

Article

Defect Synthesis Using Latent Mapping Adversarial Network for Automated Visual Inspection [†]

Seunghwan Song ¹, Kyuchang Chang ², Kio Yun ¹, Changdong Jun ¹ and Jun-Geol Baek ^{1,*}

¹ Department of Industrial and Management Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Korea

² Division of Applied Artificial Intelligence Engineering, Kangnam University, 40 Gangnam-ro, Giheung-gu, Yongin-si 16979, Korea

* Correspondence: jungeol@korea.ac.kr; Tel.: +82-2-3290-3396

[†] This manuscript is an extension of the conference paper: Song, S.; Baek, J.G. Defect Information Synthesis via Latent Mapping Adversarial Networks. In Proceedings of the IEEE International Conference on Artificial Intelligence in Information and Communication (ICAIIIC2022), Jeju Island, Korea, 21–24 February 2022.

Abstract: In Industry 4.0, internet of things (IoT) technologies are expanding and advanced smart factories are currently being developed. To build an automated visual inspection (AVI) and achieve smartization of steel manufacturing, detecting defects in products in real-time and accurately diagnosing the quality of products are essential elements. As in various manufacturing industries, the steel manufacturing process presents a class imbalance problem for products. For example, fewer defect images are available than normal images. This study developed a new image synthesis methodology for the steel manufacturing industry called a latent mapping adversarial network. Inspired by the style-based generative adversarial network (StyleGAN) structure, we constructed a mapping network for the latent space, which made it possible to compose defect images of various sizes. We discovered the most suitable loss function, and optimized the proposed method in terms of convergence and computational cost. The experimental results demonstrate the competitive performance of the proposed model compared to the traditional models in terms of classification accuracy of 92.42% and F-score of 93.15%. Consequently, the problem of data imbalance is solved, and higher productivity in steel products is expected.

Keywords: automated visual inspection; internet of things; generative adversarial networks; latent mapping adversarial networks; defect synthesis



Citation: Song, S.; Chang, K.; Yun, K.; Jun, C.; Baek, J.-G. Defect Synthesis Using Latent Mapping Adversarial Network for Automated Visual Inspection. *Electronics* **2022**, *11*, 2763. <https://doi.org/10.3390/electronics11172763>

Academic Editors: Seokjoo Shin and Haeyoung Lee

Received: 29 July 2022

Accepted: 30 August 2022

Published: 1 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In Industry 4.0, internet of things (IoT) applications are becoming important. IoT connects the physical and digital worlds, enabling smart factory development through faster communication and better analytics [1]. In general, a smart factory is one in which all internal elements are organically connected and operated intelligently based on advanced information and communication technology (ICT). Product quality must be measured in real-time to manufacture products at minimal cost and time. There is an increasing demand for steel products with better surface and shape qualities [2]. The end product of the manufacturing process is directly related to economic factors as it affects productivity. IoT applications in the steel industry can make a variety of industries more efficient and flexible, thereby increasing their productivity and yield [3,4].

Defects are physical and chemical failures caused by problems in the manufacturing process, facility, or manufacturing environment. Steel is manufactured through various processes such as rolling and forging. During this process, defects such as crazing, inclusions, pitted surfaces, rolled-in scales, and scratches occur, as shown in Figure 1 [5].

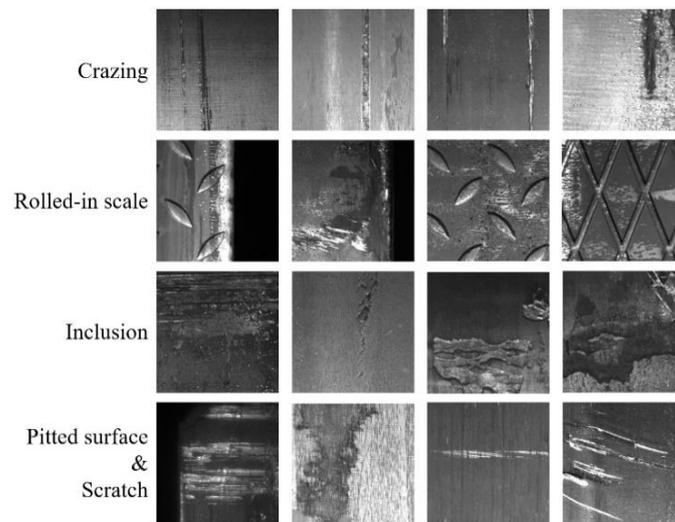


Figure 1. Types of defects in the steel manufacturing process [6].

Defect inspection, which detects defects in real-time and classifies defect types, is one of the key technologies required for smart factory implementation [7,8]. Defect detection on steel surfaces is an important task to ensure the quality of industrial production. Defect detection on a steel surface involves three preliminary steps, as shown in Figure 2. The first step is inspection, in which defects on the steel surface are detected by inspection tools [9]. The second step is review, in which images of the detected defects are captured by a specific tool. The third step is the detection and classification of defect types based on the captured images. Steel-surface defect detection processes allow engineers to perform cause analysis and defect control. However, visual inspection relies heavily on the experience and abilities of individual engineers. Additionally, this process is usually performed manually in the industry, making it unreliable and time-consuming. Therefore, automated visual inspection (AVI) targeting the surface quality has emerged as a standard configuration for steel manufacturing mills to improve product quality and promote production efficiency [10]. AVI, which performs classification through image-based algorithms, is not only widely applied to the steel manufacturing process but also to glass, fiber, and semiconductor production processes [11].

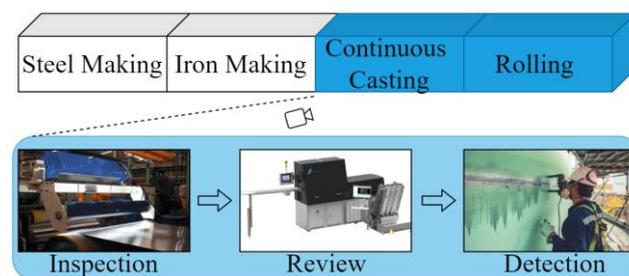


Figure 2. Steel manufacturing process and defect detection steps [6].

Although a convolutional neural network (CNN)-based AVI model exhibits excellent classification performance for numerous defect types, it has two practical problems in the steel manufacturing process. First, the frequency of defect data occurrence is extremely low, and very little data can be used for the development of a deep learning model [12]. In general, sufficient training data for both defect and normal classes are required to improve the classification performance of deep learning models [13]. However, in the actual industry, the quantity of defective data is minimal compared to that of normal data. When performing AVI with only data collected from the industry, data imbalance

issues can result in lower learning rates for defect types and poor performance. Therefore, it is necessary to balance the normal and defective classes. Class imbalance refers to a substantial proportional difference between the classes in the total dataset. When the class distribution is unbalanced, the model is trained with a bias toward the majority class, classifying the class with a large amount of data; however, the opposite is true for the minority class. Furthermore, an imbalanced class distribution can lead to serious type II errors. Therefore, preprocessing for class imbalance is essential for improving the overall classification performance in defect detection.

Second, the steel defect data consisted of defects of various sizes. Large defects can be easily generated using a simple generator when a generative model is used to solve the imbalance problem. However, the generation of small-sized defects is significantly influenced by the type of generative model used [12]. In particular, circumstances such as the cold rolling process, where the end product is 2 m wide and the size of the defect is approximately 0.2 mm, require a sophisticated classifier [14]. This study proposes a novel deep learning model for synthesizing defect data in a steel manufacturing process.

In this study, we propose a latent mapping adversarial network to overcome two practical problems in the steel manufacturing process. Our methodology was inspired by the style-based generative adversarial network (StyleGAN), which is state-of-the-art technology in the field of data generation [15]. The proposed method uses a mapping network in the latent space of a generator network. As the latent space passes through the mapping network, it becomes possible to learn the disentanglement of the training data distribution. This is the first step in the explicit learning of real data. Mapping networks allow the sophisticated generation of small defects. Our methodology also focuses on learning stability. We use the Wasserstein distance as a distribution-distance metric instead of the Jensen–Shannon (JS) divergence. Generative adversarial network (GAN) is a method of generating data based on distribution. GAN showed excellent performance mainly in image generation field. GAN has a vanishing gradient problem that occurs via the instability of model training due to the biased learning of the generator and discriminator. The Wasserstein distance solves problems such as the vanishing gradient and mode collapse witnessed in vanilla GAN [16]. A vanishing gradient is an error that occurred during the gradient descent in training, and mode collapse means a problem in which the same result is always output. The advantages of using the Wasserstein distance are discussed in Section 3.

Images of the flat steel plates were used in the production process to demonstrate the performance of the proposed method. The data generation aspect of the proposed method was first evaluated using a quantitative evaluation metric, the Fréchet inception distance (FID), and visual results. In addition, a second evaluation was performed on classification performance using a simple CNN structure [17]. Finally, to reduce the computational cost, we determined the optimal sizes of the latent space and mapping network for the data used in the experiment.

The contribution of this research is as follows:

1. To propose a novel technique in steel manufacturing for the data imbalance problem;
2. To generate effective training data to detect detailed defects;
3. To achieve the highest efficiency at the optimal time, setting the optimal potential space and mapping network;
4. To verify the generative model of the defect data using quantitative evaluation metrics, visual results, and classification results.

The remainder of this paper is organized as follows. In Section 2, we introduce previous studies on AVI. In addition, we examined the background of our research and reviewed the previous studies. The proposed methodology is described in Section 3. In Section 4, the performance of the steel surface defect dataset was evaluated using the proposed method. Finally, conclusions based on the experimental results and directions for future research are presented in Section 5.

2. Related Work

In the Introduction, two problems that need to be solved in this study are discussed. This section addresses related work on the class imbalance problem. In particular, previous studies conducted on steel manufacturing are explored. Figure 3 shows the data imbalance of the Severstal dataset in a visual context [18]. The defect data consisted of 53.04% (5680 EA) of the total data, and the proportions of each class were as follows: 12.64% (718 EA) for class 1: crazing, 3.48% (198 EA) for class 2: rolled-in scale, 72.59% (4123 EA) for class 3: pitted surface and scratch, and 11.29% (641 EA) for class 4: inclusion. In this study, the Severstal dataset was sampled and used to create a class imbalance problem. Sampling was performed only in the areas where the defect was present. Section 4 describes the sampling method used in this study.

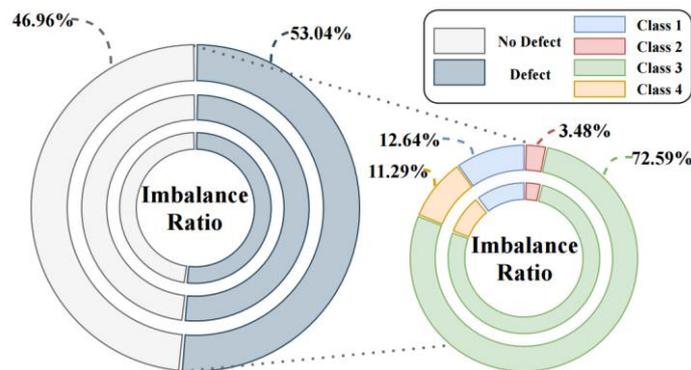


Figure 3. Imbalance ratio of steel defect images in the Severstal dataset.

The numerous solutions proposed to solve class imbalance problems in AVI can be divided into two methods: correcting the model itself and directly processing data [19]. In the former, the data instances of different classes are treated differently in a manner similar to active learning or kernel-based methods. In the latter, the direct processing of data utilizes methods such as sampling or data generation to directly control the number of instances.

Sampling is a method used to correct the bias between classes in data with an overwhelmingly small proportion of abnormal data compared with normal data. Representative methods for dealing with class imbalance include oversampling and undersampling. Oversampling is a method for creating new data of the minority class to even the class ratio, whereas undersampling is a method for removing existing data of the majority class to match the ratio. Because undersampling reduces the amount of sample data from the majority class, it has the advantage of reducing the model training time. However, it can also distort data features by removing crucial information. Regarding oversampling, the risk of data distortion is relatively small because it creates new data while preserving original data. The oversampling methods mainly used for AVI include random oversampling, synthetic minority oversampling technique (SMOTE) [20], and adaptive synthetic sampling approach (ADASYN) [21]. Random oversampling increases the amount of minority class data by randomly selecting and replicating a sample from a minority class. SMOTE synthesizes data by selecting random data belonging to a minority class and randomly selecting the closest top k number of data. ADASYN is a method of adaptively synthesizing k data from marginal minority data according to the number of majority classes after calculating the ratio of the data of a majority class. However, this oversampling method for image data generates low-resolution images.

To simplify this problem, we use a technique to handle the raw image. This method is called data augmentation and has been commonly used as a model regularization technique in recent studies [22]. Some common augmentation methods include flipping an image vertically or horizontally, shifting the image vertically or horizontally, and slightly rotating or zooming it. This method helps the training model be robust to small changes in the

image. However, simple geometric transformations do not significantly change the image characteristics, making it impossible to identify additional features.

Among data generation methods, GAN is an algorithm of great interest [23]. The GAN generates data based on the distribution and exhibits excellent performance in image generation. Various GANs have been studied previously. A deep convolutional GAN (DCGAN) was used for ball-bearing failure detection [24]. In addition, a wafer defect image was adaptively generated using conditional GAN (CGAN) [25]. The progressive growing GAN (PGGAN) increased the model training speed by gradually increasing the generator and discriminator and producing a high-quality image [26]. In addition, to address shortcomings such as the vanishing gradient or mode collapse of a GAN, Wasserstein GAN (WGAN) was proposed [27].

Liu et al. [28] proposed a GAN-based one-class classification method for detecting strip steel surface defects. Their model achieved 94% good test results on images provided by the Handan Iron and Steel Plant. Lai et al. [29] proposed a new detection method using a GAN and statistical-based representation learning mechanism. This method achieved an accuracy of 93.75% on the solar panel dataset. Akhyar et al. [30] proposed a method for generating more detailed contours in the original steel image. The method achieves better performance and effectiveness in terms of processing time compared to the original method.

The AVI is critical for effective and efficient maintenance, repair, and operation in advanced manufacturing. However, AVI is often constrained by the lack of defect samples [31]. This study compares and applies existing GAN-based generation models, whose data generation performance has already been verified, and finds an optimal generation model suitable for field data applications. The main contribution of this study is the generation of effective training data for detecting detailed defects.

3. Latent Mapping Adversarial Network

This section describes the framework of the latent mapping adversarial network, which is an approach for solving the imbalance problem in defect images. Figure 4 shows the schematic of the overall structure of the proposed method. A GAN is a neural network in which the generator and discriminator adversarially learn from each other. The generator is trained to generate an image similar to the real image, whereas the discriminator is trained to discriminate between real and generated images. The components of the proposed method are as follows. (1) Generator: this improves the quality of data generation by adopting a mapping network structure for latent space. (2) Discriminator and loss function: using the Wasserstein distance with an applied gradient penalty, the imbalanced loss function problem that occurs when the discriminator is backpropagated is addressed. The mapping network for the latent space is discussed in Section 3.1, and the imbalanced loss function is discussed in Section 3.2.

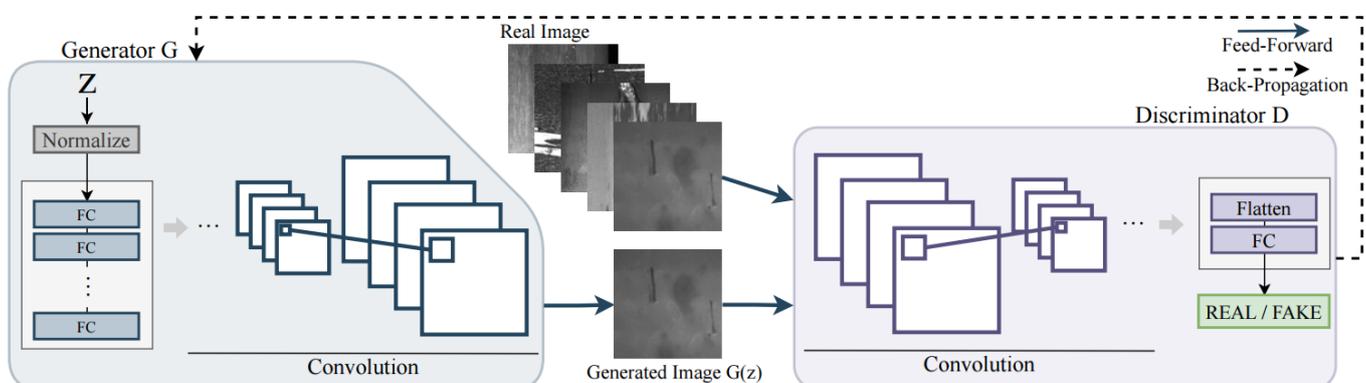


Figure 4. Latent mapping adversarial network framework for defect synthesis [6].

3.1. Mapping Network for Latent Space

Defects on the steel surface significantly affect the quality of the final steel product. Therefore, it is crucial to correctly detect defects to ensure the quality of the final product and to prevent the delivery of defective products to customers. However, because of an imbalance in the steel surface defect data, an increase in the misclassification of such data leads to a deterioration of the classification performance. Therefore, an oversampling method is required to generate the defect data.

In this study, a mapping network structure was used for the latent space to improve the quality of generated data. The latent space of a well-trained GAN model contains linear subspaces that permit direct variation adjustments [15]. However, direct control of the latent space z is impossible because z of a vanilla GAN tends to form the training data into a single Gaussian distribution. The mapping network overcomes this problem by preventing latent space z from entering the generator as an input value. Instead, we input w , which passes through the mapping network, as an input value to the generator. The latent space z cannot accurately match the feature distribution of the training data, whereas w can because it undergoes a nonlinear transformation through the mapping network. Therefore, the disentanglement characteristic of w , which is suitable for training data, leads to improved data generation. In summary, in the vanilla GAN model generator structure, the latent space is passed through a mapping network composed of fully connected layers.

This approach may seem to be simple. We used a GAN to learn the distribution of the data. When we generate a noise vector and input it into the GAN, we can generate random images that are similar to our training data but are not present in the training data. However, it is difficult to create random images with the desired characteristics. The reason for this result is that z is related to other features. One of the reasons why the axis is entangled is that the degree of z is insufficient. The mapping network disentangles the axes by making the z degree sufficient. Therefore, it achieves performance improvement in terms of generative training data.

3.2. Imbalanced Loss Function

Existing oversampling methods do not use data distributions. Additionally, GAN problems, vanishing gradient, and mode collapse have a detrimental effect on the quality of the generated data [27]. Vanishing gradient refers to a problem that occurs when the discriminator learns to perfect, as shown in Equation (1). If the discriminator D is perfect, the loss function of the GAN in Equation (2) approaches zero, and the gradient is not obtained in the learning process.

$$D(x) = \mathbf{1}^{\forall} x \in p_r, D(x) = \mathbf{0}^{\forall} x \in p_g \quad (1)$$

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_g} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2)$$

In Equations (1) and (2), p_r denotes the distribution of the real data, p_g denotes the distribution of the generated data, and p_z denotes the distribution of latent space.

Mode collapse, another characteristic problem of a GAN, occurs when the generator always outputs the same result during the learning process because the GAN uses the JS divergence as a distance metric. In this study, 1-Wasserstein was used as the distance metric instead of the JS divergence to deviate from the problems of gradient loss and mode collapse. However, the 1-Wasserstein has a weight clipping problem. The gradient penalty (GP) technique was used to solve the weight-clipping problem of 1-Wasserstein [16]. Therefore, the imbalance loss function, WGAN-GP, is expressed in Equation (3) and is learned in the direction of minimizing this constraint.

$$L = \mathbb{E}_{x \sim p_r} [D(x)] - \mathbb{E}_{z \sim p_g} [D(z)] + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right], \text{ where } \hat{x} = tx + (1-t)z \text{ with } 0 \leq t \leq 1 \quad (3)$$

In Equation (3), x denotes the actual data, and z denotes the data generated in the latent space. The remainder of Equation (3) denotes the part for the gradient ∇ of the

discriminator D with \hat{x} uniformly sampled between x and z at the ratio of t . When the L2 regularization (L2 norm) of this gradient has a value other than 1, it can be optimized by assigning a penalty equal to λ . Consequently, manipulating the loss function to obtain a meaningful value when the two distributions do not overlap in a low-dimensional manifold can solve the loss of slope and mode collapse problems.

Therefore, procedure of the proposed method is summarized as follows. First, the generator of the proposed method generates defect images. A mapping network is used for more accurate generation. Next, the discriminator differentiates the generated image from the real image. This process continues until the generator generates defect images similar to the real images.

4. Evaluation and Comparison

All experiments were performed using the PyTorch software package [32] and scikits learn (Sklearn) [33], Pandas library [34], together with Python3 language, running on a desktop with Intel(R) Core(TM) i7-9700K CPU @ 3.60 GHz, 32 GB RAM with NVIDIA GeForce RTX 3080 10 GB. For comparison, we also implemented other leading GANs using PyTorch.

4.1. Datasets

The data used for performance verification in this study were acquired from the Severstal steel manufacturing process [18]. These data were collected using a high-frequency camera that captured images of flat sheet steel during the production process. This dataset is typically subjected to defect location and type prediction in steel manufacturing. The dataset contains a single class of defect-type data, multiple classes of defect-type data, and non-defect-type data. Figure 5 shows an example of the data used in the experiment.

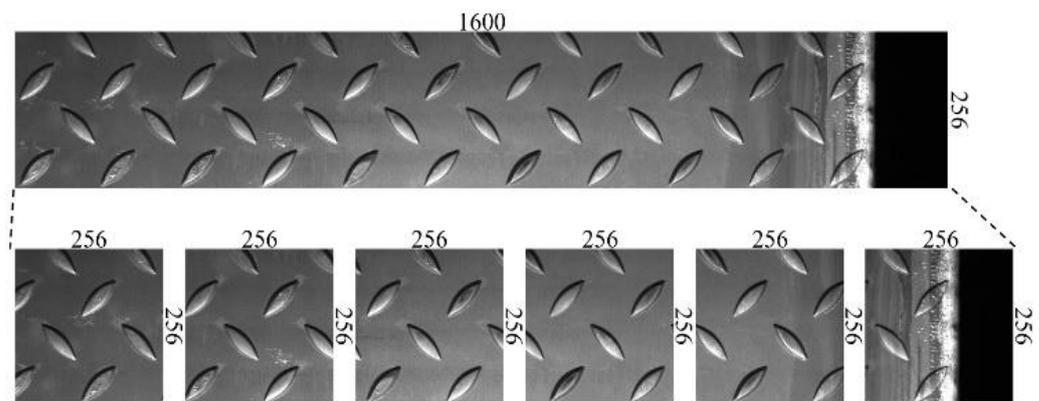


Figure 5. Example of cropped steel defect images (256×1600 to 256×256) [6].

The steel defect data comprises a schematic diagram of each class of defect data, from tiny defects to large defects. In this study, image data of size 256×1600 was cropped into a square image of size 256×256 , tailored for utilization as an input value. Overlapping parts were not allowed to crop the image and the last part was not used. The dataset was sampled according to the ratios introduced in Section 2. In the cropped defect image, sampling was performed only in the areas where the defect was present.

4.2. Experimental Design

The cropped normal image comprised 86.64% (18,884 EA) of the total data, and the cropped defect image comprised 13.36% (2913 EA) of the total data. The proportions of each class were as follows: 13.01% (379 EA) for class 1: crazing, 2.99% (87 EA) for class 2: rolled-in scale, 72.98% (2126 EA) for class 3: pitted surface and scratch, and 11.02% (321 EA) for class 4: inclusion. The preprocessed dataset was partitioned at a ratio of $Data_{train} : Data_{test} = 7 : 3$. The experiment consisted of two steps. The first-stage exper-

iment verified the generator model of the proposed method. The superior performance of the proposed method compared with other GAN-based generator models was demonstrated. Each GAN layer was uniformly composed of five layers, and 100-dimensions were used for the latent space. In the experiment, the number of defect images synthesized was the same as that of the normal image. For the optimization function, RMSProp, which is frequently used in GAN models, is used.

The second-stage experiment determined the optimal latent space size and mapping network. By structuring part of the proposed method with a mapping network, the proposed method was able to acquire disentanglement features. The optimal size of the initial latent space and mapping network was determined experimentally using the proposed model. All experiments were evaluated using the quantitative evaluation metric, FID. During data division, the seed was changed, and the average value of the ten performed results was used as the final metric.

4.3. Performance Measurement Metric

A confusion matrix, as shown in Table 1, was used to evaluate the classification performance of the model. Because this study detects defect data, there are two types of errors: false positives detecting normal data as defect data and false negatives detecting defect data as normal data. In this study, the accuracy and F-score were used as the detection performance evaluation metrics. For each method, 10-fold cross-validation was applied and the average of the results was used.

Table 1. Confusion matrix for defect detection.

	Actual: Defect	Actual: Normal
Predicted: Defect	True Positive	False Positive
Predicted: Normal	False Negative	True Negative

Manufacturing data are primarily comprised of normal data. However, in many cases, abnormal data are more critical for defect control than normal data. This imbalance becomes a problem as it leads to an increase in the misclassification error rate of abnormal data, consequently degrading the overall classification performance. In this study, the oversampling method, which randomly generates abnormal data using the GAN model, was used to solve the imbalance of the abnormal data.

Early GANs were accompanied by problems such as instability of learning and mode collapse, resulting in difficulties in performance evaluation [23]. To address such problems, the development of various GAN models on top of inception scores (IS) and FID using the inception model has made it possible to evaluate the performance of GANs [17]. The inception model, which is widely used for transfer learning and fine-tuning, is a CNN model that pre-trains ImageNet data. ImageNet consists of 1000 classes and 1.2 million images. When an image is input into the model, the inception model outputs probability vectors belonging to each of the 1000 classes. Using the generated image as an input value to the inception model, IS can be calculated as shown in Equation (4).

$$\text{Inception Score} = \exp\left(\mathbb{E}_{z \sim p_g} KL(p(y|z) || p(y))\right) \quad (4)$$

In Equation (4), $p(y|z)$ is the conditional class distribution and $p(y)$ is the marginal class distribution. The inception score can have a value between 1 and 1000. However, the inception score has the disadvantage of not using real data distributions. In this study, the shortcomings of IS were overcome using the FID, which is a measure of the difference between the two normal distributions, as shown below.

$$\text{FID} = d^2((m, C), (m_w - C_w)) = \|m - m_w\|_2^2 + \text{Tr}\left(C + C_w - 2(CC_w)^{1/2}\right) \quad (5)$$

A smaller FID indicates better quality, and (m, C) and $(m_w - C_w)$ denote the mean and covariance of the distribution between generated and real images, respectively. The results of the FID evaluation using steel defect images are shown in Figure 6.

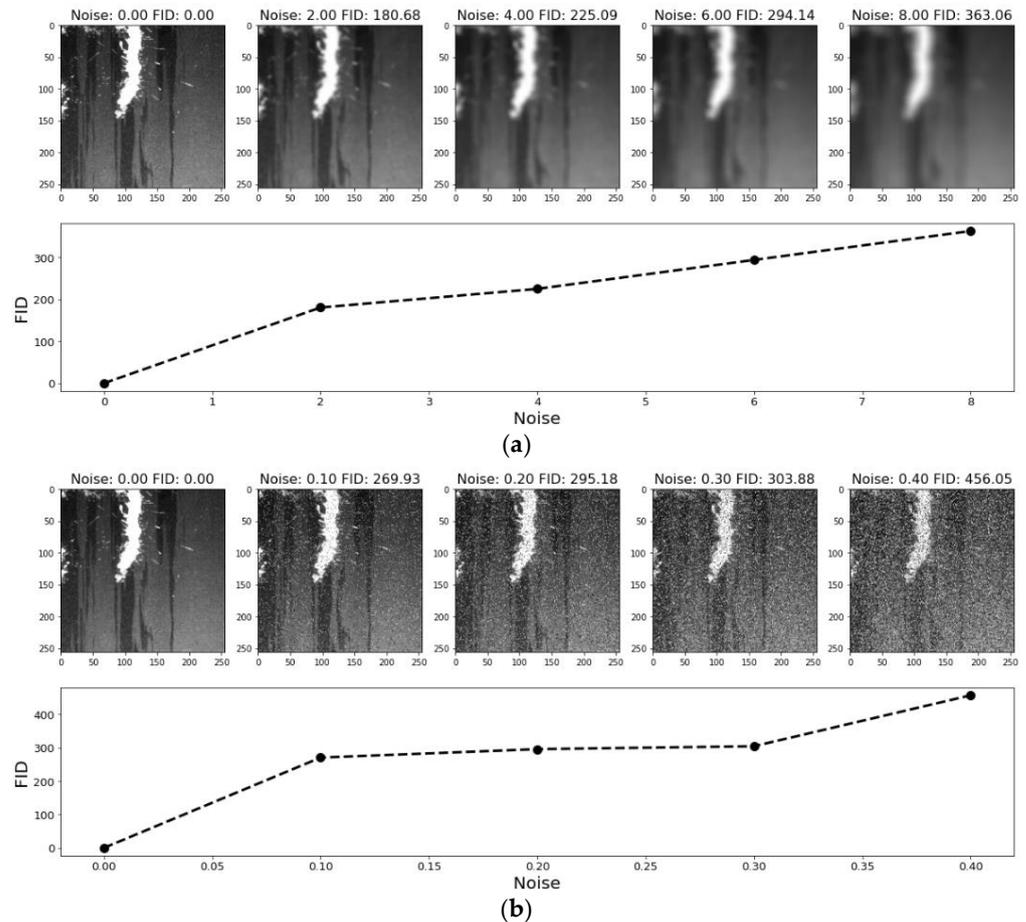


Figure 6. Example of Fréchet inception distance (FID) evaluation using steel defect images: description of (a) Gaussian blur; (b) salt and pepper.

Figure 6 shows the noise of (a) Gaussian blur and (b) salt and pepper added randomly to the raw image. Depending on the noise intensity, an increase in the discrepancy between the generated data and the raw image can be observed. As it is widely accepted that FID captures the quality of generated data better than IS, this study adopted FID as a measure to assess the quality of generated images.

4.4. Experimental Results

4.4.1. Performance Compared to Generative Model

The proposed method uses a mapping network structure and an imbalanced loss function (WGAN-GP) to improve the data quality. The latent space used in previous GAN models exhibited difficulties in avoiding entanglement owing to its tendency to follow the probability density of the training data. However, we used a mapping network to solve this problem and demonstrated disentanglement of the latent space. Therefore, direct adjustment to these changes is possible.

Table 2 lists the results of comparing the proposed method with the vanilla GAN, DCGAN, and DCGAN+WGAN-GP. The baseline is the vanilla GAN and DCGAN, in which a deep convolutional structure is added to the baseline. DCGAN+WGAN-GP is the loss function of the DCGAN changed to WGAN-GP. The proposed method adds a mapping network composed of eight fully connected layers to previous methods. The

control group was constructed as follows to evaluate briefly the effect of each method, which also demonstrates the gradual evolution of the GAN-based model.

Table 2. Comparison average Fréchet inception distance (FID) of Generative Models [6].

Method	FID
Baseline (GAN)	105.17
DCGAN	31.47
DCGAN+WGAN-GP	26.64
Proposed Method	15.81

As shown in Table 2, the proposed method exhibits excellent performance in terms of average FID. As each method was added sequentially, it was confirmed that the FID also improved sequentially. Figure 7 shows a visual comparison of the real image with the generation results of each method.

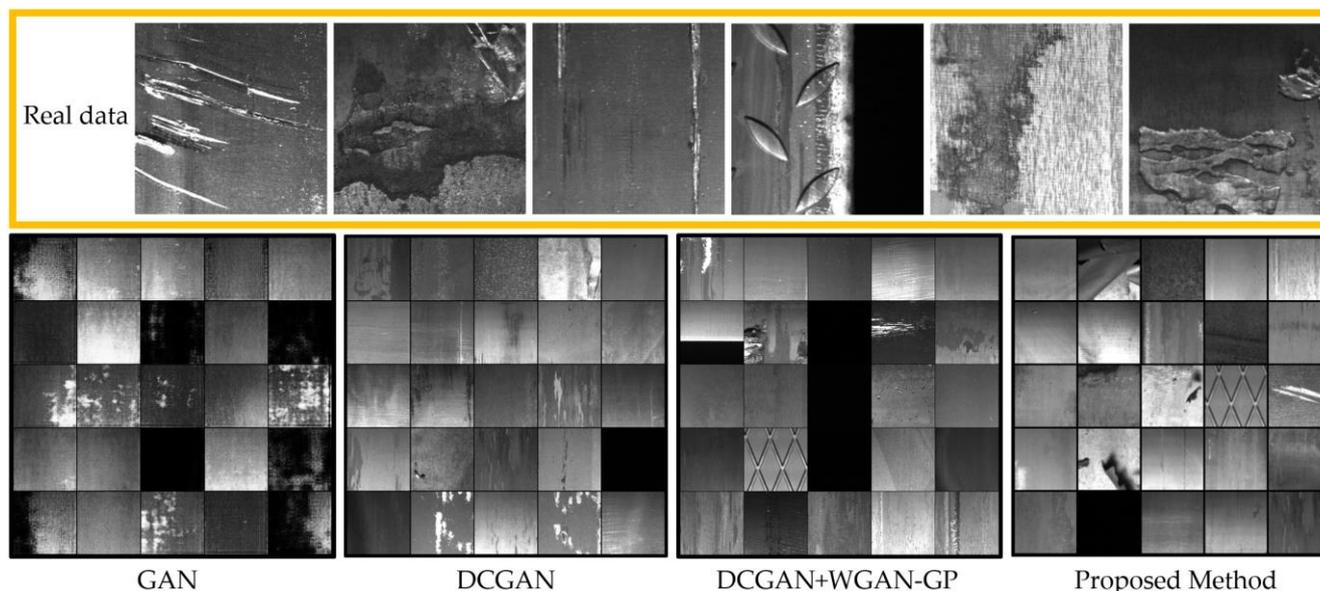


Figure 7. Visual comparison of the results of each method.

Comparing the creation results of each method as a 5×5 matrix, it was difficult to recognize a large difference when visually confirmed. As shown in Table 3, we applied the generated data to the classification task. For this task, we used a simple fully convolutional network (FCN) [35]. We trained the FCN algorithm on the generated samples and tested the accuracy of the real image. The performance comparison is displayed in order of accuracy and F-score. In the image synthesized using the generative model, both the normal and defective images have the same ratio.

Table 3. Classification result for the results of generative models. avg. (\pm std.).

Method	Classification Accuracy	F-Score
Baseline (GAN)	0.7312 (0.0237)	0.7546 (0.0348)
DCGAN	0.8566 (0.0225)	0.8486 (0.0212)
DCGAN+WGAN-GP	0.8891 (0.0187)	0.8958 (0.0201)
Proposed Method	0.9242 (0.0161)	0.9315 (0.0139)

As listed in Table 3, the FCN algorithm for each method performed 10-fold cross-validation. It was confirmed that the proposed method showed superior performance

compared to the other methods in terms of the average accuracy and F-score. The proposed method can achieve a significant performance increase of approximately 18%p compared to the baseline model. In addition, the performance was improved by 3%p using a latent mapping adversarial network. Therefore, the proposed method generates images in a manner similar to an actual image.

4.4.2. Optimal Latent Space and Mapping Network Size

The proposed method improves image generation quality by adopting a mapping network structure. Therefore, the optimization of the latent space, where random vectors generate images similar to real images, is possible. In general, a sufficiently large latent space can adequately express the characteristics of real data, leading to the use of a 100-dimensional size for the general latent space. However, the adoption of image generation in steel manufacturing requires accurate and expeditious processing. Thus, there is a need for image generation that performs well even with a simple structure. Because the size of a latent space directly affects the number of parameters, convergence speed, and computation time, determining the optimal size of the latent space is a necessary task.

In this experiment, we attempted to determine the optimal size for the latent space and mapping network. Table 4 lists the results of the experiment adjusting the mapping network size to 0, 2, 4, and 8, and correspondingly adjusting the dimension size of the latent space to 2, 5, 10, 50, and 100.

Table 4. Comparison average Fréchet inception distance (FID) of latent space and mapping network size.

Mapping Network	Latent Space				
	2	5	10	50	100
Traditional	165.11	63.26	43.57	29.21	17.42
Style-based 2	62.87	31.98	11.87	9.65	7.86
Style-based 4	49.52	26.34	7.97	7.22	7.21
Style-based 8	49.19	20.12	7.46	6.94	7.16

In Table 4, ‘traditional’ exhibits the result of using the latent space of the general GAN without a mapping network, and ‘style-based’ indicates the size of the mapping network used. The evaluation was performed using the FID, where it is known that the lower the FID, the more similar the generated data are to the real data. Excellent performance was achieved using a mapping network for all latent space sizes. In a common trend for all methods, the performance tended to improve as the size increased to 10-dimensions. When the mapping network is not used, the performance continuously improves to 100-dimensions. However, when using a mapping network, there was only a slight increase in performance after 10-dimensions. In addition, the mapping network exhibited the best performance when composed of eight fully connected layers. The latent space and mapping network sizes are closely related to computation time. Therefore, to accommodate the need for accurate and expeditious processing characteristics of the steel manufacturing process, the proposed method consisted of 50 latent spaces and an 8-layer mapping network. Consequently, it was confirmed that the proposed method generates high-quality images.

5. Conclusions

This study proposes a method to address the imbalance that exists in defect detection during the steel manufacturing process. This method improves the quality of the generated images by adopting a mapping network. Simultaneously, we achieved accurate and expeditious processing by determining the optimal size of the latent space and the mapping network. The quality of the generated images was evaluated using the quantitative metric FID, visual results, and classification performance. The experimental results demonstrated

the competitive performance of the proposed model compared to the traditional models in terms of classification accuracy of 92.42% and F-score of 93.15%.

The method proposed in this paper applies to AVI problems in various manufacturing processes, particularly those with inherent imbalance problems. In addition, owing to its practicality, the proposed method is highly applicable to various fields other than AVI. Because follow-up maintenance costs can be reduced, productivity and yield improvements are expected. Furthermore, real time measurements of the quality of steel can be performed using data collected from IoT sensors, enabling the development of smart factories. In future research, we intend to derive quality evaluation metrics suitable for manufacturing the image data.

Author Contributions: Conceptualization, S.S. and J.-G.B.; Methodology, S.S. and J.-G.B.; Formal analysis, S.S. and K.C.; Validation, K.C., K.Y. and J.-G.B.; Investigation, S.S. and C.J.; Writing—original draft preparation, S.S. and J.-G.B.; Writing—review and editing, S.S., K.C., K.Y. and J.-G.B. Supervision, J.-G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT) (NRF-2022R1A2C2004457, NRF-2021R1A6A3A13045200). This work was also supported by Brain Korea 21 FOUR and Samsung Electronics Co., Ltd. (IO201210-07929-01).

Data Availability Statement: Dataset from: <https://www.kaggle.com/c/severstal-steel-defect-detection> (accessed on 20 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brozzi, R.; Forti, D.; Rauch, E.; Matt, D.T. The Advantages of Industry 4.0 Applications for Sustainability: Results from a Sample of Manufacturing Companies. *Sustainability* **2020**, *12*, 3647. [CrossRef]
2. Akhyar, F.; Lin, C.Y.; Muchtar, K.; Wu, T.Y.; Ng, H.F. High efficient single-stage steel surface defect detection. In Proceedings of the 2019 International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–4. [CrossRef]
3. Tortorella, G.L.; Giglio, R.; van Dun, D. Industry 4.0 adoption as a moderator of the impact of lean production practices on operational performance improvement. *Int. J. Oper. Prod. Manag.* **2019**, *39*, 860–886. [CrossRef]
4. Beliatas, M.J.; Jensen, K.; Ellegaard, L.; Aagaard, A.; Presser, M. Next generation industrial IoT digitalization for traceability in metal manufacturing industry: A case study of industry 4.0. *Electronics* **2021**, *10*, 628. [CrossRef]
5. Neogi, N.; Mohanta, D.K.; Dutta, P.K. Review of vision-based steel surface inspection systems. *EURASIP J. Image Video Process* **2014**, *1*, 50. [CrossRef]
6. Song, S.; Baek, J.G. Defect Information Synthesis via Latent Mapping Adversarial Networks. In Proceedings of the IEEE International Conference on Artificial Intelligence in Information and Communication (ICAIIIC2022), Jeju Island, Korea, 21–24 February 2022; pp. 17–22. [CrossRef]
7. Yun, J.P.; Shin, W.C.; Koo, G.; Kim, M.S.; Lee, C.; Lee, S.J. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *J. Manuf. Syst.* **2020**, *55*, 317–324. [CrossRef]
8. Li, S.; Wu, C.; Xiong, N. Hybrid Architecture Based on CNN and Transformer for Strip Steel Surface Defect Classification. *Electronics* **2022**, *11*, 1200. [CrossRef]
9. Wang, J.; Yang, Z.; Zhang, J.; Zhang, Q.; Chien, W.T.K. AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 310–319. [CrossRef]
10. Luo, Q.; Fang, X.; Liu, L.; Yang, C.; Sun, Y. Automated visual defect detection for flat steel surface: A survey. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 626–644. [CrossRef]
11. Cheon, S.; Lee, H.; Kim, C.O.; Lee, S.H. Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 163–170. [CrossRef]
12. Xu, Z.J.; Zheng, Z.; Gao, X.Q. Operation optimization of the steel manufacturing process: A brief review. *Int. J. Miner. Metall. Mater.* **2021**, *28*, 1274–1287. [CrossRef]
13. Zhang, E.; Li, B.; Li, P.; Chen, Y. A deep learning based printing defect classification method with imbalanced samples. *Symmetry* **2019**, *11*, 1440. [CrossRef]
14. Kang, D.; Jang, Y.J.; Won, S. Development of an inspection system for planar steel surface using multispectral photometric stereo. *Opt. Eng.* **2013**, *52*, 039701. [CrossRef]

15. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410. [[CrossRef](#)]
16. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5769–5779.
17. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
18. Severstal: Steel Defect Detection. In Kaggle. Available online: <https://www.kaggle.com/c/severstal-steel-defect-detection> (accessed on 20 July 2022).
19. Kondo, N.; Harada, M.; Takagi, Y. Efficient training for automatic defect classification by image augmentation. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 226–233. [[CrossRef](#)]
20. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [[CrossRef](#)]
21. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the IEEE international joint conference on neural networks, Hong Kong, China, 1–8 June 2008; pp. 1322–1328. [[CrossRef](#)]
22. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 113–123. [[CrossRef](#)]
23. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural. Inf. Process. Syst.* **2014**, *27*, 2672–2680.
24. Viola, J.; Chen, Y.; Wang, J. FaultFace: Deep convolutional generative adversarial network (DCGAN) based ball-bearing failure detection method. *Inf. Sci.* **2021**, *542*, 195–211. [[CrossRef](#)]
25. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
26. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
27. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the 2017 International conference on machine learning (PMLR), Sydney, Australia, 6–11 August 2017; pp. 214–223.
28. Liu, K.; Li, A.; Wen, X.; Chen, H.; Yang, P. Steel Surface Defect Detection Using GAN and One-Class Classifier. In Proceedings of the 2019 25th International Conference on Automation and Computing (ICAC), Lancaster, UK, 5–7 September 2019; pp. 1–6. [[CrossRef](#)]
29. Lai, Y.T.K.; Hu, J.S. A Texture Generation Approach for Detection of Novel Surface Defects. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 4357–4362. [[CrossRef](#)]
30. Akhyyar, F.; Furqon, E.N.; Lin, C.-Y. Enhancing Precision with an Ensemble Generative Adversarial Network for Steel Surface Defect Detectors (EnsGAN-SDD). *Sensors* **2022**, *22*, 4257. [[CrossRef](#)]
31. Zhang, G.; Cui, K.; Hung, T.Y.; Lu, S. Defect-GAN: High-fidelity defect synthesis for automated defect inspection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2021; pp. 2524–2534. [[CrossRef](#)]
32. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
33. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
34. McKinney, W. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; pp. 51–56. [[CrossRef](#)]
35. Smith, K.E.; Smith, A.O. Conditional GAN for timeseries generation. *arXiv* **2020**, arXiv:2006.16477.