Defect Transfer GAN: Diverse Defect Synthesis for Data Augmentation

Ruyu Wang[1,2]
ruyu.wang@de.bosch.com
Sabria Hoppe[1]
sabrina.hoppe@de.bosch.com
Eduardo Monari[1]
eduardo.monari@de.bosch.com
Marco F. Huber[2,3]
marco.huber@ieee.org

[1] Bosch Center for Artificial Intelligence
Renningen, Germany
[2] Institute of Industrial Manufacturing and Management (IFF),
University of Stuttgart
Stuttgart, Germany
[3] Fraunhofer IPA
Stuttgart, Germany

Abstract

Data-hunger and data-imbalance are two major pitfalls in many deep learning approaches. For example, on highly optimized production lines, defective samples are hardly acquired while non-defective samples come almost for free. The defects however often seem to resemble each other, e.g., scratches on different products may only differ in a few characteristics. In this work, we introduce a framework, Defect Transfer GAN (DT-GAN), which learns to represent defect types independent of and across various background products and yet can apply defect-specific styles to generate realistic defective images. An empirical study on the MVTec AD and two additional datasets showcase DT-GAN outperforms state-of-the-art image synthesis methods w.r.t. sample fidelity and diversity in defect generation. We further demonstrate benefits for a critical downstream task in manufacturing—defect classification. Results show that the augmented data from DT-GAN provides consistent gains even in the few samples regime and reduces the error rate up to 51% compared to both traditional and advanced data augmentation methods.

1 Introduction

Automated Visual Inspection (AVI) is vital for quality control in modern production lines. One of the main challenges in AVI is the acquisition of suitable training data. First, labeling is usually expensive and time-consuming. Second, only very few defective parts are produced, which leads to imbalanced datasets. Both unlabelled and imbalanced data are very challenging for neural network model training.

Generative Adversarial Networks (GANs) [11] have shown promising performance to synthesize images where real samples are lacking. However, they tend to overfit on small datasets [20]. In this paper, we therefore present Defect Transfer GAN (DT-GAN), which uses defective images across multiple products to collect more information about their shared characteristics, even for products with few defects. For example, a scratch-like defect on a wooden surface may share similar shapes with a scratch-like defect on a metal surface, but

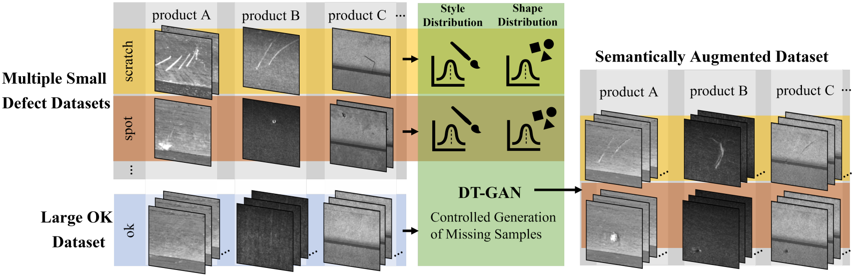


Figure 1: To enrich a dataset with few defective samples, DT-GAN synthesizes images under full control over background, defect shape, and style.

differ slightly in appearance according to their background materials. DT-GAN is based on two key features: (1) a weekly-supervised disentangling mechanism for the shared characteristics (foreground defect) and the unshared information (background product) of an input image. (2) An explicit modeling of the shape and style of foreground defects, where the styles of each defective type indicate their artistic looks such as light or heavy strokes. Since defect-specific distributions are learned, new images can be generated with style and shape sampled randomly or extracted from reference images. As a result, the proposed DT-GAN achieves semantically meaningful data augmentation by producing novel combinations of the foreground defects, their associated style and the background products, as illustrated in Fig. 1. Moreover, by jointly modeling the defect manifold from different products that have similar defect patterns, our design not only stabilizes the GAN training but also mitigates the overfitting issue on limited data.

Experiments on three industrial-oriented datasets showcase the power of DT-GAN in both defect synthesis and its usefulness in a downstream defect classification task where we report up to 51% reduction in error rates when augmenting the data with DT-GAN.

2 Related Work

Surface defect inspection aims at identifying and classifying defects with the help of machine vision. Traditional methods [29, 47] build models upon hand-crafted feature extractors, which are often outperformed by deep learning based models. However, the performance and generalization ability of deep learning approaches are restricted due to a limited number of defective samples in real-world scenarios.

Insufficient data has been addressed by multiple methods. Among them, data augmentation aims to enrich the training dataset by introducing invariances for the model to capture. Apart from the traditional augmentations [34, 36] such as random flipping and cropping, some more advanced regularization techniques [7, 44] like Cutmix [42] have been proposed. However, they do not introduce semantically new information to the training set.

In contrast, GANs augment data with meaningful semantic transformations. The power of GANs has been demonstrated in many computer vision tasks such as image synthesis [2, 8, 25], image to image translation [16, 17, 26, 28, 31, 49], style translation [4, 10, 18], image impainting [33, 39, 40, 41] and many other applications. Several recent works [30, 43] have proposed to use GANs for data augmentation with realistic defective samples. Defect-GAN [43] for instance tries to capture the stochastic variation within defects by mimicking the defacement and restoration processes. However, it still learns a deterministic mapping between inputs and outputs while our DT-GAN achieves multi-modality by varying styles. Moreover, DT-GAN incorporates the shared characteristics of defects from multiple products, which further enrich the diversity of synthetic defects for each product.

3 Methodology

Our approach is cast as an unpaired image-to-image translation problem, where we aim to achieve domain transfer between multiple domains within a single model. We define the *domain* as foreground defect types, where each type of defect is associated with a style distribution describing the artistic looks. The background product of an input should remain unaffected during the translation.

3.1 Proposed Framework

Our framework builds on StarGAN v2 [5], which transforms an image by a single vector representing the target style for the full image. However, to generate semantically meaningful defective images in our setting, it is essential for the model to understand and allow control over the components in an input image—the foreground defect pattern with its associated style and the background product. Given an image $\mathbf{x} \in \mathcal{X}$, its original defect domain $y \in \mathcal{Y}$ and its background product $p \in \mathcal{P}$, we modify and extend all four modules from [5] as follows (see Fig. 2 for the resulting model).

Style-Defect Separation. Our method models the shape and style separately by a domain-specific defect $\mathbf{c_y} \in \mathbb{R}^{H \times W \times C}$ and a style vector $\mathbf{s_y} \in \mathbb{R}^{1 \times 1 \times 512}$. The former learns to capture the shapes of the defects and the latter models their artistic looks. This feature allows our method to produce non-deterministic outputs by varying the style when the same target defect ($\mathbf{c_{\widetilde{y}}}$) is given. Thus, the mapping network M is trained to generate both defect patterns and their styles in all domains from a latent code $\mathbf{z}$. The final outputs $(\mathbf{c_{\widetilde{y}}}, \mathbf{s_{\widetilde{y}}}) = M_{\widetilde{y}}(\mathbf{z})$ are selected by the given target domain $\widetilde{y}$ among N output branches. The procedure for the style-defect encoder E is similar, except that the domain-specific defect $\mathbf{c_{\widetilde{y}}}$ and style $\mathbf{s_{\widetilde{y}}}$ are extracted from a given reference image. The two subnetworks are coupled by a consistency constraint (discussed in Section 3.2) between the joint image-style spaces, which prevents model degeneration and retains the multi-modality.

Foreground/Background (FG/BG) Disentanglement. It is crucial to identify and disentangle the FG and BG of an input image for achieving control over the defect (i.e., FG) and retraining the BG. The FG/BG disentanglement is performed by a depth-wise split at the bottleneck of G, which divides the feature map into two parts. Driven by the classification losses as discussed in Section 3.2, the model encodes the BG into the first channels and the domain-specific defect $\mathbf{c_y}$ into the latter channels. Instead of translating via a style vector, DT-GAN achieves domain transfer by altering the feature map—where $\mathbf{c_y}$ is replaced with defect $\mathbf{c_{\widetilde{y}}}$ from the target domain. The given defect style $\mathbf{s_{\widetilde{y}}}$ is applied through the adaptive instance normalization (AdaIN) [15]. However, to modulate only the fine details of the target defect $\mathbf{c_{\widetilde{y}}}$, the background BG_G is decoded separately without style modulation. Finally, BG_G and $\mathbf{c_{\widetilde{y}}}$ are concatenated together by depth-wise pooling before output. This design breaks the conditional relationship between FG and BG and therefore enables our method to freely combine them as well as learn the full variation of foreground defects.

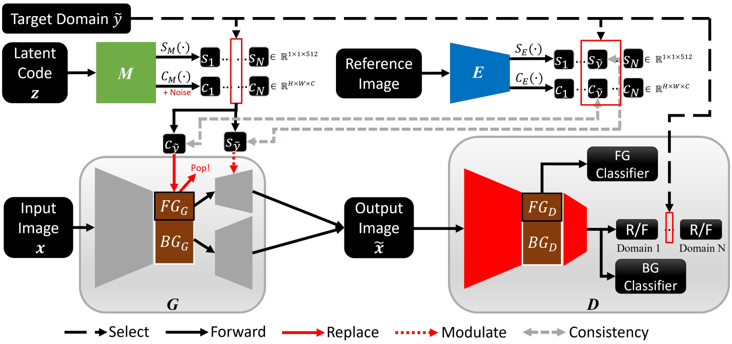


Figure 2: Overview of all modules in DT-GAN: the mapping network M, the style-defect encoder E, the generator G, and the discriminator D. Details of the modules are in Appendix D.

Multi-Task Discriminator with Auxiliary Classifiers. An FG defect classifier and a BG classifier are deployed in the multi-task discriminator to strengthen the disentanglement of FG and BG. Independent of the background, the FG defect classifier identifies the specific defect from an input image $\mathbf{x}$ in the latent space. The BG classifier acts on the image-level and decides whether the background information of the input image is well preserved. Apart from that, each branch $D_{\tilde{y}}$ in the multi-task discriminator D is trained to determine if an image $\mathbf{x}$ is a real image of its foreground defect domain or a fake image $\tilde{\mathbf{x}}$ generated by G.

Anchor Domain and Noise Injection. We impose an additional constraint on the generated and extracted foreground defect of a normal sample by setting the latent representation $\mathbf{c}_{\tilde{\mathbf{y}}}$ to zero [28]. We refer to this constraint as the 'anchor domain' and hypothesize that it supports the FG/BG disentanglement. Moreover, inspired by [19], a per-pixel noise injection is introduced to M to improve the diversity of the generated defects.

3.2 Training Objectives

Adversarial Loss. We follow the same adversarial loss as in [5] to encourage an output image $\tilde{\mathbf{x}} = G(\mathbf{x}, \mathbf{c}_{\tilde{\mathbf{y}}}, \mathbf{s}_{\tilde{\mathbf{y}}})$ to be indistinguishable from real images in the target domain $\tilde{y}$

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{\mathbf{x},y}\left[\log D_y(\mathbf{x})\right] + \mathbb{E}_{\mathbf{x},\tilde{y},\mathbf{z}}[\log\left(1 - D_{\tilde{y}}(\tilde{\mathbf{x}})\right)] \,, \tag{1}$$

where D_y and $D_{\tilde{y}}$ are the output branches of D that correspond to the source domain y and the target domain $\tilde{y}$, respectively.

Style-Defect Reconstruction Losses. To ensure G takes the domain-specific defect $\mathbf{c}_{\tilde{\mathbf{y}}}$ and the style $\mathbf{s}_{\tilde{\mathbf{y}}}$ into consideration during the generation process, we employ a style-defect reconstruction loss (cf. the gray dashed arrows in Fig. 2)

$$\mathcal{L}_{\text{sd\_rec}} = \mathbb{E}_{\mathbf{x},\tilde{y},\mathbf{z}}\left[\left\|\mathbf{c}_{\tilde{\mathbf{y}}} - C_E(\tilde{\mathbf{x}})\right\|_1\right] + \mathbb{E}_{\mathbf{x},\tilde{y},\mathbf{z}}\left[\left\|\mathbf{s}_{\tilde{\mathbf{y}}} - S_E(\tilde{\mathbf{x}})\right\|_1\right] \,, \tag{2}$$

where $C_E(\cdot)$ and $S_E(\cdot)$ indicate the extracted defect and style of an input, respectively. This objective urges E to recover $\mathbf{c}_{\tilde{\mathbf{y}}}$ and $\mathbf{s}_{\tilde{\mathbf{y}}}$ from $\tilde{\mathbf{x}}$. Besides, we apply another constraint to enforce that the detached domain-specific defect from G is consistent with the one retrieved from E

$$\mathcal{L}_{\text{d\_rec}} = \mathbb{E}_{\mathbf{x},y,\tilde{y},\mathbf{z}}\left[\left\|FG_G(\mathbf{x}) - \mathbf{c}_{\mathbf{y}}\right\|_1\right] + \mathbb{E}_{\mathbf{x},y,\tilde{y},\mathbf{z}}\left[\left\|FG_G(\tilde{\mathbf{x}}) - \mathbf{c}_{\tilde{\mathbf{y}}}\right\|_1\right] \,, \tag{3}$$

where $\mathbf{c}_{\mathbf{y}} = E_y(\mathbf{x})$, $\mathbf{c}_{\widetilde{\mathbf{y}}} = E_{\widetilde{y}}(\widetilde{\mathbf{x}})$; $FG_G(\mathbf{x})$ and $FG_G(\widetilde{\mathbf{x}})$ are the replaced defect from input image $\mathbf{x}$ and generated image $\widetilde{\mathbf{x}}$, respectively.

Diversity Loss. For a pair of random latent codes $\mathbf{z}_1$ and $\mathbf{z}_2$, we compute $\mathbf{c}_{\widetilde{\mathbf{y}}_i}, \mathbf{s}_{\widetilde{\mathbf{y}}_i} = M_{\widetilde{y}}(\mathbf{z}_i)$ for $i \in \{1,2\}$ and enforce a different outcome of G for differently mixed defect and style input pairs according to

$$\begin{aligned}\mathcal{L}_{\text{ds}} = &\mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}_1,\mathbf{z}_2}\left[\|G(\mathbf{x},\mathbf{c}_{\widetilde{\mathbf{y}}_1},\mathbf{s}_{\widetilde{\mathbf{y}}_2}) - G(\mathbf{x},\mathbf{c}_{\widetilde{\mathbf{y}}_2},\mathbf{s}_{\widetilde{\mathbf{y}}_1})\|_1\right] \\ &+\mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}_1,\mathbf{z}_2}\left[\|G(\mathbf{x},\mathbf{c}_{\widetilde{\mathbf{y}}_1},\mathbf{s}_{\widetilde{\mathbf{y}}_1}) - G(\mathbf{x},\mathbf{c}_{\widetilde{\mathbf{y}}_2},\mathbf{s}_{\widetilde{\mathbf{y}}_2})\|_1\right] \\ &+\Sigma_{m,n,o}\left[\mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}_1,\mathbf{z}_2}\left[\|G(\mathbf{x},\mathbf{c}_{\widetilde{\mathbf{y}}_m},\mathbf{s}_{\widetilde{\mathbf{y}}_n}) - G(\mathbf{x},\mathbf{c}_{\widetilde{\mathbf{y}}_o},\mathbf{s}_{\widetilde{\mathbf{y}}_o})\|_1\right]\right] ,\end{aligned} \tag{4}$$

where $m,n \in \{1,2|m \neq n\}$ and $o \in \{1,2\}$. Driven by this term, G is forced to discover meaningful defects and style features that lead to diversity in generated images. We ignore the denominator $\|\mathbf{z}_1 - \mathbf{z}_2\|_1$ of the original diversity loss [27] for stable training as in [5].

Cycle Consistency Loss. To encourage the disentanglement of the background, the domain-specific defect and the style, we impose a cycle consistency loss [48] to reconstruct the input image $\mathbf{x}$ with given defect $\mathbf{c}_{\mathbf{y}}$ and style $\mathbf{s}_{\mathbf{y}}$

$$\mathcal{L}_{\text{cyc}} = \mathbb{E}_{\mathbf{x},y,\widetilde{y},\mathbf{z}}\left[||\mathbf{x} - G(\widetilde{\mathbf{x}},\mathbf{c}_{\mathbf{y}},\mathbf{s}_{\mathbf{y}})||_1\right] , \tag{5}$$

where $\mathbf{c}_{\mathbf{y}}, \mathbf{s}_{\mathbf{y}} = E_y(\mathbf{x})$ is the defect and style of the input image $\mathbf{x}$, respectively.

Classification Losses. We employ two classification losses, which are essential to enforce the FG/BG disentanglement: First, the FG defect classification loss

$$\mathcal{L}_{\text{FG}} = \mathbb{E}_{\mathbf{x}_{\text{real}},y}\left[-\log D_{\text{FG}}(y|\mathbf{x}_{\text{real}})\right] + \mathbb{E}_{\mathbf{x}_{\text{fake}},\widetilde{y}}\left[-\log D_{\text{FG}}(\widetilde{y}|\mathbf{x}_{\text{fake}})\right] , \tag{6}$$

which aims to ensure the domain-specific defect is properly encoded and carries enough information from the target domain. Second, the BG classification loss

$$\mathcal{L}_{\text{BG}} = \mathbb{E}_{\mathbf{x}_{\text{real}},p}\left[-\log D_{\text{BG}}(p|\mathbf{x}_{\text{real}})\right] + \mathbb{E}_{\mathbf{x}_{\text{fake}},p}\left[-\log D_{\text{BG}}(p|\mathbf{x}_{\text{fake}})\right] , \tag{7}$$

where p is the corresponding background type of $\mathbf{x}_{\text{real}}$ and $\mathbf{x}_{\text{fake}}$. With the help of this objective, G learns to preserve the unshared characteristics of its input image $\mathbf{x}$ while dissociating the foreground defect.

We summarize the full objective and provide the training details in Appendix B.

4 Experiments

Dataset. We conducted the image synthesis experiments on three industrial-oriented datasets: the MVTec AD, the Magnetic Tile Defects (MTD), and a new dataset of industrial images—the Surface Defect Inspection (SDI)[1]. For all the experiments, we re-organized the defects in the datasets into three mutually exclusive classes: `Normal`, `Scratches`-like and `Spots`-like according to their visual appearance.

All three datasets are relatively small—the number of defective images for each defect category varies from 8 to 620, which is rather limited considering the sophisticated patterns of defects. This poses a major challenge for training generative models. Details of the datasets are summarized in Appendix A.1.

[1] The SDI dataset will be published with the final version of the paper.

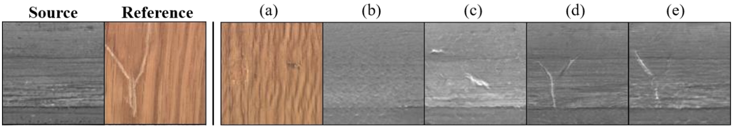


Figure 3: Ablation study. (a) The baseline StarGAN v2[5]. (b) + Style-defect branches. (c) + FG and BG classifier. (d) + Separately decoding FG and BG in G. (e) + Anchor domain (e.g. `Normal`) and Noise injection in M. (Best viewed in color.)

To study the performance of DT-GAN generated samples in defect classification, all the experiments were performed on the SDI dataset due to the limited availability of defective samples in the other two datasets. Note that only the training set of the SDI datset was used in GAN training, the validation and test set were left untouched for final evaluation in classifier training. For a fair comparison, all images were resized to 128 × 128 resolution for both GAN training and classifier training, which was also the highest resolution used in the baselines for image generation.

4.1 Defect Generation

Baselines. As discussed in Section 3, DT-GAN can either use M to randomly generate defects and styles, or use E to extract both from one or two reference images. We refer to these cases as 'latent-guided' and 'reference-guided', respectively. Since the two ways of guidance are fundamentally different, we evaluated them against two sets of baselines: our reference-guided image generation was compared to Mokady et al. [28] and StarGAN v2 [5]. Note that without the key designs we introduced in Section 3.1, DT-GAN degrades to [5]. Images generated through the latent-guided part of DT-GAN were compared to the state-of-the-art GANs in image synthesis: BigGAN [2] and StyleGAN2 [21]. We set both [2] and [21][2] to condition on defect types during training. All baselines were trained from scratch with the public implementations provided by the authors[3].

Metrics. We employed the commonly used frechet inception distance (FID) [14] to evaluate both the visual quality and the diversity of the generated images. A lower FID score indicates better performance.

Ablation Study. We visually demonstrate the effect of each feature we added to DT-GAN compared to [5] in Fig. 3, using the examples of reference-guided image synthesis from `Normal` to `Scratches`. Also, we report the average FID over all three datasets for each configuration in Appendix E.1.

Fig. 3(a) corresponds to [5] and highlights the drawback of an entangled style vector—the model extracts a style from the entire reference image instead of a style of the foreground defect and thus, changes the background product in its output, which we refer to as an 'identity-shift'. We first tackle this problem by modeling the foreground defect and style explicitly and introducing the FG/BG disentanglement, so the defect replacement and style modulation can be performed at different stages in G. This leads to better preservation of the background structure in (b), but the resulting image contains no clear defect from the reference image. Thus, we add a FG and a BG classifier to D in (c) to ensure the output image contains the desired foreground defect. Note that the additional product type labels can be

[2] We used the implementation in [20] for conditional training.

[3] We could not obtain the code of Defect-GAN [43] to reproduce their results.

Table 1: Quantitative comparison of DT-GAN with baseline image synthesis methods using FID(↓). Note that * indicates that the model is trained with augmentation methods.

Method	A	B	C	CARPET	LEATHER	TILE	WOOD	MTD	All
Mokady et al. [28]	68.69	66.9	**36.21**	41.87	60.26	275.12	81.71	68.30	**87.38**
StarGAN v2 [5]	96.85	58.28	50.95	354.31	336.63	434.77	411.37	84.49	228.46
StyleGAN2 [21]	90.1	**52.95**	138.09	51.37	**51.6**	**225.96**	140.01	**51.39**	100.18
BigGAN* [2]	218.74	134.41	270.89	34.47	101.7	391.54	113.32	67.91	166.62
Ours	**65.62**	53.62	37.94	**27.33**	78.01	352.15	**77.11**	78.41	96.27

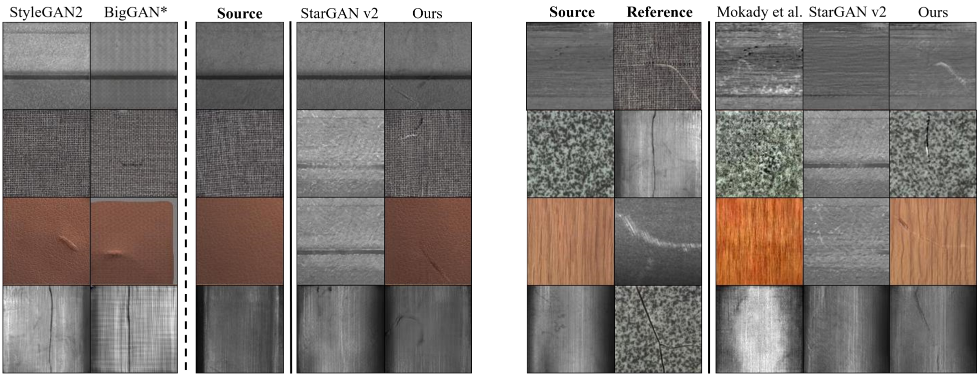
acquired automatically from production lines. These two auxiliary classifiers improve the image quality by a big margin, however, the generated defects fail to preserve the structure shown in the given reference. To address this issue, we add separate decoders for FG and BG in G. As seen in Fig. 3 (d), this enhances the preservation of background characteristics like lighting even more and the foreground defect characteristics start to match the patterns from the reference. Finally, we impose the anchor domain constraint and the per-pixel noise injection to M. This leads to more diverse defects which are not clear in Fig. 3 (e) but clearly affect the FID scores.

Quantitative and Qualitative Evaluation. The quantitative comparison of DT-GAN with baseline image synthesis methods on all datasets is shown in Table 1, and the qualitative comparison is in Fig. 4. For a fair comparison, we trained [2], [21] and [28] on each product separately to have control on background products. We also experimented with augmentation methods for GAN training [20, 46] and only report the best setting (see Appendix E.4). Note however, the images from [5] and DT-GAN were always obtained from a single model.

As observed in Table 1, our method outperforms the rest in 3 out of 8 cases and provides the second best overall performance. Our method is often outperformed by [21] and [28], however, we note that FID is not sensitive to detect overfitting, which often happens when training on a small dataset. We thus present the nearest neighbor results in Appendix E.5 to demonstrate that the low FID scores of [21] and [28] come from memorizing the training dataset. Also, there are further evaluations in the downstream task which support our assumption (see Appendix C.2).

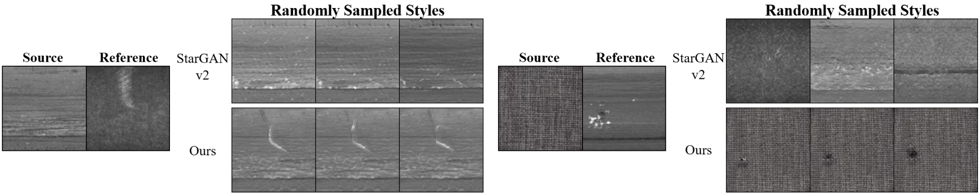
The latent-guided image synthesis results are presented in Fig. 4 (a). We observe that generated samples from [2] often present abnormal grid patterns and samples from [21] either overfit or contain no clear defect. Both methods do not take images as inputs but infer both FG and BG of a synthetic image from a given latent code. This conditioning leads to limited diversity and artifacts in the output images while making the models less robust to overfitting. In contrast, [5] performs translation based on input images but suffers from the same entanglement issue. As discussed in the ablation study, due to the lack of our designed features, [5] fails to preserve the product type in its output images (i.e., identity-shift) and results in undesired outputs, which is also reflected in the FID scores. Our architecture, which disentangles FG and BG, mitigates these issues and provides visually convincing results.

Also for reference-guided image synthesis, where we used defects from different foreground reference images as illustrated in Fig. 4 (b), only our method produces high-quality images with preserved background from the source and transferred foreground defect from the reference. This again showcases the effectiveness of the FG/BG disentanglement. See Appendix E.2 for more images, where DT-GAN is the only method that can perform translations between all domains, including defect-to-defect translations.



(a) Latent-guided (b) Reference-guided

Figure 4: Qualitative comparison of latent-guided and reference-guided image synthesis results on case `Normal`-to-`Scratches`. In each subfigure, the **Source** column indicates the expected background in the output images. (a) The defective images of the first two columns are fully generated from random noise, while random defects are synthesized onto given source images in the last two columns. * indicates that the model was trained with DiffAug [46]. (b) Each method transforms the given source images into the target defect domain with the defects and styles extracted from the reference images. (Best viewed in color and zoom in.)



(a) `Normal`-to-`Scratches` (b) `Normal`-to-`Spots`

Figure 5: The visual effect of randomly sampled styles on StarGAN v2 and DT-GAN when given a fixed pair of source background and reference defect images.

Styling. We visually demonstrate the effect of the style vector in Fig. 5. When combing randomly sampled styles with a fixed pair of input images, [5] suffers from the identity-shift and fails to produce meaningful defects while DT-GAN provides a variety of artistic styles in its outputs due to the style-defect separation as discussed in Section 3.1.

4.2 DT-GAN for Data Augmentation

To demonstrate the effectiveness of our synthetic images, we also evaluated DT-GAN as a data augmentation method for defect classification as an exemplary downstream task. Therefore, we used all defective samples from the SDI dataset and an additional 4,000 normal (non-defective) images for each product to generate further defective samples for classifier training (see Appendix C for more details).

As backbone we employed the widely used ResNet-50 [12] with ImageNet pretrained weights. For experiments with synthetic data, we attached an auxiliary classifier to the network through a Gradient Reversal Layer (GRL) [9], ensuring the extracted features by the backbone are invariant for both the real and the synthetic samples. Since the SDI dataset is

Table 2: Quantitative comparison of the baseline methods on the defect classification task. The reported values are the achieved error rates (%) and standard deviation over five runs.

Aug. Method	**Syn. Data**	**ResNet-50**		
		A	B	C
None	None	14.91±1.52	8.2±1.49	15.24±1.51
Trad	None	13.81±2.36	6.8±1.64	16.57±3.20
Trad	Mokady et al. [28]	20.72±1.49	5.8±2.77	24.76±10.1
Trad	StarGAN v2 [6]	10.60±1.99	7.4±3.44	15.81±1.44
Trad	StyleGAN2 [21]	29.45±9.13	6.8±2.05	13.14±3.12
Trad	BigGAN* [2]	12.17±1.99	5.8±1.93	15.62±3.06
Trad	Ours	**6.72±1.65**	**4.6±0.89**	**12.76±1.97**
CutMix [42]	None	13.63±2.87	7.4±1.52	14.09±2.27
CutOut [7]	None	12.36±0.50	6.2±0.84	12.95±2.19
MixUp [44]	None	14.36±1.75	6.2±1.79	16.38±2.80
CutMix [42]	Ours	14.54±3.02	5.2±0.45	19.42±3.47
CutOut [7]	Ours	**12.18±1.99**	**4.0±1.22**	**11.42±1.50**
MixUp [44]	Ours	15.27±2.98	8.2±3.49	21.52±3.96

Table 3: Quantitative comparison of the image synthesis methods using LPIPS to measure the similarity between the synthetic samples. The lower score indicates more similarity.

Method	A	B	C	CARPET	LEATHER	TILE	WOOD	MTD
Mokady et al. [28]	0.34	0.46	0.22	0.14	0.28	0.22	0.22	0.38
StarGAN v2 [6]	0.32	0.33	0.20	0.37	0.38	0.40	0.38	0.38
StyleGAN2 [21]	0.29	0.36	0.19	0.09	0.26	0.27	0.18	0.36
BigGAN* [2]	0.30	0.29	0.19	0.08	0.22	0.21	0.18	0.37
Ours	**0.28**	**0.28**	**0.17**	**0.07**	**0.18**	**0.19**	**0.17**	**0.30**

highly imbalanced, we oversampled the minority classes [23] unless the data was balanced through synthetic images. Traditional data augmentations like random horizontal flips, jittering, and lighting [36] were always applied unless otherwise specified. All following results were evaluated by the achieved error rates over five runs with different random seeds.

Effectiveness of Synthetic Data. We first compare the classifier performance for no augmentation, traditional data augmentation (Trad-Aug), advanced regularization techniques like [42], [7] and [44] and a combination of traditional augmentation with synthetic images for GAN methods including DT-GAN in Table 2. For brevity, the detailed results are in Appendix E.3. We found consistent improvements when combining our method with [7]. In contrast, [42] and [44] seemed to jeopardize the performance. We hypothesize that it is due to the real-fake domain gap—both methods regularize the training by randomly concatenating two training images in different manners, however, it destructs the backpropagation from the GRL, which results in poor performance.

Methods like [28] and [21] outperformed our method in some cases concerning FID. However, our method led to better performance in downstream classifier training. We hypothesize this is because the baselines overfit the training set. Quantitatively this is supported by the LPIPS [45] scores in Table 3, which computes the similarity of the synthetic samples to each other. Qualitatively this could be seen in the nearest neighbor analysis in Appendix E.5. None of the commonly used metrics is designed to indicate small perturbations like the variance of defects. However, by combining FID, LPIPS, the nearest neighbor

Table 4: Classifier performance using synthetic images generated by DT-GAN trained on reduced (20A) and full-scale (All) of the SDI dataset.

Dataset Size	20A		All	
	Trad-Aug	Ours	Trad-Aug	Ours
A	34.18±4.39	**28.55±7.32**	13.81±2.36	**6.72±1.65**
B	5.8±0.45	**5.6±1.14**	6.8±1.64	**4.6±0.89**
C	16.95±1.17	**10.86±1.28**	16.57±3.20	**12.76±1.97**

Table 5: Cross-product defect transfer on classifiers trained with reference-guided synthetic images of DT-GAN using different products as defect reference. For 'v-Others', we only report the best results from all experiments with other products as reference.

	Trad-Aug	v-Same	v-Others	v-ABC
A	13.81±2.36	11.81±2.65	11.99±1.63	**11.09±3.49**
B	6.8±1.64	6.6±1.52	6.4±1.34	**5.6±1.34**
C	16.57±3.20	14.85±1.73	**11.23±0.80**	11.42±0.96

analysis, and the classifier performance, we believe that our method improves performance on all products due to the combination of high visual quality and diversity in our samples.

Impact of Dataset Size. Motivated by the limited availability of data in real-world production scenarios, we evaluated DT-GAN for data augmentation on the full SDI dataset (All) as well as a subset, which only contains 20 defective samples of product A for each defect type (20A). In this case, DT-GAN was also trained on the reduced subset. As shown in Table 4, there is a clear improvement when synthetic images from DT-GAN are used as data augmentation, even for the extremely limited data subset.

Cross-Product Defect Transfer. We hypothesized that limited data can be counteracted by transferring defects across multiple background products if there are at least some defects that occur on multiple products. We tested this approach by comparing the performance of classifiers trained on synthetic images with defects from the same product (v-Same), from another product (v-Others) and from all products (v-ABC). As we can see in Table 5, the best performances are achieved by the models that transfer defects across products (v-Others or v-ABC). We interpret this as support for our hypothesis and its practical usefulness. The full scale results are in Appendix E.3.

5 Conclusion

We propose a novel method, DT-GAN, which allows diverse defect synthesis and semantic data augmentation by exploiting shared defect characteristics across multiple products. Due to explicit style-defect separation and FG/BG disentanglement, DT-GAN achieves higher image fidelity, better variance in defects, and full control over FG and BG while being sample-efficient and robust against model overfitting. We demonstrated the feasibility and benefits of DT-GAN on a real industrial defect classification task and the results show that our method provides consistent gains even with limited data and boosts the performance of classifiers up to 51% compared to traditional augmentation and state-of-the-art image synthesis methods. For future investigation, we aim to represent defects and their styles more explicitly (e.g., localization), improve the explainability of the model and also enhance the model transferability to unseen products.

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD — a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. doi: 10.1109/CVPR.2019.00982.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[3] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. pages 3339–3348, 2018.

[4] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[7] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[8] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. doi: 10.1109/CVPR.2016.265.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[12] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[15] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[16] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *Proceedings of the European Conference on Computer Vision (ECCV)*, abs/1804.04732, 2018.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[20] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:12104–12114, 2020.

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[22] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

[23] Charles X. Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 73–79. AAAI Press, 1998.

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[25] Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4183–4192.

[26] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[27] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1429–1437, 2019.

[28] Ron Mokady, Sagie Benaim, Lior Wolf, and Amit Bermano. Masked based unsupervised content transfer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[29] Henry Y. T. Ngan, Grantham K. H. Pang, and Nelson H. C. Yung. Review article: Automated fabric defect detection-a review. *Image and Vision Computing*, 29(7):442–458, June 2011. ISSN 0262-8856. doi: 10.1016/j.imavis.2011.02.002.

[30] 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*, 17(3):1611–1622, 2020. doi: 10.1109/TASE.2020.2967415.

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zach DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[33] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[34] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *ArXiv*, abs/1712.04621, 2017.

[35] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[36] Connor Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019.

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

[38] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[39] Raymond A. Yeh, Chen Chen, Teck-Yian Lim, Alexander G. Schwing, Mark A. Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with deep generative models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6882–6890, 2017.

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5505–5514, 2018. doi: 10.1109/CVPR.2018.00577.

[41] Jiahui Yu, Zhe L. Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4470–4479, 2019.

[42] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Young Joon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6022–6031, 2019.

[43] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. Defect-gan: High-fidelity defect synthesis for automated defect inspection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2524–2534, January 2021.

[44] Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, volume abs/1710.09412, 2018.

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:7559–7570, 2020.

[47] 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, January 2014. ISSN 0925-2312. doi: 10.1016/j.neucom.2013.07.038.

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[49] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.