by γ and β in each feature map.

The upsampling is performed by a pre-trained decoder (G). After the input image is encoded to latent space (W), the decoder generates an output image with an identical resolution as the input image. Consequently, by utilizing a segmentation map to assign weights to the defect area of the input defect image, the model could focus on the defect portion during the synthesis.

For the generated image, the discriminator utilizes EfficientNet [31], a pre-trained feature network, to obtain features for the synthesized image from the corresponding layer at each resolution. It associates each discriminant with the features of that layer and used a simple convolutional architecture. All discriminators output predictions at the same resolution and are summed together. Using these pre-trained multi-resolution discriminators can improve performance when extracting features from synthetic images, and improves quality when only limited data are available. We also apply differentiable random projection to perform channel-specific and resolution-specific feature mixing. This allows us to weaken the dominant features by applying a discriminator that considers these multi-level resolution features when the defect size is small, as in our dataset. Consequently, the discriminator focuses equally on possible subspaces, including local features, in terms of the semantics of the deeper layers.

C. OBJECTIVE FUNCTION

1) PRE-TRAINED GENERATOR

The adversarial loss $\mathcal{L}_{adv}$ of the pretrained generator in this paper is given as

$$\mathcal{L}_{adv} = \sum_{l=\mathcal{L}} \Big(\mathbb{E}_{x\sim p_{data}}[\log D_l(P_l(x))] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_l(P_l(\hat{x})))] \Big) \quad (1)$$

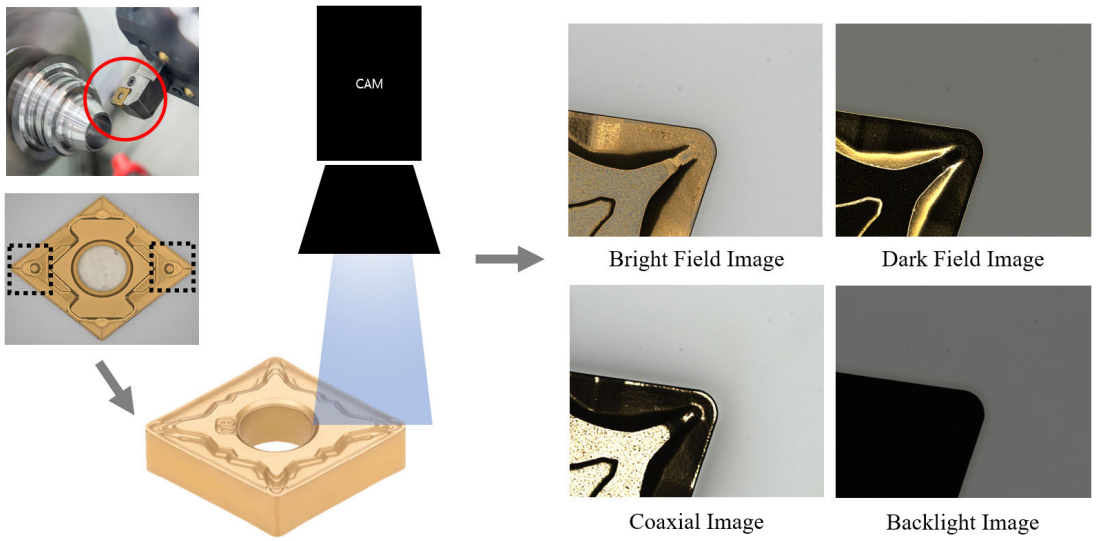


FIGURE 3. Actual product from the manufacturing industry which is used in this study. Four types of images under four different illuminations are shown by enlarging the part that is in contact with the machining surface. They are supplied as input to the defect detection model. "Carbide inserts" image is retrieved from R.D.BARRETT [32].

where D_l represents the discriminator and G, the generator. $\hat{x} = G(z)$ is an image generated from a latent vector through G, and P_l is an EfficientNet model pre-trained on ImageNet [33], which serves as a feature extractor. D_l and P_l operate at multiple resolutions, and features are obtained for each resolution from the four layers of the feature network. Through this, various feature projections can be obtained from each independent D_l.

2) SYNTHESIS NETWORK

The loss function $\mathcal{L}_u$ of the defect synthesis model proposed here is as follows.

$$\mathcal{L}_u = \mathcal{L}_{pix} + \mathcal{L}_{adv} + \mathcal{L}_{lpips} + \mathcal{L}_{inv} \tag{2}$$

where $\mathcal{L}_{pix}$ represents the mean absolute error (MAE) [34], and a pixel-based distance function is expressed as

$$\mathcal{L}_{pix} = ||U(I_{D,M}) - I_D||_1 \tag{3}$$

where U is the proposed model that simultaneously receives a defect image I_D, a mask M containing defect location, and class information as input; it outputs a defect image synthesis result and is trained to reduce the L1 distance to I_D.

$\mathcal{L}_{adv}$ is an adversarial loss that allows a discriminator to determine a defect image generated to perform realistic defect image synthesis as an actual defect image. $\hat{I}_D = U(I_{D,M})$, where D_l represents the discriminator and G, the generator. In this synthesis network, D and G are pre-trained models.

$$\mathcal{L}_{adv} = \sum_{l=\mathcal{L}} \Big(\mathbb{E}_{I_D \sim p(I_D)}[\log D_l(P_l(I_D))] + \mathbb{E}_{\hat{I}_D \sim p(\hat{I}_D)}[\log(1 - D_l(P_l(\hat{I}_D)))] \Big) \tag{4}$$

where $\mathcal{L}_{lpips}$ is a perceptual loss based on the LPIPS [35] model used to prevent blurry image synthesis and perform high-quality image synthesis. The backbone model uses the VGG-16 [36] network and is expressed as

$$\mathcal{L}_{lpips} = \sum_i ||f_i(\hat{I}_D) - f_i(I_D)||_2 \tag{5}$$

where, f_i represents the feature map passing through each i-th layer of the VGG-16 network and applies the L2 distance function between the generated and ground truth images.

$$\mathcal{L}_{inv} = ||Enc(I_{\hat{I}_{D,M}}) - Enc(I_{I_{D,M}})||_2 \tag{6}$$

$\mathcal{L}_{inv}$ is a loss function proposed by In-Domain GAN Inversion [21], used as a type of constraint function for an encoder (Enc), and is expressed as follows.

By comparing the latent vectors of the real input image and the generated image passed through the encoder, the latter remains aligned within the latent space of the input image. Domain knowledge guides the encoder's training, resulting in improved embedding of the encoder.

IV. IMPLEMENTATION DETAILS

A. EXPERIMENTAL SETUP

We train our framework using PyTorch [37] on a single NVIDIA RTX A6000 GPU. Adam [38] optimizer is employed with the following parameter settings: $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learning rates of the generator and discriminator are set to $2{*}10^{-4}$ and $1{*}10^{-4}$ respectively. Pre-training on the entire dataset takes about 1.5 days. Notably, the ProjectedGAN framework, distinguished for its data efficiency, enables training within a reasonable timeframe while utilizing a reduced computational workload.

The resolution of the original insert image is 2440 × 2040. To ensure computational efficiency and enable direct comparisons with other state-of-the-art synthesis models, we conduct experiments at a resolution of

1024×1024. We train the generators of StyleGAN2 (referred to as Projected StyleGAN2) and FastGAN (referred to as Projected FastGAN) and use them as decoders for the image-to-image translation model to synthesize defect images. To conduct the model's performance comparison, we select three image generation models (ADA, Projected StyleGAN2, and Projected FastGAN) and five image translation methods (ADA, Projected FastGAN, CycleGAN, StyleMapGAN, SyNDGAN without a pretraining process).

B. DATASET

1) TRAINING DATA

In this paper, we acquire real product data from an actually existing industrial corporation which manufactures insert inspection equipment. Cemented carbide inserts are consumables used in machining processes to cut metal parts, and are utilized in various sectors of manufacturing, including precision machining and polishing, as well as in industries such as semiconductors, automobiles, medical care, and aviation.

We follow the insert data acquisition process depicted in Figure 3. We obtain data by magnifying the area in contact with the machining surface, which is a significant inspection area that influences the machining outcome and product life. Furthermore, we capture images of the same location under four different illumination conditions such as bright field, dark field, coaxial, and backlight. After classifying the acquired insert data, the obvious imbalance is observed between the number of normal (3,575 samples) and defect images (761 samples). The volume of defect data is approximately 18% of the total. Compared to previous studies [2], [3], [39], our research tackles more challenging scenarios in terms of both data quantity and resolution. This shortage and imbalance of defect data degrade the performance of defect detection models that aim to improve production efficiency by identifying defects in advance.

2) DEFECT SEGMENTATION MAP

The insert dataset used in this study contains fine defects, with a minimum defect size of four pixels at a high resolution. Moreover, automatic defect detection and detected shapes are inconsistent across the four different illuminations during the data acquisition. Therefore, determining a single and consistent defect area requires a comprehensive evaluation of all the four field images. Consequently, it is nontrivial to randomly generate defect images by extracting random vectors from the latent space.

To overcome the limitation, we employ an annotation strategy in a semi-supervised manner. We mark the defect area using an annotation tool based on the defect information specified in the ground truth for the defect image. The corresponding polygon is set to assign a defect label to each defect area and secure the corresponding segmentation map data. A total of 587 segmentation map annotation is conducted through a final approval by the the manufacturer's professional inspector. Using the constructed data, we employ an image-to-image translation technique to synthesize defect images for each field using both the normal image and the defect segmentation map information.

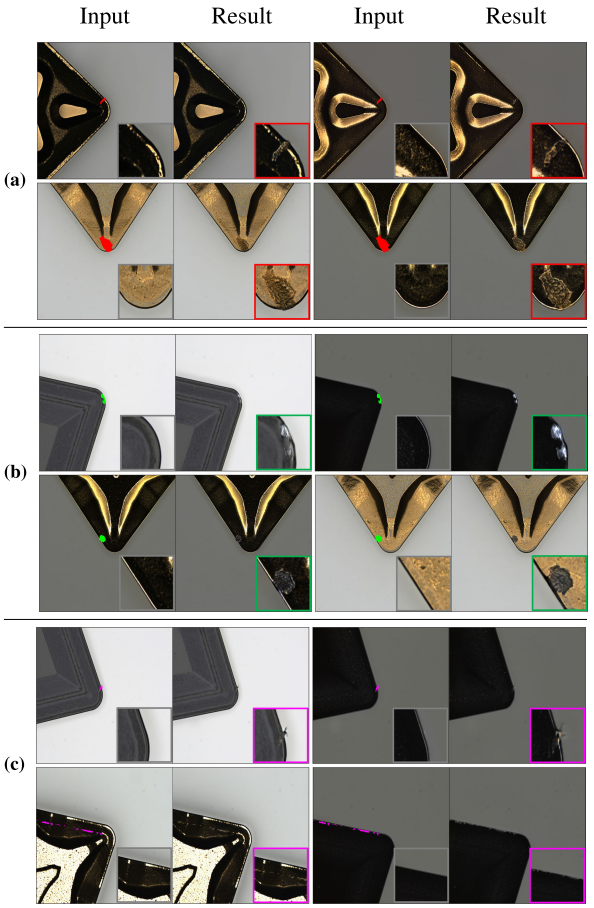


FIGURE 4. Qualitative results for each defect class. (a) Bump, (b) Chipping, and (c) Dust.

C. EVALUATION METRICS

To evaluate the model's performance in both fidelity and diversity of the synthesized images, we employ the FID [40], KID [41], and LPIPS [35] metrics. The FID exhibits bias with test datasets under 20K due to its sensitivity to limited samples and covariance calculations. On the other hand, the KID, utilizing independent samples, is valuable for assessing dissimilarity in such scenarios. To ensure objectivity, we employ both FID and KID metrics in our evaluation. Lower scores in both metrics indicate improved realism and diversity in the generated images. Moreover, LPIPS evaluates image similarity by calculating the feature space distance using the pre-trained VGG network. A lower score indicates higher perceptual alignment between synthesized and reference images.

V. EXPERIMENTAL RESULTS

A. COMPARISON WITH ESTABLISHED MODELS

1) DEFECT SYNTHESIS

Figure 4 shows the defect image results synthesized through our proposed method. When an image is given as an input

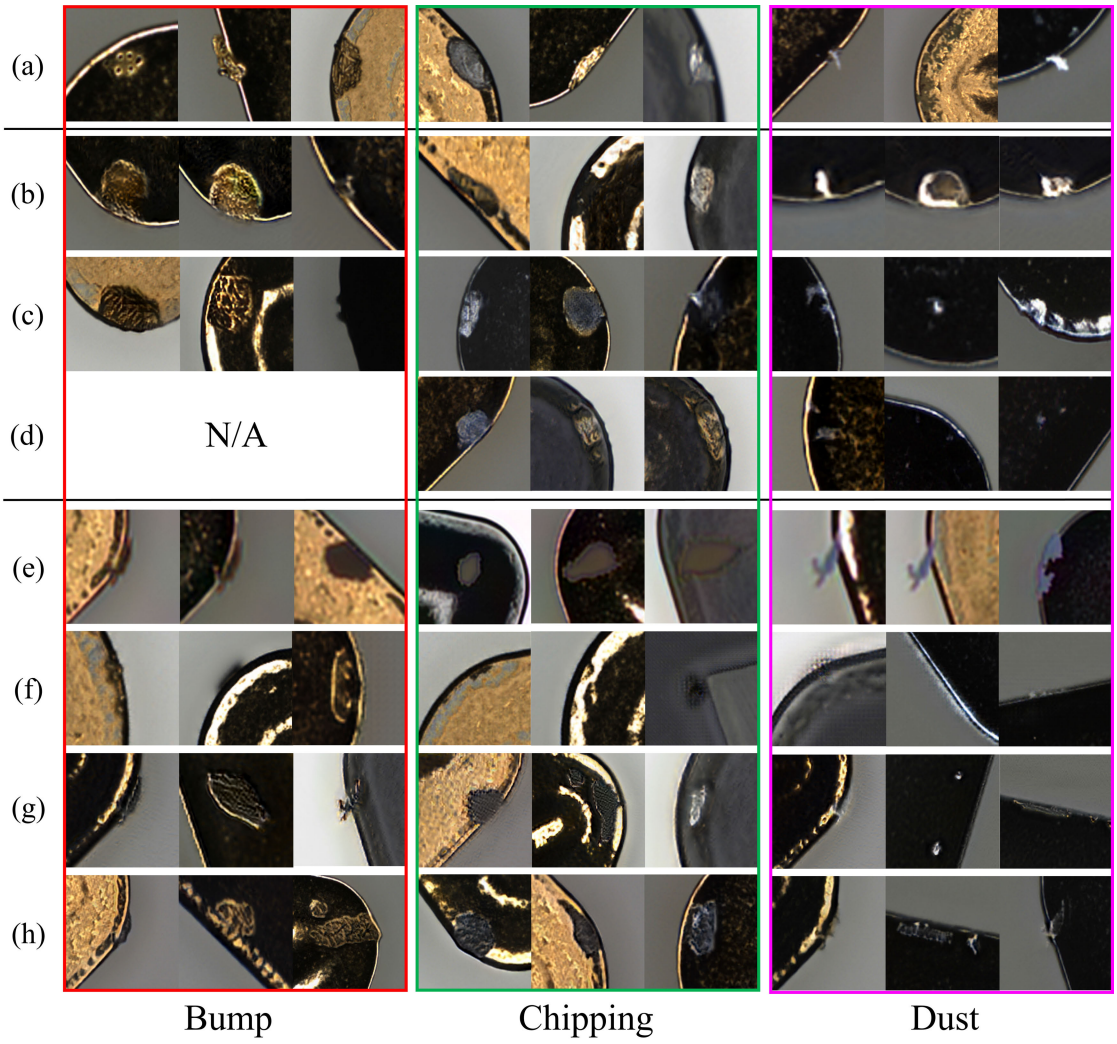


FIGURE 5. Comparison of qualitative results of different methods: (a) Real defect, (b) ADA (Generation), (c) Projected StyleGAN2, (d) Projected FastGAN, (e) SyNDGAN*, (f) CycleGAN, (g) ADA (Translation), and (h) SyNDGAN (Ours).

to the model in which a normal image and a mask including defect information (position and defect type) are combined, the corresponding defect image is output. Each defect class is displayed in a different color, and it can be confirmed that defects are generated in the area corresponding to the mask. The synthesized defect image is similar to the actual normal image except for the defect area, and through this, it is confirmed that a realistic defect image can be synthesized.

Figure 5 provides a qualitative comparison of defect synthesis results between the proposed model and other models. In this study, we compare the results of two approaches: generating defect images (b, c, and d) from random vectors within the training distribution through transfer learning, and translating normal images into defect images (e, f, g, and h). Generation-based methodologies usually produce images that closely resemble real-world defects. However, they often represent limitations in expressing the full diversity of defect representations. In the case of Projected FastGAN, it tends not to generate certain class of defects. This implies a limitation in that various defects cannot be generated when randomly generated from the distribution of training data. When defect synthesis is performed using SyNDGAN* that had not been pre-trained, it could be seen that the generated defects had unnatural textures and uniform colors.

Translation-based methodologies exhibit a greater diversity in defect types and representations compared to generation-based methodologies. However, the quality of synthesis itself is somewhat inferior in translation-based approaches. The defect image synthesized through our model is realistic and shows that various defects can be synthesized for each class. Table 1 shows the results of quantitative comparison of synthesized defect images. The proposed model yields the lowest FID [40] compared to the other models [22], [23], [28], [42] based on translation method. Although KID [41] yields a lower value in the ADA [23] model, it can be confirmed that the proposed technique performs better when both evaluation indicators are considered. Generation-based methodologies demonstrate a lower figure compared

TABLE 1. Quantitative comparison of defect synthesis. * indicates non-pretrained method.

Method	Model	FID↓	KID↓ ($\times 10^3$)
Generation	Projected StyleGAN2	27.57	4.02
	ADA	25.94	6.14
	Projected FastGAN	**24.98**	**1.59**
Translation	Projected FastGAN	250.26	259.00
	SyNDGAN*	76.89	18.00
	StyleMapGAN	61.31	19.00
	CycleGAN	51.11	18.00
	ADA	50.60	**10.00**
	SyNDGAN (Ours)	**41.95**	11.00

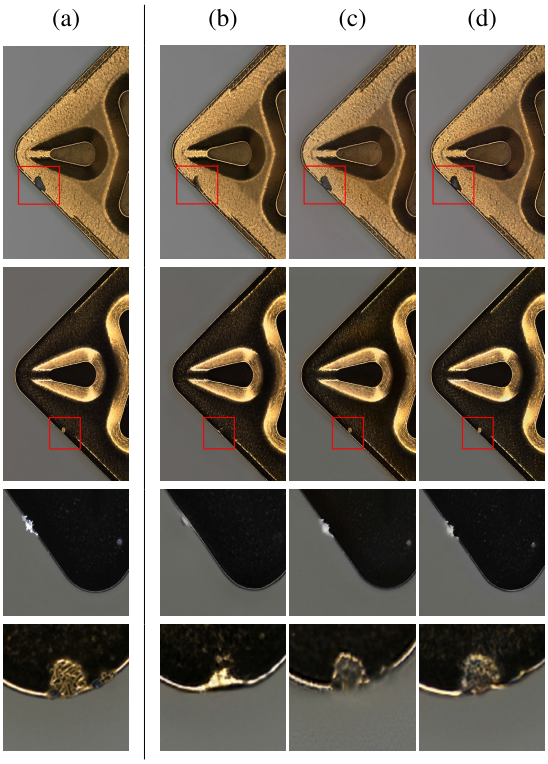


FIGURE 6. Comparison of reconstructed defect images from different models: (a) Real defect, (b) StyleMapGAN, (c) ADA, and (d) SyNDGAN (Ours).

to translation-based methodologies. However, as previously mentioned in Figure 5, when considering qualitative results, they exhibit limited diversity in defect generation relative to quality.

Figure 6 presents a qualitative comparison of the reconstructed images for a sample of defects for each model. For ADA, the defect representation produces images that are similar to the actual defect, but blurred. The proposed technique recovers the most realistic defect images among the compared models. Furthermore, Table 2 indicates the reconstruction performance of defect synthesis models using LPIPS. We evaluate whether realistic reconstruction is possible when normal and defect images are inputted along with a segmentation map devoid of defect information. Our method has the lowest value, which means that the difference between the actual image and the reconstructed image is small and the encoder and decoder performance is superior.

TABLE 2. LPIPS evaluation results - real and reconstructed images of different models.

Method	Normal ↓	Defect ↓
StyleMapGAN	0.1856	0.2197
ADA	0.3375	0.3594
SyNDGAN (Ours)	**0.1743**	**0.1813**

TABLE 3. Comparison of the amount of data before and after augmentation for each defect.

Defect class	*w/o* Augmentation	*w/* Augmentation (# of synthesized data)
Normal	270	400
Bump	261	401 (+140)
Chipping	308	483 (+175)
Dust	192	308 (+116)

Overall, our proposed model demonstrates superiority in both quantitative and qualitative results compared to other methods. Notably, our model excels in synthesizing fine details such as chipping and dust.

2) CLASSIFICATION PERFORMANCE

Table 3 displays the number of items for the normal and defect data for each class, both prior to and after the augmentation. The number of normal samples is computed by taking the average of the number of defective samples before and after the augmentation. Each defective sample is augmented with about 50% of the normal sample quantity. The dataset is divided into training and testing sets in the ratio 8:2.

Table 4 quantitatively presents the classification performance according to the augmentation method. The numbers for the augmented data are: 140 bumps, 175 chippings, and 116 dusts. The overall and defect-specific classification accuracies are shown together; EfficientNet, a ResNet-based methodology, is used as the classification model. This model shows excellent performance in image classification with fewer parameters than conventional models [29], [43], and in this experiment, we apply the pre-trained model to ImageNet [33]. Cutmix [44] is a technique that takes Mixup [45] and Cutout [46] a step further by cutting and gluing regions to fill-in parts of an image with patches from other images. It is generally a high-performance methodology; however, it does not generalize well when the size of the defect is small relative to the resolution, as is the case with our dataset. The application of Cutmix to the training data resulted in a slight increase in total accuracy.

It is observed that Projected FastGAN and ADA, which utilize augmented data through generation, result in

TABLE 4. EfficientNet classification performance with different augmentation techniques.

Augmentation Methods	Per class Accuracy (%)		Total Accuracy (%)
Original data	Normal	98.15	72.55
	Bump	48.08	
	Chipping	85.00	
	Dust	50.00	
w/ Cutmix	Normal	98.15	74.02
	Bump	61.54	
	Chipping	68.33	
	Dust	65.79	
w/ Projected FastGAN	Normal	**100.00**	81.37
	Bump	**80.77**	
	Chipping	75.00	
	Dust	65.79	
w/ ADA	Normal	**100.00**	82.84
	Bump	63.46	
	Chipping	86.67	
	Dust	**78.95**	
w/ Ours	Normal	**100.00**	**84.31**
	Bump	73.08	
	Chipping	**88.33**	
	Dust	71.05	

TABLE 5. Ablation study on the proposed method - LPIPS results for normal and defect reconstruction.

Method	Normal ↓	Defect ↓
w/o Pretrained	0.2516	0.2615
w/o In-domain loss	0.1805	0.2006
w/ Auxiliary classifier D	0.1730	0.1898
w/ Single scale D	**0.1724**	0.1895
SyNDGAN (Ours)	0.1743	**0.1813**

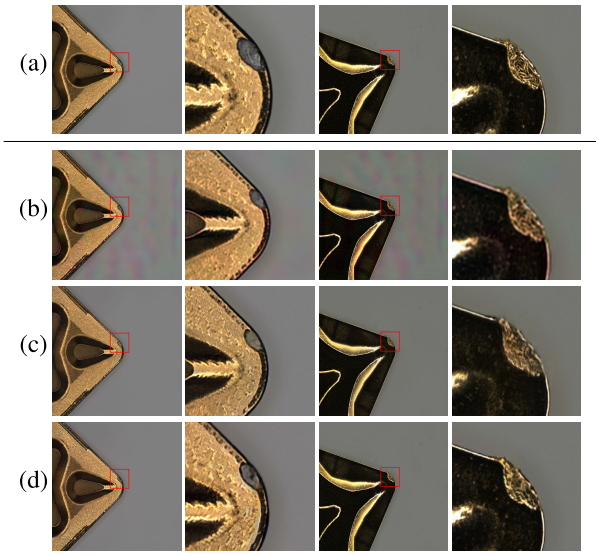


FIGURE 7. Comparison of reconstructed images of defects: (a) Real defect, (b) *w/o* Pretrained, (c) *w/o* In-domain loss, (d) SyNDGAN (Ours). For each restored defect image, a bounding box and an enlarged image of the defective part are shown.

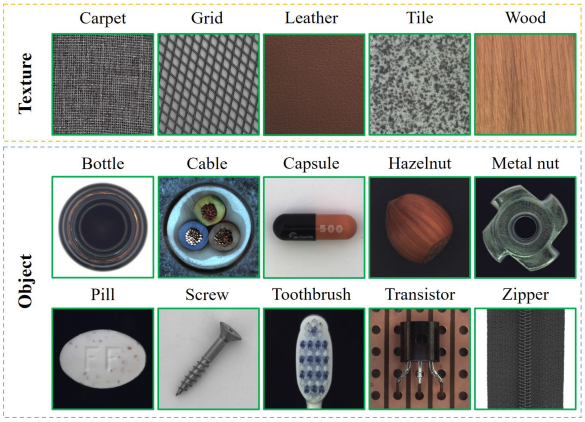


FIGURE 8. Composition of the MVTec AD dataset.

significantly improved accuracy compared to Cutmix. It is worth noting that while the generation-based models exhibit lower values in the evaluation metrics presented in Table 1, the proposed translation-based methodology demonstrates superior classification performance. The analysis suggests that the increased diversity of samples achieved through the proposed methodology enhances the generalization performance of the model.

B. ABLATION STUDY

We perform an ablation study to verify the effectiveness of our proposed method by measuring the difference between the real and reconstructed images. First, we conduct qualitative analysis using LPIPS metric as presented in Table 5. As shown in Table 5, without the pretraining stage, the performance significantly drops for both normal and defect images. Similarly, removing in-domain loss results in performance degradation to generate defect images since this loss function helps the encoder to map images to latent vectors effectively. Additionally, auxiliary classifier and single-scale training strategy for the discriminator make a small contribution to the enhancement of defect image generation. The proposed technique yields the best quantitative results for defect reconstruction. Considering that our model performs defect image synthesis, its efficacy can be readily noticed. Lastly, we conduct qualitative analysis as shown in Figure 7 based on reconstructed defect images. In terms of the non-pretrained model, we can observe that noticeable noise in the background and a lack of details of defects in reconstructed images. Additionally, the removal of the in-domain loss training strategy drops the performance of generating realistic defect images.

C. ADDITIONAL ANALYSIS

To evaluate the generalization performance of the proposed defect synthesis network, and to demonstrate its feasibility

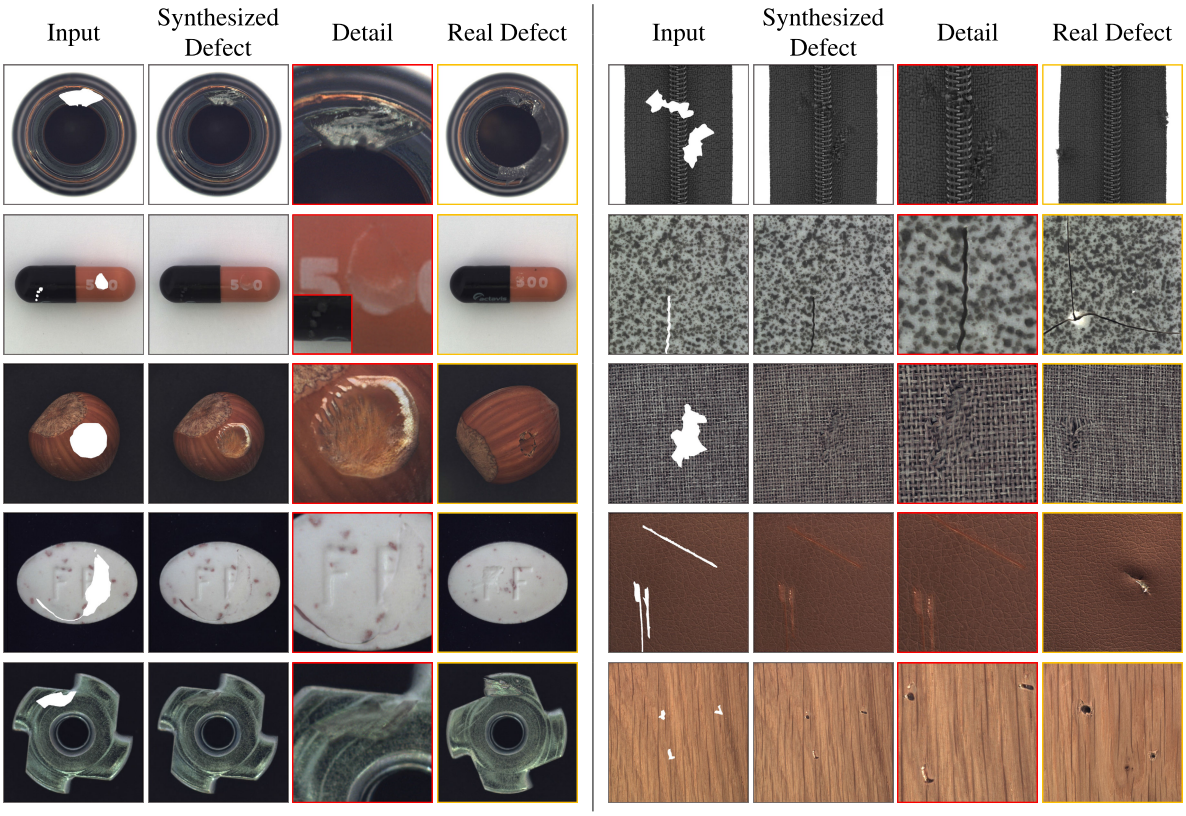


FIGURE 9. Qualitative results of defect synthesis using the proposed technique.

TABLE 6. LPIPS results of the proposed defect synthesis technique with and without pre-training on MVTec AD dataset.

Method	Normal ↓	Defect ↓
w/o Pretrained	0.2466	0.2512
w/ Pretrained	0.1771	0.1779

both qualitatively and quantitatively, we conduct additional experiments on other dataset. We use the MVTec AD [47] dataset as the training data, as it is widely adopted for defect detection [48] in the field of anomaly detection.

1) MVTEC AD DATASET

The MVTec AD dataset comprises real-world data, wherein the training set only includes normal images, while the testing set contains both normal and defective images along with segmentation map information as the ground truth for the corresponding defects. As shown in Figure 8, the dataset consists of 15 different classes, including 5 different textures and 10 different objects, with a total of 73 different types of defects. These defects encompass not only surface-level defects, but also structural abnormalities, such as missing or distorted parts. Anomaly detection [9] tasks differ from general classification in that the model is not trained on anomalous data. Therefore, we reconstruct the dataset by integrating both normal and defective data. Our proposed model is trained and then tested on 3,384 and 1,503 data samples, respectively. The segmentation maps are used as the ground truth for the defects.

2) QUANTITATIVE ANALYSIS

In quantitative analysis, we analyze the impact of pre-training strategy by measuring the perceptual distance between reconstructed images and their ground truth using LPIPS. Table 6 demonstrates the LPIPS results of the reconstruction images without pre-training for the proposed method. As presented in Table 6, the performance of pretrained model is significantly improved in both normal and defect scenarios compared to the from-scratch trained model. Interestingly, the gap becomes larger in defect scenarios compared to normal scenarios. These overall results highlight the critical importance of utilizing pretrained models to capture fine details and enhance performance.

3) QUALITATIVE ANALYSIS

We also conduct qualitative analysis as shown in Figure 9 and 10. In Figure 9, we present synthesized defect images from our model. The defect results are synthesized from the normal images and segmentation maps. It is noteworthy that the segmentation map used in this experiment does not include the defect class information and only provides the location information. Our results demonstrate that the

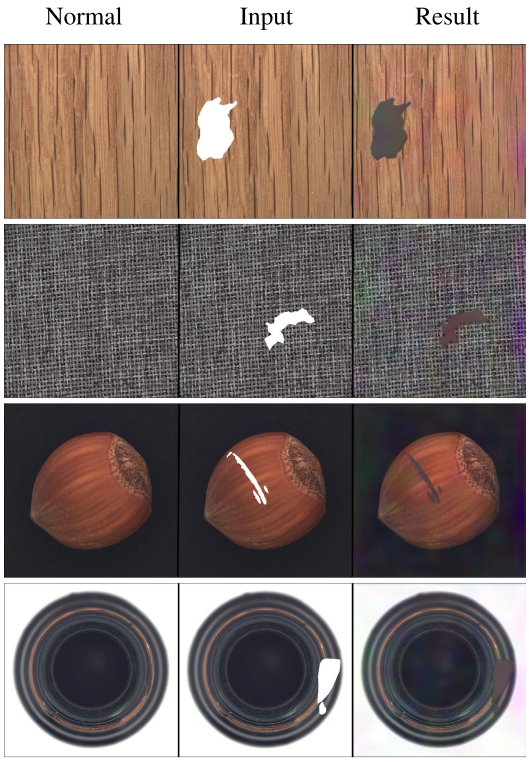


FIGURE 10. Defect synthesis results without pre-training.

synthesized defect images closely resemble the real defect images. Specifically, the synthesized defect images exhibit similar defects as those observed in the real defect images.

Furthermore, we present the defect synthesis results obtained from the model without pre-training, as shown in Figure 10. Overall, the unrealistic defect is shown with a simple color change in one region of the segmentation map. Also, critical noises are presented in the background of the synthesized results. This reveals the performance drop in the synthesis and reconstruction when no pre-training is imparted. From our experiments, our results reveal that conducting pre-training becomes more crucial in situations where data availability is limited.

VI. CONCLUSION

The data imbalance problem is prevalent in the manufacturing industry, leading to disparities between the normal and defect samples. This study introduced a novel GAN-based data augmentation framework to address this issue under limited data situation. Utilizing pretrained strategy and defect masks containing both defect types and spatial information, SyNDGAN synthesizes diverse defect data from normal samples. The applicability of our proposal is demonstrated across various real-world industrial datasets. Furthermore, our method's effectiveness is confirmed by its demonstrable improvement in the performance of defect classification models. In future work, we plan to enhance controllability and finer detail expression.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[2] S. Niu, B. Li, X. Wang, and H. Lin, "Defect image sample generation with GAN for improving defect recognition," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1611–1622, Jul. 2020.

[3] G. Zhang, K. Cui, T.-Y. Hung, and S. Lu, "Defect-GAN: High-fidelity defect synthesis for automated defect inspection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2524–2534.

[4] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Nov. 2002.

[5] N. U. Niaz, K. M. N. Shahariar, and M. J. A. Patwary, "Class imbalance problems in machine learning: A review of methods and future challenges," in *Proc. 2nd Int. Conf. Comput. Adv.*, Mar. 2022, pp. 485–490.

[6] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2001, pp. 973–978.

[7] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[9] G. Pang, C. Shen, L. Cao, and A. Van den Hengel, "Deep learning for anomaly detection: A review," *Acm Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Apr. 2021.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Nov. 2020.

[11] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019, *arXiv:1910.13461*.

[12] Y. Li, M. Min, D. Shen, D. Carlson, and L. Carin, "Video generation from text," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2018, vol. 32, no. 1, pp. 1–8.

[13] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," 2016, *arXiv:1611.02163*.

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[15] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[16] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2794–2802.

[17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, Aug. 2017, pp. 214–223.

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.

[19] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[21] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *Proc. Eur. Conf. Comput. Vis.*, Nov. 2020, pp. 592–608.

[22] H. Kim, Y. Choi, J. Kim, S. Yoo, and Y. Uh, "Exploiting spatial dimensions of latent in GAN for real-time image editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 852–861.

[23] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2020, pp. 12104–12114.

[24] H. Zhang, Z. Zhang, A. Odena, and H. Lee, "Consistency regularization for generative adversarial networks," 2019, *arXiv:1910.12027*.

[25] Z. Zhao, S. Singh, H. Lee, Z. Zhang, A. Odena, and H. Zhang, "Improved consistency regularization for GANs," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2021, pp. 11033–11041.

[26] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10551–10560.
[27] B. Liu, Y. Zhu, K. Song, and A. Elgammal, "Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis," in *Proc. Int. Conf. Learn. Represent.*, Oct. 2020, pp. 1–13.
[28] A. Sauer, K. Chitta, J. Müller, and A. Geiger, "Projected GANs converge faster," in *Proc. Adv. Neural Inf. Process. Syst.*, Nov. 2021, pp. 17480–17492.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[30] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 606–615.
[31] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2019, pp. 6105–6114.
[32] (Sep. 2020). *How to Identify Carbide Inserts*. [Online]. Available: https://rdbarrett.co.uk/blog/how-to-identify-carbide-inserts/
[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[34] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Process. Lett.*, vol. 27, pp. 1485–1489, 2020.
[35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.
[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[37] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2019, pp. 1–12.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[39] Y. Wang, C. Wu, L. Herranz, J. Van de Weijer, A. Gonzalez-Garcia, and B. Raducanu, "Transferring GANs: Generating images from limited data," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2018, pp. 218–234.
[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 6626–6637.
[41] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," 2018, *arXiv:1801.01401*.
[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.
[43] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2019, pp. 1–10.
[44] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6023–6032.
[45] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.
[46] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.
[47] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9592–9600.
[48] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.

EUNHEE CHO (Student Member, IEEE) received the B.S. degree in aerospace software engineering from Hanseo University, in 2019, and the M.S. degree in electrical and computer engineering from Inha University, in 2023. From May 2019 to December 2020, she was a Research Engineer with the Center for Aviation Industry, Incheon Technopark. Her research interests include computer vision, deep learning, and image synthesis.

BYEONGHWAN JEON (Member, IEEE) received the B.S. and M.S. degrees in control and instrumentation engineering and the Ph.D. degree in electrical and computer engineering from Seoul National University, in 1988, 1990, and 2004, respectively. From February 1990 to December 2018, he was with the Mechatronics Research and Development Center, Samsung Electronics. From January 2019 to December 2020, he was with the Department of Electrical and Computer Engineering, Seoul National University, as a Professor, specialized in the collaboration between university and industry. Since January 2021, he has been with the Artificial Intelligence Convergence Research Center, as a Research Professor. His research interests include metrology and inspection technologies in semiconductor or display device manufacturing, including computer vision, deep learning, and signal processing.

IN KYU PARK (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Seoul National University, in 1995, 1997, and 2001, respectively. From September 2001 to March 2004, he was a member of Technical Staff with the Samsung Advanced Institute of Technology. Since March 2004, he has been with the School of Information and Communication Engineering, Inha University, where he is currently a Full Professor. From January 2007 to February 2008, he was an Exchange Scholar with Mitsubishi Electric Research Laboratories. From September 2014 to August 2015, he was a Visiting Associate Professor with the MIT Media Laboratory. From July 2018 to June 2019, he was a Visiting Scholar with the Center for Visual Computing, University of California, San Diego. His research interests include the joint area of computer vision and graphics, including 3D shape reconstruction from multiple views, image-based rendering, computational photography, deep learning, and GPGPU for image processing and computer vision. He is a member of ACM.