applied sciences

MDPI

Article

Deep Convolutional Generative Adversarial Networks-Based Data Augmentation Method for Classifying Class-Imbalanced Defect Patterns in Wafer Bin Map

Sangwoo Park [1] and Cheolwoo You [2,*]

[1] C4I R&D Center, LIG Nex1 Co., Ltd., Seongnam 13488, Republic of Korea; qkrtkddn0415@gmail.com
[2] Department of Information and Communications Engineering, Myongji University, Yongin 17058, Republic of Korea
* Correspondence: cwyou@mju.ac.kr

Abstract: In the semiconductor industry, achieving a high production yield is a very important issue. Wafer bin maps (WBMs) provide critical information for identifying anomalies in the manufacturing process. A WBM forms a certain defect pattern according to the error occurring during the process, and by accurately classifying the defect pattern existing in the WBM, the root causes of the anomalies that have occurred during the process can be inferred. Therefore, WBM defect pattern recognition and classification tasks are important for improving yield. In this paper, we propose a deep convolutional generative adversarial network (DCGAN)-based data augmentation method to improve the accuracy of a convolutional neural network (CNN)-based defect pattern classifier in the presence of extremely imbalanced data. The proposed method forms various defect patterns compared to the data augmentation method by using a convolutional autoencoder (CAE), and the formed defect patterns are classified into the same pattern as the original pattern through a CNN-based defect pattern classifier. Here, we introduce a new quantitative index called PGI to compare the effectiveness of the augmented models, and propose a masking process to refine the augmented images. The proposed method was tested using the WM-811k dataset. The proposed method helps to improve the classification performance of the pattern classifier by effectively solving the data imbalance issue compared to the CAE-based augmentation method. The experimental results showed that the proposed method improved the accuracy of each defect pattern by about 5.31% on average compared to the CAE-based augmentation method.

Keywords: wafer bin map; wafer defects; deep learning; data imbalance; data augmentation

check for updates

Citation: Park, S.; You, C. Deep Convolutional Generative Adversarial Networks-Based Data Augmentation Method for Classifying Class-Imbalanced Defect Patterns in Wafer Bin Map. *Appl. Sci.* **2023**, *13*, 5507. https://doi.org/10.3390/app13095507

Academic Editor: Alexandre Carvalho

Received: 18 March 2023
Revised: 21 April 2023
Accepted: 27 April 2023
Published: 28 April 2023

Copyright: © 2023 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/).

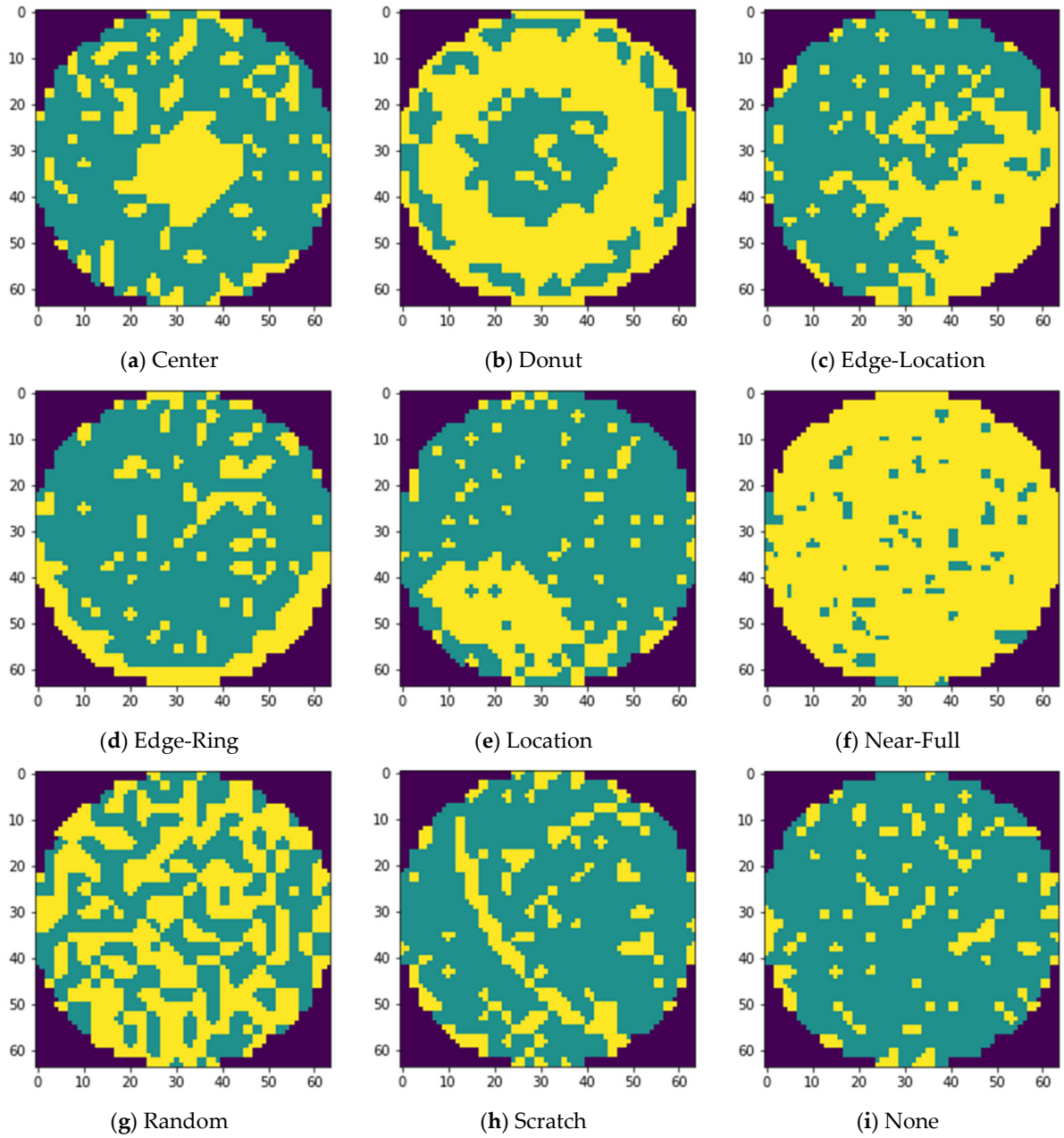
1. Introduction

As the Fourth Industrial Revolution has continued to develop, the semiconductor market has also continued to grow rapidly. In addition, the emergence of 5G, artificial intelligence (AI), and the Internet of Things (IoT) have contributed to further driving demand in the semiconductor market. High-performance integrated circuit (IC) chips are currently used in many industries that are receiving a considerable amount of attention, such as AI, IoT, autonomous vehicles, smartphones, and smart devices. Accordingly, the demand for high-performance IC chips has significantly increased. To meet this demand, the modern semiconductor manufacturing industry has developed to include mass production systems that entail numerous processes. Since the productivity of semiconductors derives from the yield of the involved wafers, various studies have been conducted to increase this yield [1–3]. Since the manufacturing processes for semiconductor ICs are sequential, minor anomalies in a given process can lead to potential yield degradation in the subsequent process. Therefore, it is important to effectively control the manufacturing process to achieve high yields [4]. In particular, for semiconductors, the time from design to production is relatively long; thus, if errors cause delays, significant losses accumulate. Therefore,

quickly identifying the root causes of process anomalies and solving problems early are important tasks for achieving a high yield in semiconductor manufacturing. However, as semiconductor ICs continue to be miniaturized and design complexity increases, many defects continue to occur in the semiconductor manufacturing process [5].

Various methods are being studied to determine the root cause of such errors so that they can be corrected quickly. One such method uses the wafers generated during the overall manufacturing process. Prior to performing the packaging process, various electrical tests are performed on a generated wafer to ensure the function of the semiconductor product. As a result, a wafer bin map (WBM) can be obtained from the test result. A wafer is composed of a large number of dies, and a WBM refers to a wafer map that records spatial information of good and failed chips on a wafer, as determined through probe testing. In semiconductor manufacturing, WBMs are used to visualize defect patterns and identify potential process anomalies. Failed dies on wafers tend to form clusters [6], and the type of defect pattern that appears on the wafer provides process engineers with indirect information regarding the root causes of errors in the manufacturing process. Figure 1 shows examples of the various potential defect patterns of a wafer map. Here, failed dies are shown in yellow, good dies are shown in green, and non-wafer areas are shown in purple. As can be seen in Figure 1, the defect patterns in a WBM can be visually classified according to the spatial shape in which defect clusters are located on the wafer. These classified defect patterns provide information related to issues encountered in the semiconductor manufacturing process based on the defect type [7]. For example, 'center' patterns can occur when there is a problem with wafer thickness uniformity during the chemical–mechanical planarization process; an 'edge ring' pattern can occur because of a problem during the etching process; an 'edge location' pattern can occur when there is a problem in the thin film deposition process; and 'scratch' patterns can occur due to improper handling of the wafer [8,9]. Engineers can infer how defect patterns arise based on the visual information obtained through WBMs and their domain knowledge. Based on this inference, an abnormal process can be corrected to improve the yield. Therefore, accurately and efficiently classifying defect patterns is an effective means for identifying the root cause of a defect and improving both the yield and process reliability. However, most companies rely on the eyes and experience of skilled engineers to identify specific WBM defect patterns. This is highly labor-intensive and time-consuming work. In addition, since WBMs are classified according to specific defect patterns in keeping with the domain knowledge and judgment of each engineer, there is a problem in that the classification standard is not constant. To address this problem, several studies have been conducted to classify defect patterns of WBMs through automated processes [10].

In a recent study, several researchers applied various machine learning (ML)- and deep learning (DL)-based [11] approaches to WBM analysis to improve the performance of the involved automated processes. To achieve a high automation performance with ML- and DL-based approaches, a sufficient number of WBMs for each defect pattern is required to classify each defect pattern. However, in the semiconductor manufacturing process, the incidence rates for each defect pattern differ. In addition, even if classified as belonging to the same defect pattern, the spatial location, shape, and size of the clustered defects on a given wafer may vary depending on the WBM in which the defect occurs. A new WBM that has not been used for classifier training can degrade the classification accuracy of the classifier, which is one of the reasons it is difficult to improve the performance of WBM defect pattern classifiers through ML and DL techniques. Therefore, achieving a high WBM classification accuracy requires an appropriate WBM augmentation method to solve the data imbalance problem as well as an index to evaluate it.



(**a**) Center (**b**) Donut (**c**) Edge-Location

(**d**) Edge-Ring (**e**) Location (**f**) Near-Full

(**g**) Random (**h**) Scratch (**i**) None

Figure 1. Examples of WBM defect patterns.

In this paper, we discuss an appropriate augmentation method for improving WBMs' defect pattern classification performance when the data imbalance in the available WBMs data is extreme. To confirm the effectiveness of the proposed method, we used two augmentation models—a deep convolutional generative adversarial network (DCGAN)-based model and a convolutional autoencoder (CAE)-based model—which were trained using real-world data generated during the semiconductor manufacturing process. We performed comparative experiments to assess the classification accuracy and effectiveness of each augmentation method. The classification accuracy according to each defect pattern was calculated from a convolutional neural network (CNN)-based defect pattern classifier. The experimental results showed that the DCGAN-based augmentation method can provide an improved WBM classifier performance compared to the CAE-based augmentation method when augmenting a very small amount of WBM defect data. In the present work, we propose a new index with which to quantify the performance of augmentation methods and evaluate the effectiveness of augmentation methods using this index.

This paper is organized as follows. In Section 2, we present a brief overview of the related works. In Section 3, we introduce the proposed method in detail. In Section 4, we describe the experimental setup and results, and conduct an analysis. In Section 5, we discuss the experimental results. Finally, we conclude this work with a look towards future work in Section 6.

2. Related Works

2.1. Deep Learning

Deep learning is being widely used in various fields, not just in the industrial sector, to solve difficult problems that were previously considered unsolvable. With its ability to analyze and learn from vast amounts of data, deep learning has proven to be effective in fields such as healthcare, finance, and even entertainment [12]. It has been used to diagnose diseases, predict market trends, and create personalized recommendations for users. The versatility and potential of deep learning make it a valuable tool for solving complex problems and advancing various industries. Alabdulkreem, Eatedal et al. proposed a Metaheuristic with Weighted-Fusion-Based Insider Data Leakage Detection and Classification (MWF-IDLDC) model, which applies three DL techniques including LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), and SAE (Stacked Autoencoder) for ubiquitous computing systems [13]. Yang Yu et al. applied a 2D CNN-based model to predict the torsional strength of reinforced concrete (RC) beams. They showed that their proposed 2D CNN model performed better than other machine learning models [14]. Additionally, in [15], they proposed a novel hybrid framework of optimized deep learning models combined with multi-sensor fusion.

2.2. Wafer Map Defect Pattern Classification

To effectively increase productivity, various studies have been conducted to minimize unexpected process abnormalities that occur during the manufacturing process [16–18]. In particular, as mentioned in Section 1, the spatial defect pattern shown on the wafer map provides information to help identify errors in the related processes. Therefore, several studies have been conducted to improve yield by identifying the root causes of the related processes through the spatial patterns shown in WBMs. J. Blue et al. proposed a spatial analysis spectrum method by which to characterize spatial variations in wafers without prior knowledge of wafer patterns [19]. Youngji Yoo et al. proposed an analytical methodology that predicts the results of package testing from the properties obtained by analyzing the defect patterns of wafer maps at the chip level [20].

In recent studies, ML and DL approaches to analysis have been actively examined. Cheng Hao Jin et al. proposed a WBM defect pattern detection method that can simultaneously detect single defect patterns and mixed defect patterns of WBMs using the density-based spatial clustering of applications with noise (DBSCAN) [8,21]. DL-based classifiers using CNN methods [22] in image classification tasks have exhibited particularly high performance. Since CNNs achieved state-of-the-art performance in image classification in 2012, DL has shown excellent performance in image classification and computer vision to date [23,24]. Yang et al. experimented with WBM pattern classifiers through various ML structures (SVM/AdaBoost/XGBoost), including CNNs. They showed that the proposed CNN model performed better in the WBM classification task than other ML-based classifiers [25]. As such, various methods have recently been studied to create better-performing WBM pattern classifiers using various ML/DL approaches. However, achieving a high-performance defect pattern classification accuracy with ML/DL approaches typically involves a large amount of defect pattern data.

2.3. Data Augmentation

In the manufacturing industry, since the number of normal products far exceeds the number of defective products, it is difficult to secure a large quantity of defective data. If the number of defect patterns used in the model learning is significantly low, the model cannot

sufficiently learn features to accurately classify these defects. This reduces the classification accuracy of the model. Additionally, the number of each defect pattern collected through industrial data is imbalanced because the frequency of occurrence for each defect pattern varies depending on the abnormality occurring in the process. To address this problem, various data augmentation methods are used to improve the performance of classifiers. Data augmentation is a method used to increase the size of the training set to be used for classifier learning by artificially generating a training sample. The generated samples should be as close to authentic as possible, and ideally should not allow us to determine whether they were artificially generated or not. Classical image data augmentation methods include resizing, rotating, or inverting the training data, as well as changing the brightness of, cropping, or enlarging the image. This is an effective data augmentation method for learning several image classifiers. However, some of these methods are unsuitable for use in classifying wafer map defect patterns because these methods destroy the characteristics that distinguish the defect patterns present on WBMs. For example, when the middle region of a WBM with a 'center' defect pattern is cropped and used for model learning, the classifier fails to fully learn the important features that distinguish 'none' from 'center.' As such, training the classifier by augmenting the data in a way that is not suitable for the characteristics of the data degrades the performance of the model. Therefore, in order to improve the performance of the classifier, it is important to select an appropriate data augmentation method. The following two strategies are widely known as methods for improving the classification performance of such imbalanced data.

(1) Transfer learning [26]: Transfer learning is not a data augmentation method, but it is a way to improve the classification performance of imbalanced data. Transfer learning is a method that can be generalized to classify a small number of data using a pre-trained model with a large amount of data.

Zongli Shen et al. showed that in the case of a wafer map with a significant class imbalance problem, the transfer learning structure proposed in their paper identified defects better than a general CNN structure [27]. However, the disadvantage of transfer learning is that it requires a lot of effort to tune the hyperparameters to fit the model being used.

(2) Sampling techniques: Another way to improve the classification performance is to use sampling techniques to address data imbalance. Two types of sampling methods are typically used. The first is the under-sampling method of reducing the number of relatively large classes compared to other classes to evenly match the overall ratio, and the other is the over-sampling method of increasing the data of insufficient classes compared to other classes. Under-sampling methods can be used to balance classes, but insufficient samples per class can result in the loss of useful characteristics for classifiers in learning class diversity. It could be alleviated by means of over-sampling methods. In the context of classifier training, it is crucial to consider not only the quantity but also the quality and diversity of data to prevent classifiers from being biased towards certain areas of the image. A classifier trained on a small amount of data may exhibit overfitting, causing it to be overly biased towards the given training data. Similarly, a large amount of training data alone does not guarantee generalization if the diversity of the data is lacking. Hence, it is essential to ensure the quality and diversity of data when securing data for classifier training. Therefore, the over-sampling method is needed to increase the number of samples in an appropriate manner. Furthermore, appropriate data augmentation methods should be applied to over-sampling methods to prevent overfitting.

Takeshi Nakazawa et al. created a synthetic wafer map with a specific defect pattern. The authors showed that a CNN model trained only on synthetic wafer maps for network training could classify real wafer maps with a high accuracy. The authors showed that the network was not significantly overfitted by using real wafer map images as a validation data set [28]. Shon et al. proposed automatic techniques using a policy search strategy to reduce the human effort required to augment WBM data. They considered rotation, flipping, shifting, shearing range, and zooming as data augmentation methods to effec-

tively solve data imbalance problems [29]. Kang proposed a rotation-based augmentation method as a way to mitigate the decrease in the accuracy of the classifier due to the lack of WBM defect patterns [30]. The author augmented wafers by seven times compared to the original wafer by rotating it by 90 degrees and flipping the rotated wafer horizontally. From the experimental results, the author argued that CNN models that learn augmented data through rotational augmentation methods can achieve a higher classification performance by providing consistent predictions of rotational variations. YongSung Ji and Jee-Hyong Lee proposed a DCGAN-based augmentation model to generate imbalanced WBM defect patterns. The authors showed that a classifier that has learned wafers generated from a DCGAN-based augmentation model exhibited a higher classification performance than a classifier that has learned wafers generated from rotation-based augmentation techniques [31]. As such, several studies have been conducted on data augmentation methods to improve the classification performance of WBM defect patterns. In addition, because it is difficult to collect abundant industry data, predicting the appropriate sample size of the training data required for training a classifier is the most common limitation to consider in reality. Addisson Salazar et al. proposed a 'proxy approach' to measure the degree of convergence of other functions related to the final error probabilities of the classifiers to predict the optimal learning sample size required for learning the classifiers [32]. Instead of calculating the actual error probability of the classifier according to the training set size, they calculated a proxy curve represented only by the training set size and a function represented by feature space dimension to obtain the optimal training sample size. However, the results of studies on numerical analysis related to the efficiency of the methods applied to solve the data imbalance problem remain unclear.

2.4. Generative Adversarial Networks

Generative adversarial networks (GANs) [33] are widely used as a generative model. GANs are deep-learning algorithms of unsupervised learning. A GAN learns the features of the input data to generate new data. Fundamentally, a GAN consists of two types of networks: generator networks denoted by $D(x)$, and discriminator networks denoted by $G(z)$. The generator receives random noise as an input value and generates data through the generator network. The discriminator network determines whether the real data and generated data from the generator network are 'real' or 'fake'. Next, the generator network updates the network so that the generated data are determined to be 'real' from the discriminator network. Through this process, the GAN can generate more realistic images. In [28], the relationship between $G(z)$ and $D(x)$ is defined by the two-player-minimum game and expressed by the following:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)} \log[1 - D(G(z))] \quad (1)$$

where $x \sim p_{data}(x)$ is sample x from the real distribution, $z \sim p_z(z)$ is the sample latent vector z from the Gaussian distribution, $0 \leq D(x) \leq 1$, and $0 \leq G(z) \leq 1$.

Since the GAN was first introduced by Ian Goodfellow in 2014, various studies have been conducted to improve its performance, with the DCGAN in [34] being one. To better learn the features of images, the DCGAN adds a convolutional layer to the conventional GAN. In the present work, we constructed a DCGAN-based WBM defect pattern augmentation model to address data imbalances.

3. Proposed Methodology

3.1. Overview

In this paper, we propose a DCGAN-based WBM augmentation model as a data augmentation method for improving the performance of WBM defect pattern classifiers on extremely imbalanced WBM datasets. Figure 2 shows a schematic of the proposed methodology. First, (a) the WBMs to be used for augmentation for each defect pattern are selected. Here, the number of WBMs for each defect pattern selected for augmentation was

limited to not exceed 20% of the corresponding defect pattern to maintain the imbalance between each defect pattern and the deficiency of the number of defects. Next, (b) a DCGAN-based WBM augmentation model is trained using the selected WBM. Then, (c) the wafer augmented with a DCGAN-based model is refined using a masking process. This step corrects errors in the augmented WBM without manual intervention and improves the quality of the generated WBM. After this, (d) the refined WBM is used to train a CNN-based WBM defect classifier. We constructed a CNN-based WBM defect pattern classifier to determine the quality of WBMs generated by CAE-based augmented methods and DCGAN-based augmented methods, comparing the two types of methods based on the accuracy of the model. Additionally, we propose an index for numerically estimating the polymorphism of generative models and compare the performance of generative models using the index, which we named the polymorphic generative index (PGI).

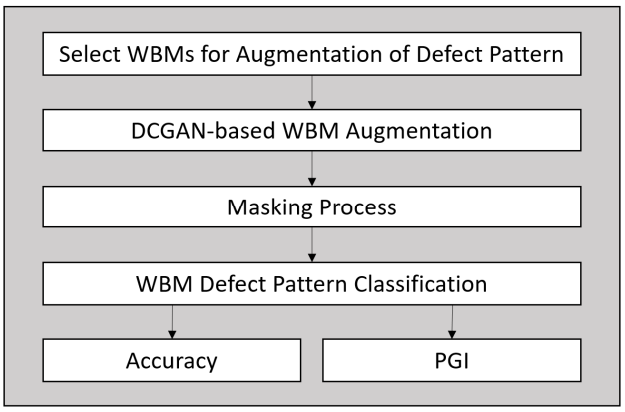


Figure 2. Schematic of the proposed methodology.

For defect pattern recognition and classification of the experiments, we defined each WBM via classification as one of the nine defect patterns represented in Figure 1. We used the WM-811k dataset [35] for our experiments. The WM-811k dataset was used to train and test the WBM augmentation model and the CNN-based WBM defect pattern classifier.

3.2. Dataset and Pre-Processing

The WM-811k dataset includes the largest quantity of publicly available WBM data. The WM-811k consists of approximately 811,000 WBMs produced by the semiconductor industry. In the WM-811k dataset, labeled wafers account for 21.3% of the total data, or 172,950 wafers, and among the labeled wafers, those labeled with defect patterns account for 3.1% of the total, or 25,519 wafers. When labeling, the WM-811k dataset assumed that one wafer map had one defect pattern. Here, the label consisted of a total of nine defect patterns, including the 'none' pattern (i.e., no defect on the wafer).

To reliably train defect pattern classifiers and generative models, dependable and clearly labeled data are required. As mentioned above, the WM-811k dataset has a significant amount of unlabeled data, and the wafers that make up this dataset are of 632 different sizes. Therefore, it is not suitable to use the WM-811k dataset directly for experiments, and appropriate preprocessing steps must be performed.

We preprocessed the WM-811k dataset in the following order. First, we selected all (26, 26)-sized WBMs. Since the WBMs of wafer size (26, 26) had all been labeled by domain experts, this was an effective way to obtain WBMs that were well refined and uniform without performing unnecessary work for data preprocessing. Table 1 shows the number of WBMs of (26, 26) size in the WM-811k dataset. As shown in Table 1, most defects were classified as 'none', and the frequency of the occurrence of each defect pattern was significantly less than that of the 'none' pattern. In addition, the frequency of occurrence for each defect pattern was extremely different. We used two different data augmentation methods to address these data imbalance problems and compared the performances of

each method. Figure 3 shows the proportion of each defect pattern, excluding the 'none' pattern, among the (26, 26)-sized WBMs selected for experiment in the WM-811k dataset.

Table 1. The number of each defect type in the WM-811k dataset wherein the wafer size is 26 × 26.

Defect Type	Number of Samples
Center	90
Donut	1
Edge-Location	296
Edge-Ring	31
Location	290
Near-Full	16
Random	74
Scratch	72
None	13,489

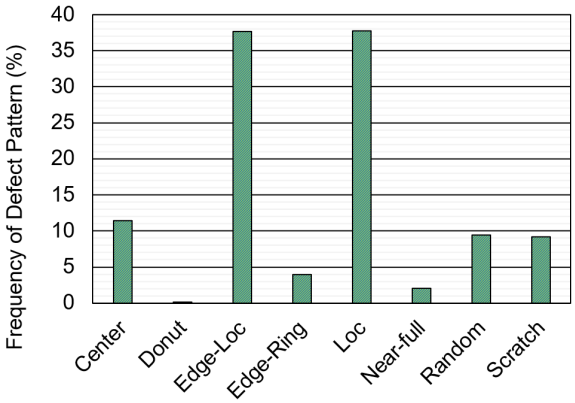


Figure 3. Defect pattern frequency of WM-811k at size 26 × 26, excluding 'none' pattern.

Next, the selected (26, 26) WBMs were resized to (64, 64). This preprocessing procedure helps the WBMs generated through the generative model to include more diverse types of defect bits compared to the case wherein the 26 × 26-sized WBM is used for generative model learning. After resizing, we mapped the 'good' die, 'failed' die, and 'non-wafer areas' to different channels. In the WM-811k dataset, a WBM is composed of a combination of numbers 0, 1, and 2. We reconstructed the original dataset of (26, 26) shape into a (64, 64, 3) shape, specifying that only one of the same coordinates of each reconstructed channel had a value of 1, while the rest had a value of 0. In other words, the (n, m) coordinates of the reconstructed WBM had one of the values of [1, 0, 0], [0, 1, 0], or [0, 0, 1], where $0 \leq n \leq 63$, $0 \leq m \leq 63$.

As shown in Figure 3, there were extreme differences between the frequencies of the occurrence of defect patterns. This data imbalance is common in industry data. To prove the effectiveness of the proposed generative model on extremely imbalanced data, we selected only a part of the original data and used it for augmentation model learning; the remaining original WMBs were not used for learning. At this time, the data were selected so that the imbalanced nature of the original data was maintained. In Figure 4, the ratio of the number of the WBMs selected for augmented model learning to the number of original WBMs with the (26, 26) shape in the WM-811K dataset is shown for each defect pattern. As can be observed in Figure 4, we selected WBMs at various ratios from a minimum of 2.7% to a maximum of 18.75% for each defect pattern, except for the 'donut' pattern. As shown in Table 1, there was only one 'donut' WBM in the WM-811K dataset with a 26 × 26 shape. Since there was only one original WBM for this pattern, the accuracy of the classifier may indicate how similar the WBM generated using the defect pattern was to the original WBM, but not how much diversity the generative model can augment for that defect pattern. In this paper, we compare the classification accuracy of the generated WBM and the original

WBM using two types of augmentation models, respectively. We also cross-validate the classification accuracy of WBMs generated via two different augmentation methods to numerically compare the efficiency of each augmentation model. In addition, we quantify the augmented effects of both augmentation models through the PGI.

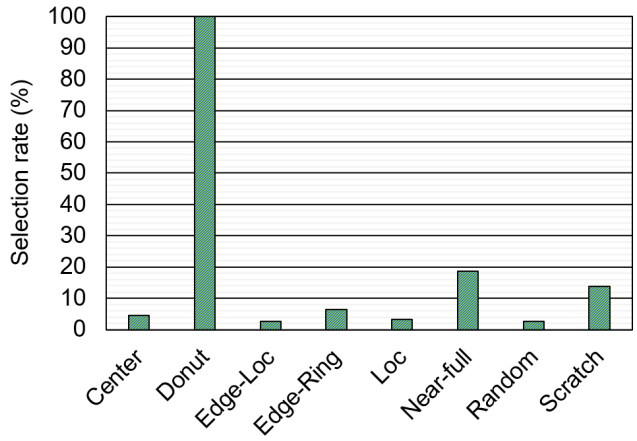


Figure 4. Percentage of WBMs selected for augmentation among the specified WBMs for each defect pattern.

3.3. DCGAN-Based Generative Model

Figure 5 shows the architecture of the proposed DCGAN. As described in Section 2.4, DCGAN consists of two types of networks: a generator network and a discriminator network. Tables 2 and 3 show the detailed structures of the generator network and the discriminator network of the DCGAN-based model to augment each defect pattern, respectively.

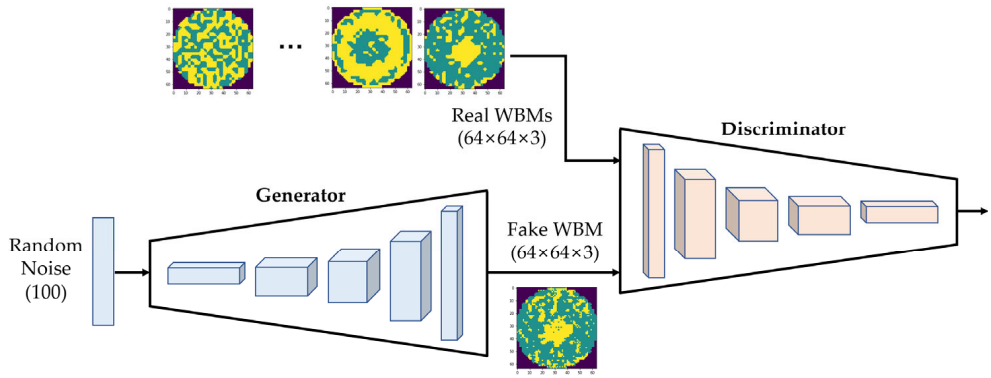


Figure 5. The architecture of the proposed DCGAN.

Table 2. Generator network architecture and hyperparameters.

Layer	Kernel Size, Stride	No. of Parameters	Output Shape
Input	-	-	(100)
Dense	-	6,553,600	(16, 16, 256)
Reshape	-	-	(16, 16, 256)
Convolutional	5 × 5 × 128, 1	819,200	(16, 16, 128)
Batch Normalization	-	512	(16, 16, 128)
Convolutional	1 × 1 × 128, 1	16,384	(16, 16, 128)
Batch Normalization	-	512	(16, 16, 128)
Convolutional	5 × 5 × 64, 1	204,800	(32, 32, 64)
Batch Normalization	-	256	(32, 32, 64)
Convolutional	5 × 5 × 3, 1	4800	(64, 64, 3)

Table 3. Discriminator network architecture and hyperparameters.

Layer	Kernel Size, Stride	No. of Parameters	Output Shape
Input	-	-	(64, 64, 3)
Convolutional	$5 \times 5 \times 64$, 2	4864	(32, 32, 64)
Dropout (0.4)	-	-	(32, 32, 64)
Convolutional	$5 \times 5 \times 128$, 2	204,928	(16, 16, 128)
Dropout (0.4)	-	-	(16, 16, 128)
Flatten	-	-	(32,768)
Dense	-	32,769	(1)

The generator network configured as shown in Table 2 received 100 noise vectors following Gaussian distribution and generated a WBM image of (64, 64, 3) shape. The activation function of the generator network was scaled as the exponential linear unit (SELU), and the tanh function was used as the activation function of the last convolutional layer. The generator network consisted of a total of three convolutional layers. Using a 1×1 convolutional layer on CNN was shown to be effective in reducing the dimensionality of the model, reducing computational requirements, enhancing nonlinearity, and preventing overfitting [36,37].

The WBM of (64, 64, 3) shape augmented from the generator network was used to train the discriminator network, and the discriminator network was trained to discriminate the authenticity of the images generated by the generator network. The discriminator network was composed of two 5×5 convolutional layers and two dropout layers. The leaky rectified linear unit (ReLU) ($\alpha = 0.2$) was used as the activation function of the discriminator network, and the sigmoid function was used as the activation function for the dense layer.

3.4. Masking

In order for the augmentation method to be applied in the industry, stable augmentation performance must be guaranteed. However, it is already widely known that GAN-based models are difficult to train and easily become unstable due to issues such as mode collapse, Nash equilibrium, and non-convergence. These problems degrade the quality of the augmented WBMs and cannot guarantee the quality of the generated WBMs. Moreover, there was a problem with broken edges in some wafers in the WM-811k dataset. Augmented WBMs using imperfect WBMs, such as broken-edge WBMs, are a factor that can impact the overall quality of the classification model's reliability. To alleviate the imperfections of the augmented WBM and ensure the reliability of the generative model, it is effective to properly calibrate the augmented wafer. In [38], the authors improved the quality of augmented wafers through wafer repair when the shape of the wafer was deformed, or when a good bit or failed bit was in the wrong location on the wafer map (e.g., outside the wafer area).

As mentioned above, GAN-based augmentation model learning can be unstable, which can sometimes lead to the production of low-quality augmented images. It is labor-intensive to manually remove these low-quality augmented images, and even if such manual work is performed it is difficult to solve the problem of deviation in the quality of the augmented WBMs due to problems caused by worker bias. Therefore, we refined the augmented WBMs through a defined process. We performed the masking process on the augmented WBMs to correct the errors in the WBMs. The masking process was performed with the following steps. First, assuming that the size of the generated WBM was $2r \times 2r$, a circle with a radius of r was formed. The circle formed was called the masking wafer. The inner region of this circle was assumed to be the inner area of the original wafer. After overlapping the masking wafer and the generated WBM, the good bits and the failed bits which formed outside the wafer of the generated WBMs were removed. The removed bits were marked as areas outside the wafer. Figure 6 shows the process of refining the WBMs via the masking process (Figure 6a is an example of a WBM augmented with poor quality by an augmentation model). Then, the masking wafers on the augmented

WBMs were covered, as shown in Figure 6b,c. The good/failed bits were removed from the areas outside the wafer. The masking process had the advantage of being able to check whether a defect pattern occurred at the incorrect spatial position on a given wafer. A WBM augmented through the DCGAN-based model tended to be augmented by focusing on the features constituting the defect pattern on the wafer rather than the absolute spatial shape of the wafer; thus, the boundary between the wafer itself and the outside of the wafer may appear ambiguous. The masking process thus ensures that the generated WBM maintains the shape of the original wafer. It is possible to ensure the quality of the augmented wafer while reducing the labor (human power) required for manual operation by an engineer.

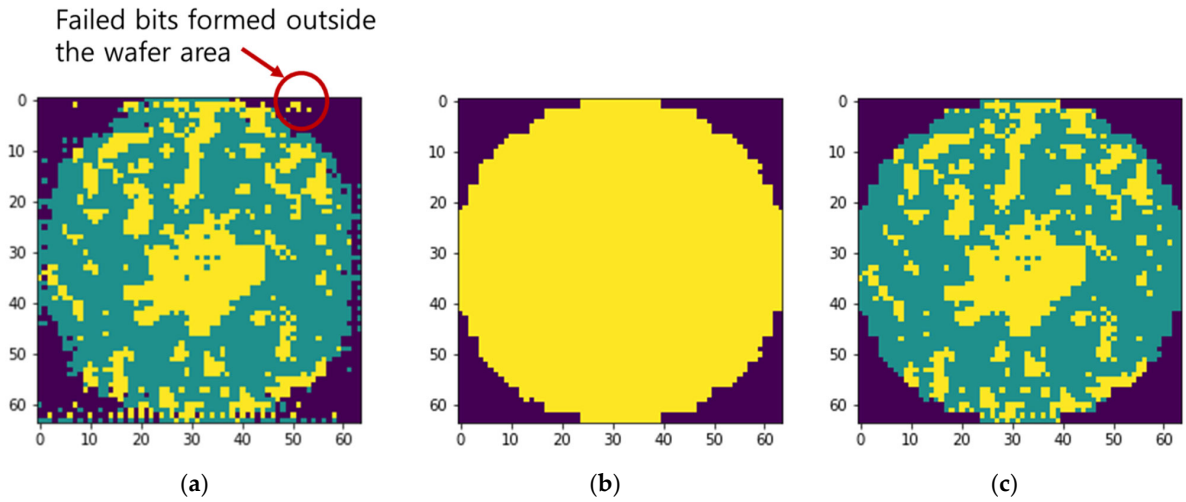


Figure 6. Masking process for wafer refining: (**a**) augmented WBM with poor quality; (**b**) masking wafer; (**c**) refined augmented WBM.

3.5. Prediction Model

In this study, to verify the performance according to the generated WBMs, a CNN-based classification model was used to test the classification accuracy according to each defect pattern. As noted in Section 2, the CNN is effective in classifying WBM defect patterns. We constructed a deep CNN model with three convolutional layers, three max-pooling layers, and two fully connected layers for the experiment. Table 4 shows the CNN structure and hyperparameters constructed for the experiments. In the CNN structure, all convolutional layers used the 'same' padding and the ReLU function as an activation function. ReLU was used as the activation function of the first fully connected layer, and the softmax function was used as the activation function of the second fully connected layer.

Table 4. CNN architecture and hyperparameters.

Layer	Kernel Size, Stride	No. of Parameters	Output Shape
Input	-	-	(64, 64, 3)
Convolutional	3 × 3 × 32, 1	896	(64, 64, 32)
Max-Pooling	2 × 2, 2	-	(32, 32, 32)
Convolutional	3 × 3 × 64, 1	18,496	(32, 32, 64)
Max-Pooling	2 × 2, 2	-	(16, 16, 64)
Convolutional	3 × 3 × 64, 1	36,928	(16, 16, 64)
Max-Pooling	2 × 2, 2	-	(8, 8, 64)
Flatten	-	-	(4096)
Fully Connected	-	1,048,832	(256)
Fully Connected	-	2313	(9)

3.6. Polymorphic Generative Index

To evaluate the effectiveness of the proposed generative model, a quantitative index is needed to evaluate how diversely the model augments the data. Although many studies

have been conducted to evaluate the performance of the generative model, agreed-upon criteria for evaluating the GAN model have not yet been clearly established. The most common and intuitive method for evaluating the performance of GAN is manual evaluation, which entails a visual evaluation by human judges, such as via the use of Rating and Preference Judgment and the use of Rapid Scene Categorization. However, this qualitative method is subjective because the indicators vary based on the evaluator's stereotypes or biases, and its application is limited because it is greatly influenced by the evaluator's domain knowledge. In addition, it is difficult to verify the problem of GAN methods, such as mode collapse. Therefore, to solve this problem, the need for quantitative evaluation indicators has emerged.

In general, GANs are evaluated in terms of 'quality', or how similar the generated images are to the original images, and 'diversity', i.e., whether they can generate a variety of images across different kinds of noise vectors. The representative metrics used to evaluate GAN are IS (inception score) [39] and FID (Fréchet inception distance). In general, IS is measured using the ImageNet pre-trained model, Inception v3, and the calculation of IS is performed as follows:

$$\mathrm{IS} = \exp\left(\mathbb{E}_{x \sim p_{data}} D_{KL}(P(y \mid x) \parallel P(y))\right) \tag{2}$$

where $x \sim p_{data}$ means that x is an image sampled from p_{data}; $D_{KL}(P(y \mid x) \parallel P(y))$ is the Kullback–Leibler divergence between the distributions $P(y \mid x)$ and $P(y)$; $P(y \mid x)$ is the conditional class distribution; and $P(y) = \int_x P(y \mid x) P_{data}(x) dx$ is the marginal class distribution for noise vectors z. In the present work, IS was calculated for each class according to the preceding equation, with a large IS value indicating that the generation model had a better generation performance.

However, IS has the following disadvantages: large IS values are achieved even if the augmentation model repeatedly generates one good quality identical image for each class of the classifier. This means that IS cannot measure inner diversity within a class of generative models. In other words, IS is vulnerable to mode collapse evaluation. Furthermore, high IS values are achieved even when a GAN replicates training data exactly. When the augmentation model generates only one image per class, $P(y)$ comes close to a uniform distribution, even if the diversity is low. This causes IS to exhibit a distorted performance. Additionally, unlike human cognition, deep learning networks consider style to be more important than content. That is, image classification networks consider texture more than shape in terms of input data [40,41]. Thus, even if the augmented model outputs results that differ from human thinking, IS can achieve high numbers. Therefore, sometimes IS may not be suitable for evaluating the performance of the augmentation model. Furthermore, IS uses the Inception v3 model, a pre-trained model based on the ILSVRC 2012 dataset, which implies that it is best suited to the evaluation of the 1000 object types used in the ILSVRC 2012. This means that IS is not suitable for the evaluation of all aspects of interest to us.

IS evaluates performance using only the distribution of the generated images, while the FID compares the distribution of generated images with the distribution of a set of real images of the target domain. The FID uses Inception v3 as a feature extractor to extract the activation map between the real image and the generated image. Then, the FID is obtained by measuring d^2, which is the two-Wasserstein distance between the probability distribution of the real image r and the probability distribution of the generated image g. The FID is calculated as follows:

$$\mathrm{FID}_{(r,g)} = \mathrm{d}^2 = \|\mu_r - \mu_g\|_2^2 - Tr\left(\Sigma_r + \Sigma_g - 2\Sigma_r\Sigma_g\right) \tag{3}$$

where μ_r, μ_g, and Σ_r, Σ_g represent the mean and covariance matrices of feature vectors' output by the real images of the target domain and the generated images via the Inception v3 model, respectively. Tr is the sum of the diagonal component of the matrix.

The FID obtained by measuring the distance between the activation distributions of the two images has a lower value, as similar images are obtained. The FID entails that the

distance between sets of images of the same class is closer than that of other images of the class, and the distance between completely different sets of images is further than that of other images of the class. Therefore, a generated image with a low FID can be considered similar in probability distribution to the real image, which can be evaluated as generating a high-quality image given that the FID of the augmentation model is measured lower. Furthermore, the FID is able to evaluate the quality of the generated image more robustly than IS because the distance value is higher if the model generates only one image per class. For example, the generated image is more sensitive than IS when it contains noises such as blurring and mosaic compared to the original image.

However, the FID has the following disadvantages. (A) ILSVRC 2012-based pre-trained models such as Inception v3 are required. As noted in the description of IS, ILSVRC 2012-based pre-trained models may not be suitable for specific domains such as WBM classification; (B) it is recommended that FID is calculated using a minimum sample size of 10,000, otherwise, the true FID of the augmentation model may be underestimated [42]. As such, it is difficult to obtain the 10,000 sample data entries recommended for FID evaluation when the deviation between normal and abnormal data is large and collection is difficult, as it is for industrial data. In this paper, we propose PGI as a new index to numerically verify the effectiveness of an augmentation model. The PGI is obtained by comparing the number of original data used in augmentation model learning to the number of original data for each class correctly inferred through the preceding classifier; the corresponding equation is as follows:

$$\text{PGI} = \frac{N_{Correct}}{N_{Augment}} \tag{4}$$

where $N_{Correct}$ is the number of original wafers correctly classified through the classifier, $N_{Augment}$ is the number of original wafers used to augment the data, and $0 \leq PGI \leq N_{Correct}$.

The PGI is obtained for each defect pattern. The obtained PGI can be expressed as PGI_{Center}, PGI_{Donut}, PGI_{Loc} for each defect pattern. In the present work, the PGI was used to numerically verify how many various defect patterns can be formed with respect to the original defect pattern per WBM used for data augmentation. We estimated the quantitative effect of the augmentation model by calculating the PGI. The closer the PGI was to $N_{Correct}$, the higher the performance of the augmentation model for the class; the closer the PGI was to 0, the lower the performance of the augmentation model.

4. Experimental Results

In this section, the DCGAN-based augmented WBMs and CAE-based augmented WBMs were found through the CNN described in Section 3.5, and then the classification accuracy and PGI of each model were used to compare and analyze the performance of the generative model.

Our experimental environment was Windows OS (Redmond, Microsoft, WA, USA) with an Intel i9-10920 × 3.60 GHz CPU, GeForce RTX 3060 GPU, and 64 GB of memory. The models were trained on Keras 2.5.1 (with Tensorflow 2.5) using Python 3.8.5.

4.1. CAE-Based Generative Model

As described in Section 2, there have been several studies that have applied CAE-based augmentation methods to resolve the data imbalance in WBMs. CAE is used as an effective unsupervised learning method that can be applied even when there is an imbalance between classes in the training data [43].

Table 5 shows the structure of the CAE-based augmentation model used in the experiment to compare the performance of the DCGAN-based augmentation model. Data augmentation using a CAE is performed by adding noise to the latent vector generated as a result of passing the CAE encoder learned through the original data to generate data similar to the WBM used for learning. After the training of the CAE-based model is completed, the encoder and decoder generate a WBM that is similar to the input by passing it through the network. We used ReLU for the activation function of all layers except for the activation

function of the output layer of the decoder; we used mean squared error (MSE) as the cost function, and Adam as the optimizer. Sigmoid was used for the activation function of the output layer of the decoder.

Table 5. Convolutional autoencoder architecture and hyperparameters.

Part	Layer	Kernel Size, Stride	No. of Parameters	Output Shape
Encoder	Input	-	-	(64, 64, 3)
	Convolutional	$3 \times 3 \times 64$, 1	1792	(64, 64, 64)
	Max-Pooling	2×2, 2	-	(32, 32, 64)
	Convolutional	$3 \times 3 \times 128$, 1	73,856	(32, 32, 128)
	Max-Pooling	2×2, 2	-	(16, 16, 128)
	Convolutional	$3 \times 3 \times 256$, 1	295,168	(16, 16, 256)
	Max-Pooling	2×2, 2	-	(8, 8, 256)
Decoder	Conv. Transpose	$3 \times 3 \times 256$, 1	590,080	(8, 8, 256)
	Up-Sampling	-	-	(16, 16, 256)
	Conv. Transpose	$3 \times 3 \times 128$, 1	295,040	(16, 16, 128)
	Up-Sampling	-	-	(32, 32, 128)
	Conv. Transpose	$3 \times 3 \times 64$, 1	73,792	(32, 32, 64)
	Up-Sampling			(64, 64, 64)
	Conv. Transpose	$3 \times 3 \times 3$, 1	1731	(64, 64, 3)

$\sigma = 0.2$ was used for the noise added to the latent vector. Here, σ means a value to be multiplied by a random value that follows the normal distribution added to the latent vector generated through the encoder of the CAE so that the CAE-based model can form more diverse defect patterns. The shape of the WBM generated using a CAE depends on the value of σ to be multiplied by the Gaussian noise added to the latent vector. When a small value of σ is used, the WBM generated from the CAE simply forms a highly similar shape to that of the original WBM. In this case, there is a concern that the augmented dataset may be biased toward specific defect patterns. On the other hand, if σ is set too high, the generated WBM may show a defect pattern deviating from the pattern characteristics of the original WBM. In this case, the generated WBM may not satisfy the classification criteria of the dataset.

4.2. Augmented Results

Figure 7a–c show examples of WBMs with a 'donut' defect pattern generated through each augmentation model. Figure 7a shows the original WBM used to train the augmentation model, and Figure 7b,c show the output results of each augmentation model trained using the original WBM. As shown in Figure 7b, the CAE-based model produced almost the same results as the original WBM. However, as shown in Figure 7c, the DCGAN-based model created more diverse types of WBMs while maintaining the characteristics of the defect patterns of the original WBM. This applies not only to the 'Donut' defect pattern, but to all defect patterns learned by the DCGAN-based model. Figure 7d displays defect patterns and their labels randomly generated from DCGAN-based models. These results demonstrate that the DCGAN-based model can generate more diverse WBM defect patterns than the CAE-based model, not only for the 'Donut' pattern but also for several other defect patterns. Randomly generated WBM defect patterns from the DCGAN-based model can create WBMs with meaningful variations while maintaining the characteristics of the defect patterns for all WBMs learned during training. Similar to WBMs generated in DCGAN-based models, WBMs generated to have more diverse shapes while maintaining the characteristics of defect patterns alleviate the overfitting of classifiers to specific defect patterns. This can be seen in the experimental results of Section 4.3.1 and Section 4.3.2.

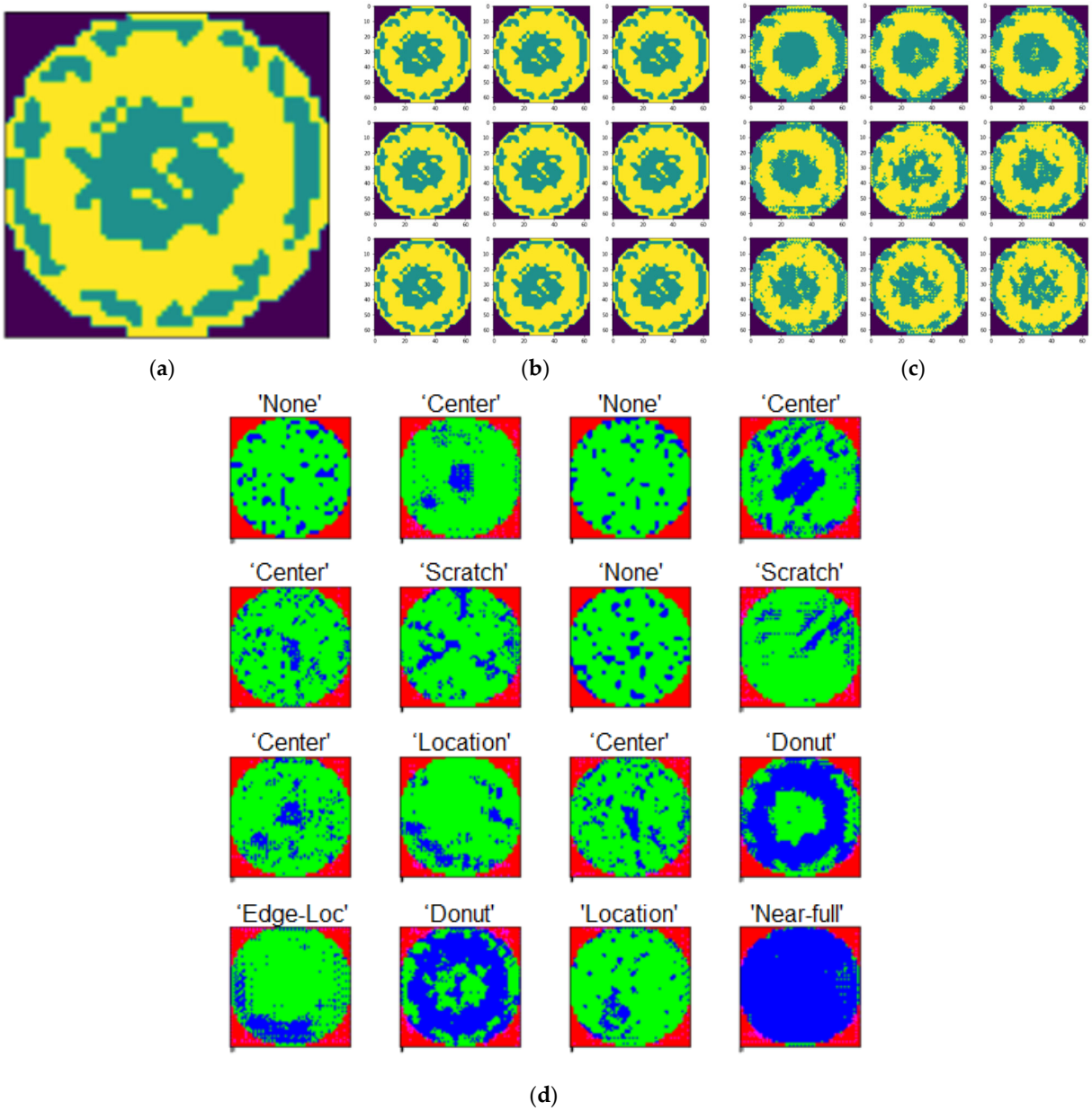


Figure 7. Results generated by each augmented model for the 'donut' defect pattern: (**a**) original wafer ('donut'); (**b**) augmented wafers from the CAE-based model (σ = 0.2); (**c**) augmented wafers from the DCGAN-based model; (**d**) randomly generated defect patterns and their labels from the DCGAN-based models.

4.3. Performance Evaluation

4.3.1. Comparison of Original Data Classification Performance of Models Trained via Augmented Wafers

Tables 6 and 7 show the classification performance of the WBM defect pattern classifier trained through the DCGAN-based model and the WBM defect pattern classifier trained through the CAE-based model, respectively. As can be seen in Tables 6 and 7, the difference in the classification accuracy for each defect pattern of the CNN classifier trained with augmented WBMs in the two augmented models varied from a minimum of 0.57% to a maximum of 10.81%. Furthermore, for all defect patterns, the performance of the CNN classifier trained using augmented data from the DCGAN-based model was higher than that of the classifier trained using augmented data from the CAE-based model.

Table 6. Confusion matrix of classifier trained with augmented data from DCGAN-based model.

Actual \ Predicted	Center	Donut	Edge-Loc	Edge-Ring	Loc	Near-Full	Random	Scratch	None
Center	57 (63.33%)	0	0	2	0	0	0	1	30 (33.33%)
Donut	0	1 (100%)	0	0	0	0	0	0	0 (0%)
Edge-Loc	1	0	108 (36.48%)	11	12	0	0	4	160 (54.05%)
Edge-Ring	0	0	0	20 (64.51%)	0	0	0	0	11 (35.48%)
Loc	21	0	32	0	90 (30.30%)	0	0	50	104 (35.02%)
Near-Full	0	0	0	0	0	16 (100%)	0	0	0 (0%)
Random	2	1	23	0	9	1	35 (47.29%)	3	0 (0%)
Scratch	0	0	0	0	1	0	0	21 (29.16%)	50 (69.44%)
None	7	0	3	0	20	0	0	44	13,415 (99.45%)

Table 7. Confusion matrix of classifier trained with augmented data from CAE-based model.

Actual \ Predicted	Center	Donut	Edge-Loc	Edge-Ring	Loc	Near-Full	Random	Scratch	None
Center	48 (53.33%)	0	0	0	1	0	0	9	32 (35.56%)
Donut	0	1 (100%)	0	0	0	0	0	0	0 (0%)
Edge-Loc	0	0	80 (27.02%)	34	34	0	0	19	129 (43.58%)
Edge-Ring	0	0	4	18 (58.06%)	0	0	0	9	31 (29.03%)
Loc	23	0	13	0	67 (22.55%)	0	2	69	123 (41.41%)
Near-Full	0	0	0	0	0	16 (100%)	0	0	0 (0%)
Random	2	2	26	0	7	2	27 (36.48%)	8	0 (0%)
Scratch	0	0	0	0	2	0	0	19 (26.38%)	51 (70.83%)
None	3	0	11	1	27	0	0	108	13,339 (98.88%)

In evaluating industrial data classifiers, distinguishing abnormal from normal is as important as the classification accuracy of each abnormality. The misclassification rate of all original WBMs was 1.41% lower in classifiers trained with DCGAN-based augmented wafers than in classifiers trained with CAE-based augmented wafers. In addition, the classification accuracy for each defect pattern was approximately 5.31% higher in the classifier trained with DCGAN-based augmented wafers than in the classifier trained with CAE-based augmented wafers. This showed that the WBMs generated by the DCGAN-based model had a probability distribution which was more similar to the original WBMs than the WBMs generated by the CAE-based model.

4.3.2. Comparison of the Performance of Different Augmented Data Classification Models Trained via Augmented Wafers

Table 8 shows the classification accuracy for each defect pattern in the WBMs generated by the CAE-based augmentation model using the classifier trained with the WBMs generated by the DCGAN-based augmentation method. In the same way, Table 9 shows the classification accuracy for each defect pattern in the WBMs generated by the DCGAN-based augmentation model using the classifier trained with the WBMs generated by the CAE-based augmentation method. The experimental results showed that each defect

pattern-specific accuracy of a model trained via a WBM generated by a DCGAN-based model was always higher than each defect pattern-specific accuracy of a model trained via a WBM generated by a CAE-based model, as well as being 14.26% higher on average. Furthermore, the experimental results also showed that WBMs augmented using a CAE-based model were fully classified by defect patterns in a classifier trained with DCGAN-based augmented wafers, while some WBMs augmented with a DCGAN-based model were not accurately classified by a classifier trained through CAE-based models. These results intuitively show that the DCGAN-based model formed diverse defect patterns compared to the CAE-based model.

Table 8. Confusion matrix of the classifier trained with data augmented by DCGAN-based model when using data augmented by CAE-based model as a validation dataset.

Actual \ Predicted	Center	Donut	Edge-Loc	Edge-Ring	Loc	Near-Full	Random	Scratch	None
Center	500 (100%)	0	0	0	0	0	0	0	0 (0%)
Donut	0	500 (100%)	0	0	0	0	0	0	0 (0%)
Edge-Loc	0	0	500 (100%)	0	0	0	0	0	0 (0%)
Edge-Ring	0	0	0	500 (100%)	0	0	0	0	0 (0%)
Loc	0	0	0	0	500 (100%)	0	0	0	0 (0%)
Near-Full	0	0	0	0	0	500 (100%)	0	0	0 (0%)
Random	0	0	0	0	0	0	500 (100%)	0	0 (0%)
Scratch	0	0	0	0	0	0	0	500 (100%)	0 (0%)
None	0	0	0	0	0	0	0	2	498 (99.6%)

Table 9. Confusion matrix of the classifier trained with data augmented by CAE-based model when using data augmented by DCGAN-based model as a validation dataset.

Actual \ Predicted	Center	Donut	Edge-Loc	Edge-Ring	Loc	Near-Full	Random	Scratch	None
Center	481 (100%)	0	0	0	13	0	0	4	2 (0.4%)
Donut	0	483 (96.6%)	0	0	0	17	0	0	0 (0%)
Edge-Loc	0	0	465 (93%)	0	13	6	1	0	15 (3%)
Edge-Ring	4	0	12	103 (20.6%)	2	0	0	22	357 (71.4%)
Loc	1	0	14	0	473 (94.6%)	0	0	0	12 (2.4%)
Near-Full	0	0	0	0	0	500 (100%)	0	0	0 (0%)
Random	0	4	8	0	19	90	369 (73.8%)	10	0 (0%)
Scratch	5	0	0	0	1	0	0	467 (93.4%)	27 (5.4%)
None	0	0	0	0	0	0	0	4	496 (99.2%)

4.3.3. Comparison of PGI

Table 10 shows the PGI for each augmentation method calculated according to the results of Tables 6 and 7 in Section 4.3.1, above. The selection rate in Table 10 refers to the ratio of WBM used for augmentation to the number of original WBMs for each defect pattern. As shown in Table 10, the DCGAN-based model had the same or higher PGI compared to

the case of the CAE-based model for all defect patterns. Notably, there was no correlation between the PGI and the proportion of the original WBMs used for augmentation.

Table 10. Comparison of PGI according to each defect pattern.

Defect Pattern	PGI_{DCGAN}	PGI_{CAE}	Selection Rate (%)
Center	14.25	12	4.44
Donut	1	1	100
Edge-Location	13.5	10	2.7
Edge-Ring	10	9	6.45
Location	9	6.7	3.37
Near-Full	5.33	5.33	18.75
Random	17.5	13.5	2.70
Scratch	2.5	1.9	13.89
Average	9.135	7.429	19.04

There are several limitations of the PGI when the quantity of the original data is highly scarce. First, as seen in the 'donut' defect pattern, the PGI cannot exceed 1 when $N_{Defect} = 1$, where N_{Defect} is the number of WBMs used to generate each defect pattern. In this case, it is difficult to use the PGI as an indicator reflecting how many distinct defect patterns the augmentation model can form. Moreover, it is difficult for the PGI to reveal how many more diverse defect patterns can be formed by augmented models, even when augmented results using only some defect patterns can generate accurately classifiable data for all original defect patterns not used for augmentation, as in the case of 'near-full' patterns. In other words, the PGI cannot be regarded as a true PGI. In this case, PGI may be considered to have a value of PGI > $N_{Defect}/N_{Augment}$. To alleviate the problem of PGI not reflecting a true PGI, we recommend referring to the results produced via verifying each augmented data entry using classification models trained using different augmented models, such as those in the experiments outlined in Tables 8 and 9.

Despite the limitations of the PGI listed above, it remains an effective index for the performance evaluation of augmented models. The PGI has the following advantages: first, it is simple and intuitive to calculate. As mentioned in Section 3.6, FID and IS are the typical metrics used to evaluate the performance of GAN-based generative models. However, these calculations require pre-trained models, such as the Inception v3 model. The PGI does not require an additional model for calculation. A large amount of sample data is required to evaluate the FID. Due to the nature of industrial sites, it is difficult to obtain a true FID because it is difficult to obtain sufficient abnormal data. Therefore, it may be difficult to use the FID as an evaluation index of the generation model. The calculation of the PGI uses relatively fewer sample data compared to FID. The PGI can be used as an evaluation index for generative models of images specialized to specific areas, such as industrial data, where it is difficult to apply pre-trained classifiers based on ImageNet. The PGI reflects the diversity of each class. However, IS is vulnerable to mode collapse due to its inability to consider inner diversity. For the above reasons, the PGI is easy to use as an intuitive index for evaluating the performance of a generative model.

5. Discussion

The experimental results in Tables 8 and 9 can be suspected not because the DCGAN-based model generates more diverse defect patterns than the CAE-based model, but because the DCGAN-based model creates a completely different type of WBM from the existing defect patterns. However, the results in Tables 6, 7 and 10 show that this is not a useful doubt. If the accuracy of the original WBM according to the classifier trained through the WBM generated from the DCGAN-based model is lower than the classifier trained through the WBM generated from the CAE-based model, it can be assumed that the WBM generated from the DCGAN-based model does not reflect the defect pattern characteristics of the original WBM. However, the experimental results showed that DCGAN-based

WBM augmentation methods always have a higher classification accuracy for each defect pattern compared to CAE-based WBM augmentation methods. For example, comparing the accuracy of 'random' defect patterns in Tables 8 and 9, the DCGAN-based model showed about a 26.2% higher classification performance difference for 'random' defect patterns, and the accuracy of the 'random' defect patterns in Tables 6 and 7 was also about 10.81% higher. The results showed that for all defect patterns, the classification performance of WBMs augmented from a CAE-based model was lower than that of WBMs augmented from a DCGAN-based model. Moreover, the CAE-based model tended to form relatively restrictive defect patterns for all defect patterns compared to the DCGAN-based model.

6. Conclusions

In this paper, we have proposed a DCGAN-based model as an effective way to augment defect patterns to improve the WBM defect pattern classification performance in WBM data where class imbalance in datasets is extremely severe. To confirm the effectiveness of the proposed method, the WBMs generated through two augmented methods trained using real-world data generated in the semiconductor manufacturing process were compared in terms of the pattern classification accuracy and the effectiveness of each augmented method. We have proposed a new quantitative index called the PGI to compare the effects of the augmented models. We have also proposed a masking process as a method for solving the problem of the quality degradation of augmented WBMs due to the tendency to augment by focusing on the defect pattern characteristics on wafers rather than absolute coordinates, which is one of the drawbacks of DCGAN-based models. In the masking process, the augmented WBMs were refined using a masking wafer.

We experimentally confirmed that DCGAN-based models can be applied to address the extreme data imbalance problem which is common in industrial data. The experimental results show that the accuracy of each defect pattern of the classifier trained using WBMs generated by the DCGAN-based model was always higher than that of the classifier trained using WBMs generated by the CAE-based model, on average about 14.26% higher for all defect patterns. Furthermore, the experiments show that WBMs augmented with CAE-based models are fully classified into defect patterns by classifiers trained with DCGAN-based augmented wafers, while some of the WBMs augmented with DCGAN-based models are not accurately classified by the classifier trained with CAE-based models. These results intuitively demonstrate that the DCGAN-based model formed various defect patterns which were better than the CAE-based model. Furthermore, we quantitatively evaluated the improvement efficiency of each augmentation model through the proposed evaluation index. The proposed evaluation index, PGI, does not require a pre-trained model such as Inception v3 to calculate the index. In addition, the PGI has the advantage that calculation is simple and intuitive, as well as relatively free from the demand for a large amount of original sample data, as required for index evaluation such as FID. The PGI can be used as an evaluation index for image generation models specialized for specific domains, such as industrial data, where it is difficult to use a pre-trained model based on the ImageNet dataset. The proposed index is expected to be used for a quantitative evaluation of augmentation models in cases such as extremely small and imbalanced industrial data, for which it is difficult to use existing evaluation indicators such as FID or IS. Through experimental results, we have shown that the proposed DCGAN-based augmentation model produced a better performance than the CAE-based augmentation model.

However, the PGI has the following issues:

- When $N_{Defect} = 1$, the PGI cannot exceed 1. In this case, it is difficult for the PGI to reflect how various defect patterns can be formed by the augmentation model.
- When $PGI = N_{Defect}/N_{Augment}$, it is difficult for the PGI to indicate how many more diverse defect patterns the augmented model can form. In this case, the PGI can be considered to have a value greater than or equal to at least $N_{Defect}/N_{Augment}$.
- The PGI depends on the performance of the classification model. When images generated from the same augmentation model are used to train classification models

composed of different structures, PGIs may differ depending on the structure of the classification model. Therefore, in order to compare the PGIs of different augmentation models, the results calculated from the same classification model should be used.

Nevertheless, due to the aforementioned advantages, the PGI is an effective quantitative evaluation metric for generative models.

Our future work will investigate more sophisticated generative models for the augmentation of extremely imbalanced data. We will further investigate the improvement of the proposed augmentation model based on DCGAN and explore WBM handling techniques that are better suited to the tasks of WBM defect pattern recognition and classification beyond the masking process. Additionally, we plan to study the structure of a generally applicable generative model for data classification with sparse classes. This is related to the expansion and variability of the proposed method. Furthermore, we intend to research to improve the limitations of PGI to make it more suitable for various application in general generative models. Furthermore, we plan to study new metrics by which these augmentation models can be evaluated. Ultimately, we plan to improve our research to make it suitable for augmenting all types of extremely imbalanced data.

Author Contributions: C.Y. and S.P. contributed to the design of the proposed scheme and wrote the manuscript. 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) grant funded by the Korean government (MSIP) (No. 2021R1A2C1005877).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Information about the data is contained within the text. We used the WM-811k dataset, which is publicly available data.

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

References

1. Shin, C.K.; Park, S. A machine learning approach to yield management in semiconductor manufacturing. *Int. J. Prod. Res.* **2000**, *38*, 4261–4271. [CrossRef]
2. Kumar, N.; Kennedy, K.; Gildersleeve, K.; Abelson, R.; Mastrangelo, C.M.; Montgomery, D.C. A review of yield modelling techniques for semiconductor manufacturing. *Int. J. Prod. Res.* **2006**, *44*, 5019–5036. [CrossRef]
3. Chien, C.-F.; Wang, W.-C.; Cheng, J.-C. Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Syst. Appl.* **2007**, *33*, 192–198. [CrossRef]
4. Hsu, S.-C.; Chien, C.-F. Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. *Int. J. Prod. Econ.* **2007**, *107*, 88–103. [CrossRef]
5. May, G.S.; Spanos, C.J. *Fundamentals of Semiconductor Manufacturing and Process Control*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2006; pp. 272–331.
6. Fan, S.-K.S.; Hsu, C.-Y.; Tsai, D.-M.; He, F.; Cheng, C.-C. Data-Driven Approach for Fault Detection and Diagnostic in Semiconductor Manufacturing. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1925–1936. [CrossRef]
7. Yuan, T.; Kuo, W.; Bae, S.J. Detection of Spatial Defect Patterns Generated in Semiconductor Fabrication Processes. *IEEE Trans. Semicond. Manuf.* **2011**, *24*, 392–403. [CrossRef]
8. Jin, C.H.; Na, H.J.; Piao, M.; Pok, G.; Ryu, K.H. A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 286–292. [CrossRef]
9. Liu, C.-W.; Chien, C.-F. An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing. *Eng. Appl. Artif. Intell.* **2012**, *26*, 1479–1486. [CrossRef]
10. Wang, C.-H.; Kuo, W.; Bensmail, H. Detection and classification of defect patterns on semiconductor wafers. *IIE Trans.* **2006**, *38*, 1059–1068. [CrossRef]
11. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1998.
12. Abukhodair, F.; Alsaggaf, W.; Jamal, A.T.; Abdel-Khalek, S.; Mansour, R.F. An Intelligent Metaheuristic Binary Pigeon Optimization-Based Feature Selection and Big Data Classification in a MapReduce Environment. *Mathematics* **2021**, *9*, 2627. [CrossRef]

13. Alabdulkreem, E.; Alduailij, M.; Alduailij, M.; Mansour, R.F. Optimal weighted fusion based insider data leakage detection and classification model for Ubiquitous computing systems. *Sustain. Energy Technol. Assess.* **2022**, *54*, 102815. [CrossRef]
14. Yu, Y.; Liang, S.; Samali, B.; Nguyen, T.N.; Zhai, C.; Li, J.; Xie, X. Torsional capacity evaluation of RC beams using an improved bird swarm algorithm optimised 2D convolutional neural network. *Eng. Struct.* **2022**, *273*, 115066. [CrossRef]
15. Yu, Y.; Li, J.; Li, J.; Xia, Y.; Ding, Z.; Samali, B. Automated damage diagnosis of concrete jack arch beam using optimized deep stacked autoencoders and multi-sensor fusion. *Dev. Built Environ.* **2023**, *14*, 100128. [CrossRef]
16. Borisov, V.; Scheible, J. Lithography hotspots detection using deep learning. In Proceedings of the 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Prague, Czech Republic, 2–5 July 2018; pp. 145–148.
17. Hong, S.J.; Lim, W.Y.; Cheong, T.; May, G.S. Fault Detection and Classification in Plasma Etch Equipment for Semiconductor Manufacturing e-Diagnostics. *IEEE Trans. Semicond. Manuf.* **2011**, *25*, 83–93. [CrossRef]
18. Popov, A.A.; Bilevich, D.V.; Salnikov, A.S.; Kalentyev, A.A.; Garays, D.V. Plasma-chemical etching process behavioral models based on tree ensembles and neural network. In Proceedings of the 2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE), Novosibirsk, Russia, 2–6 October 2018; pp. 128–131.
19. Blue, J.; Chen, A. Spatial Variance Spectrum Analysis and Its Application to Unsupervised Detection of Systematic Wafer Spatial Variations. *IEEE Trans. Autom. Sci. Eng.* **2010**, *8*, 56–66. [CrossRef]
20. Yoo, Y.; An, D.; Park, S.H.; Baek, J.-G. Wafer Map Image Analysis Methods in Semiconductor Manufacturing System. *J. Korean Inst. Ind. Eng.* **2015**, *41*, 267–274. [CrossRef]
21. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. Kdd* **1996**, *96*, 226–231.
22. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
24. RAWAT, Wassem; WANG, Zenghui. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef]
25. Yuan-Fu, Y. A deep learning model for identification of defect patterns in semiconductor wafer map. In Proceedings of the IEEE 2019 30th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA, 6–9 May 2019.
26. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
27. Shen, Z.; Yu, J. Wafer map defect recognition based on deep transfer learning. In Proceedings of the 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Macao, China, 15–18 December 2019.
28. Nakazawa, T.; Kulkarni, D.V. Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 309–314. [CrossRef]
29. Shon, H.S.; Batbaatar, E.; Cho, W.-S.; Choi, S.G. Unsupervised Pre-Training of Imbalanced Data for Identification of Wafer Map Defect Patterns. *IEEE Access* **2021**, *9*, 52352–52363. [CrossRef]
30. Kang, S. Rotation-Invariant Wafer Map Pattern Classification With Convolutional Neural Networks. *IEEE Access* **2020**, *8*, 170650–170658. [CrossRef]
31. Ji, Y.; Lee, J.-H. Using GAN to Improve CNN Performance of Wafer Map Defect Type Classification: Yield Enhancement. In Proceedings of the 2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA, 24–26 August 202l; pp. 1–6. [CrossRef]
32. Salazar, A.; Vergara, L.; Vidal, E. A proxy learning curve for the Bayes classifier. *Pattern Recognit.* **2023**, *136*, 109240. [CrossRef]
33. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
34. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434, 2015.
35. Wu, M.-J.; Jang, J.-S.R.; Chen, J.-L. Wafer Map Failure Pattern Recognition and Similarity Ranking for Large-Scale Data Sets. *IEEE Trans. Semicond. Manuf.* **2014**, *28*, 1–12. [CrossRef]
36. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
37. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400, 2013.
38. Tsai, T.-H.; Lee, Y.-C. A Light-Weight Neural Network for Wafer Map Classification Based on Data Augmentation. *IEEE Trans. Semicond. Manuf.* **2020**, *33*, 663–672. [CrossRef]
39. Barratt, S.; Sharma, R. A note on the inception score. *arXiv* **2018**, arXiv:1801.01973, 2018.
40. Lee, H.; Kim, H.E.; Nam, H. Srm: A style-based recalibration module for convolutional neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1854–1862.
41. Brendel, W.; Bethge, M. Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. *arXiv* **2019**, arXiv:1904.00760.

42. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
43. Masci, J.; Meier, U.; Cireşan, D.; Schmidhuber, J. Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. In *Artificial Neural Networks and Machine Learning—ICANN 2011, 21st International Conference on Artificial Neural Networks, 14–17 June 2011, Espoo, Finland*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; pp. 52–59.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.