DefectGen: Few-Shot Defect Image Generation Using Stable Diffusion for Steel Surface Analysis

Adnan Md Tayeb, Hope Leticia Nakayiza , Heejae Shin, Seungmin Lee,
Chaesoo Lee*, YeongHun Lee*, Dong-Seong Kim, Jae-Min Lee
Networked Systems Laboratory, Department of IT Convergence Engineering,
Kumoh National Institute of Technology, Gumi, South Korea.
(mdtayebadnan, hopeleticia, shinheejae, tmdals6164, dskim, ljmpaul)@kumoh.ac.kr
*4IND Co., Ltd.
*(ceo, yhoon)@4ind.co.kr,

***Abstract*—Detecting defects on steel surfaces is crucial for maintaining production standards and minimizing material waste. However, the scarcity of defect images poses a significant challenge in developing robust detection models, which hampers effective defect inspection. To address this issue, we propose DefectGen, a novel data augmentation approach that leverages Stable Diffusion and blending techniques to enhance defect datasets and improve detection performance. With only a limited number of defect images, our method aims to generate a larger and more realistic defect image dataset. The defect image generation process is structured into two stages: the Defect-Free Stage and the Defect Generation Stage. In the Defect-Free Stage, Stable Diffusion is trained on a defect-free dataset to generate realistic steel surface images. The Defect Generation Stage is divided into two phases: defect extraction and modification, followed by blending. In the first phase, defects are extracted from a small number of defect images and modified according to specified conditions. The second phase involves deep blending of the modified defects into the generated defect-free images based on the provided locations, resulting in realistic defect images.Experiments on our steel surface dataset demonstrate the effectiveness of DefectGen, leading to a 11.91% improvement in Mean Average Precision (mAP) for defect detection.**

***Index Terms*—Generative AI, Stable Diffusion, Few-Shot Learning, Defect generation, Defect detection**

I. Introduction

The fast growth of artificial intelligence in recent years has led to many creative advancements in different sectors, with a notable impact on real-world industrial uses. Due to its convenient framework, supervised learning is widely used in creating vision models for tasks like object detection [5], [11], [23] and segmentation [22] in industrial settings. Even though the supervised method is effective, it encounters a major obstacle: obtaining large quantities of accurately annotated data can be difficult and require significant resources [24]. For instance, gathering datasets with defective images presents a challenge since engineers focus on refining processes to reduce the occurrence of defects. Additionally, generating high-quality annotations is both expensive and often requires specific skills. It is essential to tackle and minimize the reliance on large amounts of data and annotations needed to train supervised models in order to enhance the incorporation of AI in industrial settings and improve its accessibility.

To address the challenge of limited data, researchers have explored various strategies. For instance, DeVries and Taylor (2017) attempted to artificially create defective images by manually introducing artifacts. Similarly, Li et al. (2021) employed techniques such as cutting and pasting patches from defect-free images or transferring defect regions from one image to another. However, these methods often resulted in images that lacked realism and diversity. The advent of generative AI represents a transformative advancement in this field, offering the ability to produce highly realistic and varied defective images. In recent years, text-to-image generative models have displayed impressive progress, demonstrating their capacity to create high-quality images based on text descriptions. These models have been trained on large image datasets, enabling them to produce samples with significant fidelity and diversity. An example worth mentioning is Stable Diffusion, a publicly available model that has enabled a variety of powerful uses because of its effectiveness and flexibility. To further enhance image generation, researchers have explored various techniques to integrate key elements from reference images into diffusion models, thereby improving content precision when working with a limited set of images [1], [2], [4], [6], [20]. Chen et al. [1] pioneered a complete parameter adaptation approach, which involves modifying the entire diffusion model to better align with the reference images. Han et al. [6] proposed a method that uses SVG decomposition with a small set of trainable parameters to prevent catastrophic forgetting in scenarios with few reference images. This approach helps the model align more accurately with reference images while reducing the risk of overfitting. Additionally, Chen and colleagues [2] developed an image-conditioned adapter that retains essential characteristics from the reference images without requiring optimization of the network parameters.

However, these approaches still require a substantial number of images to effectively train the model for generating new images [13], [18], [25], [27]. In industrial settings, for instance, developing a large dataset of defect images is both challenging and costly. To address this issue, some studies [3], [10], [12], [15] have explored zero-shot and few-shot techniques for defect generation, but these methods still fall short of

addressing the data insufficiency problem in real-world scenarios. Wang et al. (2020), Zhao, Cong, and Carin (2020), and Robb et al. (2020) have investigated few-shot image generation by leveraging pretrained models that can adapt from large domains to smaller ones. However, these approaches primarily focus on transferring entire images rather than emphasizing critical defect regions. Addressing the unique distribution of defects and defect-free areas individually could enhance defect generation methods.

In line with this, our study introduces, DefectGen, a few-shot defect image generation approach that creates new defect images using only a small number of existing defect images. In DefectGen, the process of generating new defect images involves two main stages: training on defect-free images and generating defects. First, a Stable Diffusion model is trained to generate a diverse set of high-quality, defect-free images. This model serves as the backbone for producing intermediary images that maintain visual fidelity and detail. In the second stage, defects are extracted from a limited set of defect images using binary masks, isolating only the defective regions. These defects are then modified—adjusting their size, shape, and position—to introduce variability. Finally, these modified defects are seamlessly blended into the defect-free images using an advanced blending technique, resulting in realistic, visually coherent defect images. This approach simplifies and strengthens defect analysis by improving the availability of diverse and realistic defect images with minimal data. We tested DefectGen on our Steel Surface defect dataset, and the results demonstrated its effectiveness. The augmented dataset generated by DefectGen achieved an 11.9% higher accuracy in terms of Mean Average Precision (mAP) for steel surface defect detection, highlighting the significant improvement in detection performance enabled by our approach.

To summarize, our contributions are as follows:

- We propose a novel model, DefectGen, capable of generating annotated defect images, which can be utilized for robust data augmentation.
- We successfully apply few-shot defect image generation using a limited number of defect images, tailored for industrial settings.
- We demonstrate the advantages of the proposed method in industrial applications by conducting experiments on a steel surface dataset. The results highlight the improved performance in defect detection.

II. Preliminary

A. Diffusion Models

Diffusion models (DDMs) [8], [9], [16], [17], [21], [26] are probabilistic models designed to generate data by progressively denoising normally distributed variables. Training these models involves a forward diffusion process that adds Gaussian noise to an image x_0 over a series of steps, forming a Markov Chain of length T. The objective is to train the model to reverse this diffusion process, reconstructing clean samples from pure noise through the reverse diffusion process.

Given a sample x_0 drawn from the real data distribution $q(x)$, the forward diffusion process incrementally adds Gaussian noise to x_0 at each step, governed by a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$. This can be formalized as:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \tag{1}$$

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \tag{2}$$

For sufficiently large T, x_T approximates an isotropic Gaussian distribution.

In the reverse diffusion process, the model aims to undo the noising steps, mapping noisy samples back to clean images. This is formalized as:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \tag{3}$$

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{4}$$

For simplicity, practitioners often assume $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$.

The model's output is frequently structured as a residual, predicting the noise ϵ_t added at each step rather than directly generating the image. The reconstructed image x_0 is obtained by subtracting ϵ_t from x_t. Thus, the training objective is:

$$L_t = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2\right] \tag{5}$$

where L_t represents the loss function at time step t.

B. Stable Diffusion

Latent Diffusion Models (LDMs) [19] are a widely used variant of diffusion models (DDMs) that operate in a latent space rather than directly in pixel space, thus reducing both training duration and inference costs. In an LDM, an encoder $E(\cdot)$ compresses an input image into a lower-dimensional latent representation z, where the diffusion and denoising operations are carried out. The decoder $D(\cdot)$ then reconstructs the image from this latent representation, producing $\tilde{x} = D(z)$.

A prominent example of an LDM is Stable Diffusion, which employs cross-attention layers to accommodate various conditioning inputs, such as text. Training Stable Diffusion involves two principal regularization techniques: i) KL-reg, which aligns the learned latent representation with a Standard Normal distribution, and ii) VQ-reg, which incorporates a vector quantization layer within the decoder (akin to VQGAN, but with the quantization layer embedded in the decoder). Stable Diffusion, trained on a comprehensive dataset of natural images, has achieved outstanding results across a range of tasks. Its pre-trained weights are publicly available, facilitating its use as a foundational model for diverse downstream applications.

III. METHOD

Given input defect free image $x_i \in \mathbb{R}^{W\times H\times C}$, a few number of defect images $x_d \in \mathbb{R}^{W\times H\times C}$ along with a binary mask $x_m \in \{0,1\}^{W\times H}$ where a value of 1 identifies the defective regions. The main goal is to generate new defect images $\hat{y} \in \mathbb{R}^{W\times H\times C}$. In this context, we refer to the input x_i as the defect free images, x_d as the defective image, x_g is the intermediate generated image by Stable diffusion and $\hat{y}$ as the generated image. DefectGen consists of two key steps: (i) training Stable Diffusion on defect-free images, and (ii) generating defect images. The detailed training procedure is outlined in Algorithm 1. The overall process of DefectGen is illustrated in Figure 1 and can be defined mathematically as below:

$$\hat{y} = (x_m \cdot x_d) + (x_g \cdot (1 - x_m)) \tag{6}$$

Algorithm 1 Training of DefectGen

Input: Defect-free image x_i, few annotated images x_a, corresponding mask x_m
Output: Defect Image x_f
Train Stable Diffusion with Defect-free Images:
  Sample $z_0 \sim E(x_i)$
  $z_T \sim$ noise(z_0, T): $x_t = \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \epsilon$
for all t from T to 0 do
  Denoise (z_T, t): $x_{t-1} = x_t - \beta_t \cdot \epsilon_\theta(x_t, t)$
end for
  $x_q = D(z_0)$
Generate Defect
  $x_f = (x_m \cdot x_a) + (x_q \cdot (1 - x_m))$
Return x_f

A. Training on Defect-free Images

In the first phase of our approach, we focus on training a Stable Diffusion model to serve as the foundational backbone of DefectGen. The primary objective at this stage is to leverage the powerful capabilities of Stable Diffusion to produce a wide variety of defect-free images, which will later be instrumental in the process of defect generation and inpainting. Stable Diffusion has demonstrated remarkable success in generating high-quality, photorealistic images across a diverse range of contexts. By integrating this model within DefectGen, we capitalize on its strengths to ensure that the generated images not only exhibit high fidelity but also maintain the rich detail and visual consistency expected in real-world imagery. Given a clean, defect-free input image $x_i \in \mathbb{R}^{W\times H\times C}$. The Stable Diffusion model processes this input to generate a new image $x_g \in \mathbb{R}^{W\times H\times C}$. This generated image x_g is the intermediary output of the proposed DefectGen.

B. Generation of Defects

The second phase of DefectGen involves two key steps: (i) defect extraction and modification, and (ii) deep blending.

(i) Defect Extraction and Modification: In the defect extraction step, a small set of defect images is utilized, each accompanied by its corresponding binary mask. The process begins by applying the binary mask to the defect image, effectively extracting only the defective region while suppressing the rest of the image. Specifically, all areas of the image outside the defect region are set to zero, leaving a clean, isolated defect. This approach ensures that only the defect is preserved, allowing for more controlled and targeted manipulation.

After isolating the defect, the next step involves modifying these extracted defects according to specific conditions. This modification can include altering the size, shape, or location of the defect within the image. By introducing variability in these parameters, a diverse set of defects can be generated. This randomization is crucial as it enables the model to handle a wide range of defects, enhancing its robustness and generalization capability in the subsequent stages of DefectGen. The output of this step is a collection of defects where each defect has been carefully extracted, modified, and positioned.

(ii) Blending. In the blending step, an advanced image blending technique is employed to seamlessly integrate the defects extracted and modified in the previous step into a generated image. By this stage, we have a diverse set of defects, each uniquely tailored in terms of size, shape, and location.

The blending process begins by randomly selecting one of the generated images from the first phase of DefectGen. This image serves as the base onto which defects will be applied. Next, one or more defects are randomly chosen from the diverse set of defects created in the extraction and modification step. These selected defects are then blended into the base image. The blending technique ensures that the defects are seamlessly integrated, preserving the visual coherence and realism of the image.

IV. EXPERIMENT

To verify the effectiveness of DefectGen, we conduct experiment on our Steel Surface dataset to extend the dataset and tested it for defect detection task.

A. Dataset

We used our Steel Surface dataset for this experiment, which comprises 400 defect-free samples. The dataset includes two types of defects: wrinkles and nozzles, with 20 samples of wrinkle defects and 25 samples of nozzle defects. All images were captured using a high-resolution camera.

To prepare the dataset, we manually annotated the defects and created binary masks. The images and masks were then resized to 600x600 pixels. Stable Diffusion was initially trained on the defect-free images to establish a baseline model. This initial training aimed to accurately capture the characteristics of defect-free steel surfaces before generating defect images. Subsequently, we generated a total of 3,000 defect images, with 1,500 images for each defect category. These synthetic images were split into training (80%), testing(10%), and

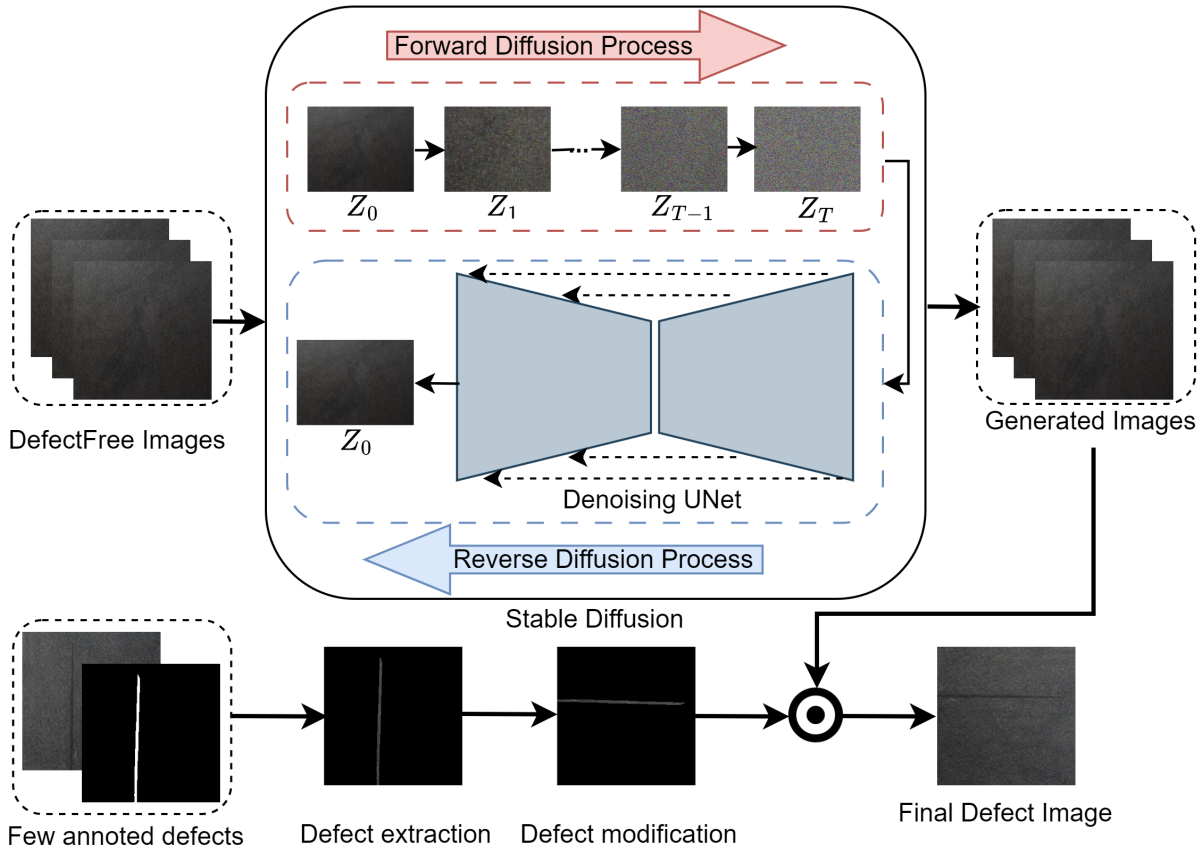

Fig. 1. Overview of the proposed DefectGen: This approach involves two main steps: i) training Stable Diffusion to generate images (top part of the figure) and ii) generating defect images (bottom part of the figure).

validation(10%) sets. The total number of images used for training, testing, and validation is summarized in Table I.

B. Implementation Details

We implemented our network and conducted experiments using Python 3.8.10 and PyTorch 1.13.1, running on a system equipped with four NVIDIA GeForce RTX 3090 GPUs, each featuring 24GB of memory. To optimize model parameters, we utilized the AdamW optimizer [14] with a learning rate of $1 \times 10^{-}5$, and default beta values set to 0.9 and 0.999. The input resolution was configured to 255×255, and we set the default batch size to 16 for training.

TABLE I
OVERVIEW OF THE STEEL SURFACE DATASET USED IN THIS EXPERIMENT

Defect category	Train	Test	Validation
Wrinkles	1,200	150	150
Nozzles	1,200	150	150

C. Result Analysis

Our main objective in this study is to generate high quality synthetic images for use in industries. Since annotating data is costly, the generated images must contain the aanotations of the defects location exactly. According to Figure 3 and 2, our approach successfully generates realistic images. The appearance of the generated images closely matches the geometries of the input guide. Furthermore, the defects seen in these images match the defects shown in the provided masks.

TABLE II
AVERAGE RESULTS FOR DEFECT DETECTION ON STEEL SURFACES, SPECIFICALLY FOR WRINKLE AND NOZZLE DEFECTS.

Used data	mAP	Precision	Recall
Real only	28.54	38.12	32.71
Synthetic only	22.32	30.25	25.7
Real + Synthetic	40.45	53.89	42.35

To evaluate the authenticity of the generated data, we utilize the Frechet Inception Distance (FID) [7]. Comparing the synthetic image distribution to the Steel surface data distribution results in an FID score of 99.57. We continue to investigate the possibility of utilizing these annotated synthesized images in an actual industrial setting, particularly through the utilization of a well-known object detection model, YoLo-NAS. We trained YoLo-NAS in four different scenarios: i) using real data for training; ii) using real data with basic augmentation techniques for training; iii) training on synthetic data only and

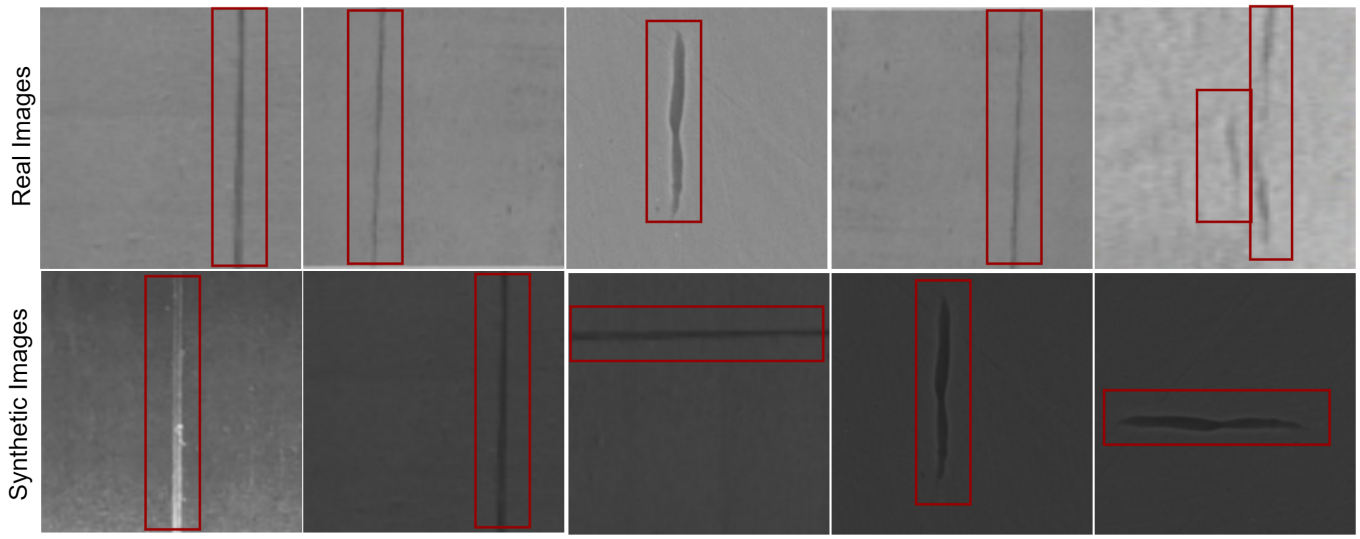

Fig. 2. Examples of real defect images and synthetic images generated by DefectGen for wrinkle defects in a steel surface dataset.

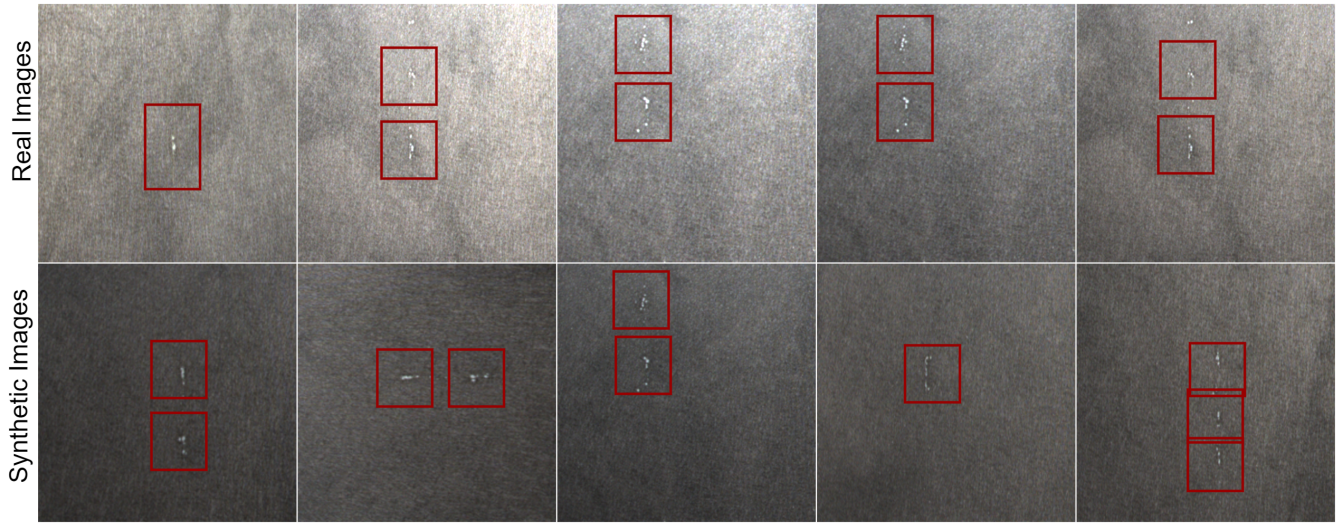

Fig. 3. Examples of real defect images and synthetic images generated by DefectGen for nozzles defects in a steel surface dataset.

iii) pre-training on synthetic data and fine-tuning on real data., results on a different test dataset show that utilizing synthetic data for training consistently improves defect detection for all considered metrics. The quantative results as illustrates in II indicate that combining real and synthetic data significantly improves defect detection performance compared to using only real or synthetic data alone. Specifically, the mean average precision (mAP) increased by approximately 10% when both real and synthetic data were used for training. This improvement highlights the benefit of incorporating synthetic images to enhance model performance.

The qualitative results are illustrated in Figures 2 and 3. Figure 2 presents a comparison between real and synthetic images for the wrinkle defect category. In this figure, the top row displays real images, while the bottom row shows synthetic images generated for wrinkles. Upon comparison, it is evident that the synthetic images closely resemble the real ones. Although there are some discrepancies, such as deeper black regions in the background of the synthetic images, the wrinkles themselves are quite similar in appearance to those in the real images. The generated wrinkles vary in size and shape, reflecting a diverse range of defect characteristics that are present in the real dataset.

Figure 3 presents a comparable evaluation for the nozzle defect category. Here, the top row shows real nozzle defect images, while the bottom row features synthetic images generated by DefectGen. As with the wrinkle images, the synthetic nozzles exhibit a high degree of similarity to the

real images, with realistic shapes and sizes, though some texture and background differences are present. These qualitative evaluations underscore the effectiveness of our synthetic data generation approach in producing realistic and varied defect images, which could enhance the performance of defect detection models.

V. Conclusion

In this paper, we have introduced DefectGen, a novel technique designed to extend steel surface defect datasets, ultimately improving the performance of steel surface defect analysis. Our approach leverages an image-to-image diffusion model, specifically Stable Diffusion, to generate highly realistic defect-free images. These images serve as the foundation for our subsequent blending technique, which seamlessly integrates defects into the generated images, resulting in a comprehensive set of defect images. The application of DefectGen to our steel surface dataset has demonstrated a significant performance improvement, with an increase of approximately 20% in defect analysis accuracy. Looking forward, we plan to extend our work by applying DefectGen to additional datasets and validating its effectiveness across different domains.

Acknowledgment

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of IITP grant funded by the Korea government(MSIT) (IITP-2024-RS-2020-II201612, 33%) and by Priority Research Centers Program through the NRF funded by the MEST(2018R1A6A1A03024003, 33%) and by the MSIT, Korea, under the ICAN support program(IITP-2024-00156394, 34%) supervised by the IITP

References

[1] H. Chen, Y. Zhang, X. Wang, X. Duan, Y. Zhou, and W. Zhu. Disenbooth: Disentangled parameter-efficient tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374*, 2023.

[2] W. Chen, H. Hu, Y. Li, N. Rui, X. Jia, M.-W. Chang, and W. W. Cohen. Subject-driven text-to-image generation via apprenticeship learning. *arXiv preprint arXiv:2304.00186*, 2023.

[3] Y. Duan, Y. Hong, L. Niu, and L. Zhang. Few-shot defect image generation via defect-aware feature manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 571–578, 2023.

[4] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[5] M. Golam, A. Riztia, P. Daely, A. Khatun, J.-H. Kim, J.-H. Lee, Y.-R. Cho, D.-S. Kim, and J. M. Lee. Development of the system architecture for black ice detection to prevent transportation calamities. In *Proceedings of the Symposium of the Korean Institute of Communications and Information Sciences*, pages 692–693, 2022.

[6] L. Han, Y. Li, H. Zhang, P. Milanfar, D. Metaxas, and F. Yang. Svdiff: Compact parameter space for diffusion fine-tuning. *arXiv preprint arXiv:2303.11305*, 2023.

[7] 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 *Advances in Neural Information Processing Systems*, volume 30, 2017.

[8] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[10] G. Huang, I. Laradji, D. Vazquez, S. Lacoste-Julien, and P. Rodriguez. A survey of self-supervised and few-shot object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4071–4089, 2022.

[11] M. A. Khatun, D.-S. Kim, J. M. Lee, and J.-H. Kim. Comparison between vision transformer and cnn for ice image classification. In *Proceedings of the Symposium of the Korean Institute of Communications and Information Sciences*, pages 1822–1823, 2023.

[12] E. Levin and O. Fried. Differential diffusion: Giving each pixel its strength. *arXiv preprint arXiv:2306.00950*, 2023.

[13] Hanxi Li, Zhengxun Zhang, Hao Chen, Lin Wu, Bo Li, Deyin Liu, and Mingwen Wang. A novel approach to industrial defect generation through blended latent diffusion model with online adaptation. *arXiv preprint arXiv:2402.19330v2*, Mar 2024. These authors contributed equally to this work.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[15] C. Mou, X. Wang, J. Song, Y. Shan, and J. Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023.

[16] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[17] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[18] S. Niu, B. Li, X. Wang, and H. Lin. Defect image sample generation with gan for improving defect recognition. *IEEE Transactions on Automation Science and Engineering*, 17(3):1611–1622, 2020.

[19] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[20] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[21] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[22] A. M. Tayeb, K.-H. Lee, J.-Y. Seo, G.-W. Kim, H.-B. Park, H.-J. Lee, and T.-H. Kim. Facial skin condition analysis based on pore and wrinkle segmentation. In *Proceedings of the Symposium of the Korean Institute of Communications and Information Sciences*, pages 1648–1649, 2023.

[23] Adnan Tayeb, Md Alam, Ayesha Khatun, Golam Mohtasin, Md Facklasur Rahaman, Md Javed Ahmed Shanto, and Dong-Seong Kim. Pureparking: A decentralized, secure framework for parking space sharing using blockchain. 06 2024.

[24] Adnan Md Tayeb, Mst Ayesha Khatun, Mohtasin Golam, Md Facklasur Rahaman, Ali Aouto, Oroceo Paul Angelo, Minseon Lee, Dong-Seong Kim, Jae-Min Lee, and Jung-Hyeon Kim. Smartrsd: An intelligent multimodal approach to real-time road surface detection for safe driving, 2024.

[25] J. Wei, F. Shen, C. Lv, Z. Zhang, F. Zhang, and H. Yang. Diversified and multi-class controllable industrial defect synthesis for data augmentation and transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4444–4452, 2023.

[26] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[27] G. Zhang, K. Cui, T.-Y. Hung, and S. Lu. Defect-gan: High-fidelity defect synthesis for automated defect inspection. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2523–2533, 2021.