mathematics

MDPI

Article

Elevating Wafer Defect Inspection with Denoising Diffusion Probabilistic Model

Ping-Hung Wu [1], Thi Phuong Hoang [2,†], Yen-Ting Chou [2,†], Andres Philip Mayol [3,4], Yu-Wei Lai [5,†], Chih-Hsiang Kang [5,†], Yu-Cheng Chan [5,†], Siou-Zih Lin [6,†] and Ssu-Han Chen [2,5,*]

[1] Product Testing Service Office, Nanya Technology Corporation, New Taipei City 243089, Taiwan; smith@ntc.com.tw

[2] Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City 243303, Taiwan; m11218031@mail2.mcut.edu.tw (T.P.H.); hjhjkoko2022@gmail.com (Y.-T.C.)

[3] Manufacturing Engineering and Management Department, De La Salle University, Manila 0922, Philippines; andres_mayol@dlsu.edu.ph

[4] Center for Engineering and Sustainable Development Research, De La Salle University, Manila 0922, Philippines

[5] Center for Artificial Intelligence & Data Science, Ming Chi University of Technology, New Taipei City 243303, Taiwan; lywei@mail.mcut.edu.tw (Y.-W.L.); c.h.kang@o365.mcut.edu.tw (C.-H.K.); chanyucheng@yahoo.com (Y.-C.C.)

[6] AI Chip Application & Green Manufacturing Department, Industrial Technology Research Institute, Hsinchu 310401, Taiwan; r97546039@gmail.com

[*] Correspondence: ssuhanchen@mail.mcut.edu.tw

[†] These authors contributed equally to this work.

Abstract: Integrated circuits (ICs) are critical components in the semiconductor industry, and precise wafer defect inspection is essential for maintaining product quality and yield. This study addresses the challenge of insufficient sample patterns in wafer defect datasets by using the denoising diffusion probabilistic model (DDPM) to produce generated defects that elevate the performance of wafer defect inspection. The quality of the generated defects was evaluated using the Fréchet Inception Distance (FID) score, which was then synthesized with real defect-free backgrounds to create an augmented defect dataset. Experimental results demonstrated that the augmented defect dataset significantly boosted performance, achieving 98.7% accuracy for YOLOv8-cls, 95.8% box mAP for YOLOv8-det, and 95.7% mask mAP for YOLOv8-seg. These results indicate that the generated defects produced by the DDPM can effectively enrich wafer defect datasets and enhance wafer defect inspection performance in real-world applications.

Keywords: wafer defect inspection; generative model; denoising diffusion probabilistic model; You Only Look Once Version 8

MSC: 68U10

check for updates

Citation: Wu, P.-H.; Hoang, T.P.; Chou, Y.-T.; Mayol, A.P.; Lai, Y.-W.; Kang, C.-H.; Chan, Y.-C.; Lin, S.-Z.; Chen, S.-H. Elevating Wafer Defect Inspection with Denoising Diffusion Probabilistic Model. *Mathematics* **2024**, *12*, 3164. https://doi.org/10.3390/math12203164

Academic Editors: Chia-Ding Hou and Rung-Hung Su

Received: 1 September 2024
Revised: 24 September 2024
Accepted: 8 October 2024
Published: 10 October 2024

1. Introduction

In the semiconductor industry, wafers play an indispensable role in development and assembly operations. During the manufacturing process, various factors, such as environmental conditions and process parameters, can affect product quality, leading to defects on the wafer surface. These defects can significantly impact the overall yield of the final production. As the electronics industry's demand for innovation escalates, wafers have tended to increase in size. With this continuous growth, wafer defects have become smaller and more complex, increasing the demand for accurate and real-time quality monitoring and control. Accurately inspecting wafer surface defects can expedite the identification of abnormalities in the production process [1]. Therefore, defect inspection during wafer production is crucial for promoting high yield, cost efficiency, and optimal performance.

In most semiconductor assembly operations, assessing wafer surface quality primarily relies on experienced inspectors conducting manual visual inspections. However, this task demands intense concentration, limiting the duration an inspector can remain productive. With increasing pressure to meet these demands, the human visual inspection process becomes prone to subjectivity, slowness, and inaccuracies, increasing the likelihood of process-based defects on the wafer surface and potentially decreasing yield [2]. Additionally, rising labor shortages, time consumption, and low consistency may result in inadequate defect labeling. As integrated circuit (IC) feature sizes continue to shrink and semiconductor processes grow more complex, new wafer defect classes emerge, further limiting yield. Consequently, the defect datasets often lack sufficient diversity, preventing them from revealing the full range of defects [3]. Given that human visual inspection increasingly fails to meet the demands of modern industrialized production [4], the scarcity of diverse defect datasets and the high cost of labeling during wafer production present significant challenges. It is difficult to obtain adequate samples of defect patterns with high diversity through manual visual inspection, which seriously affects the accuracy of identifying various defect types on the wafer surface.

Machine vision inspection has become a highly efficient and promising method for inspecting surface defects in wafer dies and has gradually replaced traditional human visual inspection. For such inspection, optical imaging charge-coupled devices (CCDs) are primarily employed. While integrating machine vision into wafer defect inspection increases efficiency and reduces time consumption, defect inspection algorithms cannot be directly applied. They need to be highly customized, require accurate alignments, and often rely heavily on expert knowledge to set up hand-crafted features [5], which can lead to inaccuracies, subjectivity, and inconsistency in defect inspection. Even deep learning models like convolutional neural networks (CNNs), which have shown remarkable performance in image classification, object detection, and object segmentation [6], while effective, require large and diverse datasets to achieve high performance. However, the wafer die images are collected from multiple customers in which the background patterns are totally different. The ongoing inspection machines even only capture the defective patches, resulting in a more diverse background. Furthermore, the defective images should encompass a wide range of patterns and be thoroughly annotated by professional inspectors with extensive training and experience in quality control.

To address the aforementioned issues, the denoising diffusion probabilistic models (DDPMs) [7] combine with proposed image preprocessing and image synthesis processes to synthesize realistic and high-quality wafer defects, even with diverse background patterns. The DDPM is able to model complex defect distributions and produce high-quality generated defective region images, even in the presence of noise or when the dataset is small. Additionally, for real images, the developed mechanism automatically annotates defect bounding boxes or defect contours through database similarity comparisons and image subtraction. For generated images, defects are automatically annotated based on the characteristics of their white background. The only task left for the users is to label the defect class, greatly reducing the burden of annotation. By generating diverse and realistic defect images, the DDPM augments the training dataset for YOLOv8, improving its ability to accurately inspect wafer defects.

The remainder of this paper is organized as follows: Section 2 focuses on related research on the surface defect inspection with the aid of generated images. Section 3 describes the methodology and implementation process. Section 4 presents the experimental results of generated images produced by generative models. Finally, Section 5 concludes this research.

2. Surface Defect Inspection Using Generated Images

Goodfellow proposed the generative adversarial network (GAN) [8]. The GAN is an unsupervised learning model where the generator, a neural network, takes random noise as input and produces synthetic images. Meanwhile, the discriminator, another neural network, determines whether the input is real or generated. GANs can generate numerous patterns in a short time and have shown better performance than other algorithms, particularly in generating diverse images. Recently, the DDPM, a newly emerging paradigm, has gained widespread attention in deep generative models, alongside the GANs. Both the GAN and DDPM use noise for image generation. The key difference between these models is that the GAN relies on adversarial training with two competing neural networks based on random noise, while the DDPM randomly samples two-dimensional noise maps using a probabilistic model and diffusion process to generate images. Yang et al. reviewed the DDPM's record-breaking performance in various generation tasks, including image synthesis and video generation [9]. Compared with state-of-the-art GAN-based methods, the DDPM achieves better sample quality in image synthesis by training on a stationary objective, although it is slower than GANs during sampling due to multiple denoising steps [10]. For these reasons, this research aims to adapt GANs and the DDPM to generate synthetic wafer defect datasets accurately. By using these methods to construct synthetic wafer defective images, this research aims to generate images of wafer datasets for defect inspection. The following sections detail the implementation and results of using these models for producing generated images and improving defect inspection performance.

2.1. Surface Defect Inspection Using GAN-Generated Images

CNN-based methods have achieved significant success in surface defect inspection. However, when the dataset is limited in size or suffers from noise, these methods face considerable challenges. The primary obstacle is the limited ability to extract effective features. To address this issue, generative models like GANs have been employed to enrich the training data with synthesized samples, which in turn enhance performance.

Several studies have demonstrated the success of GAN variants in generating images for defect inspection. For instance, Chen et al. successfully applied GANs to generate synthetic wafer defect images, which were then used to enhance the training dataset for YOLOv3-based detection, resulting in a performance boost of 7.33% in average precision (AP) [5]. Similarly, Guo et al. proposed ISU-GAN, which integrates a skip connection and attention mechanism to improve feature extraction [11]. This model demonstrated high accuracy (98.43%) and a strong F1 score (97.92%), proving effective in defect detection and segmentation tasks. Moreover, advanced GAN architectures such as DGGAN [12] and CycleGAN [13] have shown considerable improvements in specific industrial defect detection scenarios. DGGAN, which incorporates data augmentation and self-attention mechanisms, demonstrated a notable improvement in YOLOX detection accuracy by 6.1% to 20.4%. CycleGAN, applied to generate pineapple surface defects, enhanced the realism of generated images while maintaining the original image characteristics, resulting in an AP of 84.86%. These examples illustrate the diverse applications and effectiveness of GANs in generating high-quality defect images for training, thereby overcoming data scarcity and enhancing defect inspection systems. In the field of steel defect detection, models such as MAS-GAN [14] and NAM-DCGAN [15] have also demonstrated significant advances. MAS-GAN improved defect detection accuracy by focusing on data augmentation combined with a self-attention mechanism, while NAM-DCGAN enhanced feature extraction in solar cell defect detection through the use of normalization-based attention modules and MobileNet-V3 for final detection tasks. These models highlight the effectiveness of GANs in improving defect classification accuracy and generating synthetic datasets with a high degree of realism and diversity.

Despite long-standing exploration in the field of machine vision, GAN variants continue to enhance defect inspection by enabling feature control in cases with limited defect categories and regions.

2.2. Surface Defect Inspection Using DDPM-Generated Images

The DDPMs represent a novel and powerful approach to synthetic image generation. Unlike GANs, which rely on adversarial training between a generator and discriminator, DDPMs employ a diffusion process that gradually adds noise to images and then reverses the process to generate new images. This technique allows DDPMs to produce highly diverse and accurate synthetic images, making them a valuable tool for defect inspection, particularly when high fidelity and complexity are required.

Several studies have demonstrated the advantages of DDPMs over GANs in defect detection tasks. Tang et al. introduced the SN-DDPM, which adapts DDPM for fabric defect inspection [16]. By focusing on key features of the defect-free region, SN-DDPM was able to reconstruct defective images with high fidelity, significantly outperforming other unsupervised reconstruction methods in F1 and intersection of union (IoU) metrics. Similarly, Liu et al. applied the DDPM to the MURA dataset and showed impressive results in generating defective images with improved reasoning for both backgrounds and defects, evaluated through Inception score (IS), Fréchet Inception Distance (FID) score, and YOLOv5 metrics [17]. In the industrial sector, Semchyshyn applied DDPMs for high-resolution defect defective generation, focusing on poorly reconstructed areas [18]. By combining the DDPM with U-Net for segmentation, this approach achieved a superior area under curve (AUC) compared with GAN-based models, demonstrating its effectiveness in accurately reconstructing defects and inpainting defect-free regions. Huang et al. also demonstrated the power of DDPMs in detecting micro-defects in composite plates, outperforming other generative methods in terms of FID score, accuracy, and recall [19]. DeRidder et al. combined the DDPM with Mask R-CNN and ResNet for wafer defect detection and segmentation, showcasing the method's versatility in handling both object detection and instance segmentation tasks within a single model [20]. This integrated approach achieved superior performance in AP and mean average precision (mAP) compared with other models, making it a robust solution for defect detection in complex industrial environments.

Overall, DDPMs have proven to be a highly effective alternative to GANs for generating high-quality generated defect images, especially when the emphasis is on maintaining fine details and achieving diversity in image generation. Their ability to model complex distributions and recover fine features from noisy data makes them a valuable tool for improving surface defect inspection in various industries.

3. Research Methods

In this study, we aim to enhance the capability of YOLOv8 for wafer defect inspection, building upon the defect inspection process based on the augmented dataset generated by generative models. The overall flowchart is shown in Figure 1. First of all, wafer defective patches were combined with corresponding defect masks for two goals: one is to extract the defect region from the background, and the other is to pair up defective patches and masks for a further defect inspection stage. After image preprocessing, the real defective region images were used as input for the DDPM to produce the generated defective regions. Subsequently, the golden templates were served as defect-free backgrounds that were synthesized from the generated defective regions. Both the real and generated defective patches constructed a defect dataset. This dataset was auto-annotated with defect masks and can be used for the tasks of defect classification, defect detection, and defect segmentation using YOLOv8. Finally, the results were evaluated to demonstrate the performance of this method.

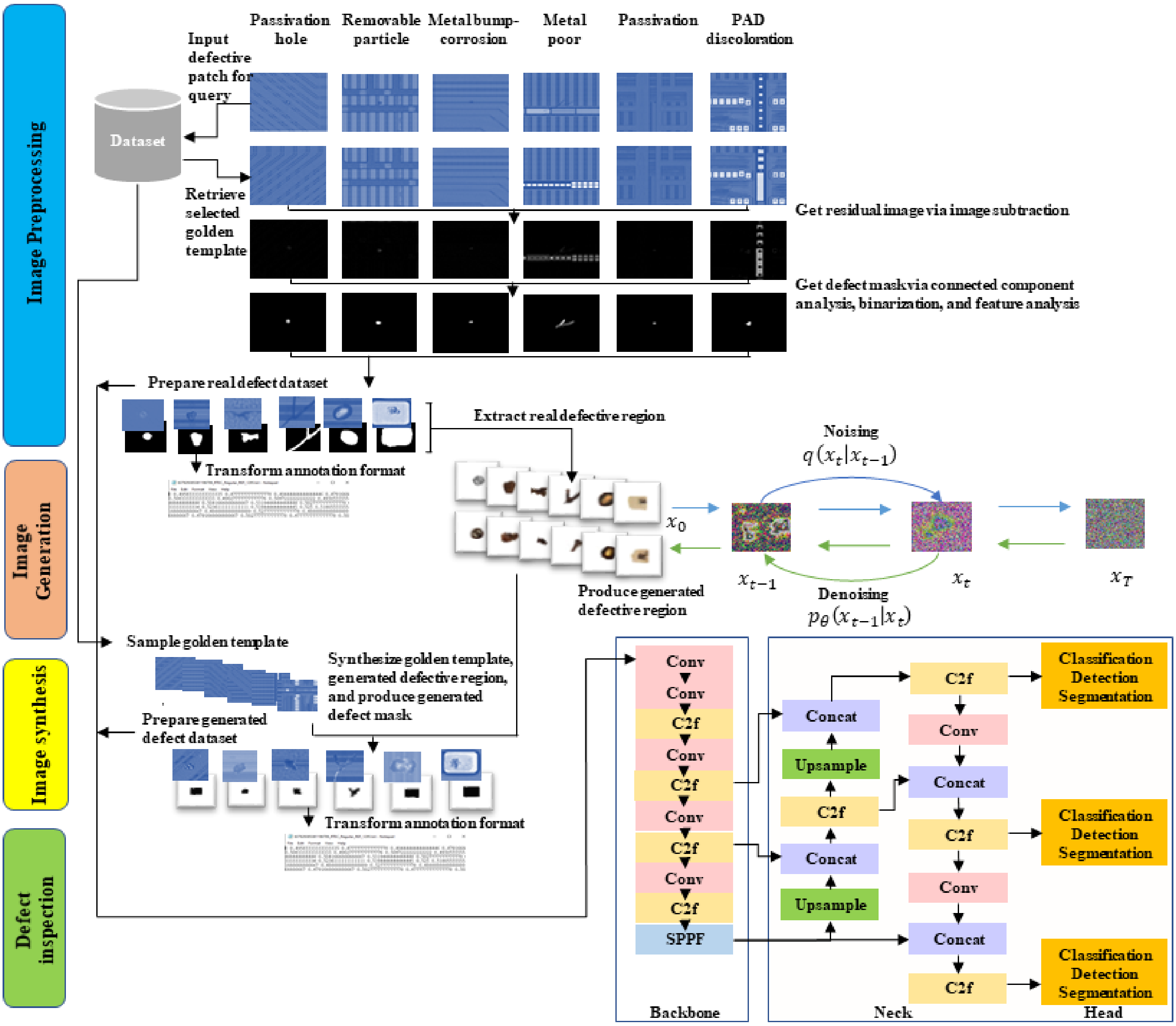


Image Preprocessing
Image Generation
Image synthesis
Defect inspection
Dataset
Input defective patch for query
Retrieve selected golden template
Passivation hole
Removable particle
Metal bump-corrosion
Metal poor
Passivation
PAD discoloration
Get residual image via image subtraction
Get defect mask via connected component analysis, binarization, and feature analysis
Prepare real defect dataset
Transform annotation format
Extract real defective region
Noising
$q(x_t|x_{t-1})$
x_0
x_{t-1}
x_t
x_T
Denoising
$p_\theta(x_{t-1}|x_t)$
Produce generated defective region
Sample golden template
Synthesize golden template, generated defective region, and produce generated defect mask
Prepare generated defect dataset
Transform annotation format
Conv
Conv
C2f
Conv
C2f
Conv
C2f
Conv
C2f
SPPF
Backbone
Concat
Upsample
C2f
Concat
Upsample
C2f
Conv
Concat
C2f
Conv
Concat
C2f
Neck
Classification Detection Segmentation
Classification Detection Segmentation
Classification Detection Segmentation
Head

Figure 1. Overall flowchart of the proposed method.

3.1. Image Preprocessing

The wafer dataset used in this experiment was provided by a semiconductor manufacturer in Taiwan. Wafer defect images were captured using a machine vision system under clean room conditions. During production, defects on the wafer surface may occur due to workers or machinery, such as holes, particles, corrosion, metal poor, passivation, discoloration, and other imperfections. This dataset contains six types of defects categorized by texture, shape, strength, and size, with each image containing only one type of defect.

The underlying machine vision machine performs an S-shaped motion across the surface of each wafer and captures images using a CCD. Each field of view of the CCD covers a partial patch of the wafer die [21]. For each patch, we search and match the database for the corresponding selected golden template, then perform image subtraction with the patch to obtain a residual image. Moreover, connected component analysis, binarization, and feature analysis are performed, and a defect mask is produced based on the remaining blobs. The underlying machine vision machine outputs a quadruplet for each sample, including the defective patch, the selected golden template, the residual image, and the defect mask, as shown in Figure 1. In this study, the colored defective patch represents the input image, the binarized mask image corresponds to the label file for the input image, and the colored golden template represents the defect-free background image. The residual image, however, was not used in this study. Due to confidentiality agreements, the images have been modified through techniques such as discoloration and flipping to protect their original features. As the causes and characteristics of defects and background vary, human experts categorize the defects manually.

Given that the defective region occupies only a small portion of the overall wafer patch, it can be challenging to accurately recognize and generate defect features in high resolution. To improve generative model performance, we focused solely on generating the defective region by separating it from the background [5,13]. To do so, we utilized real defect masks to isolate and crop the central defective region, as shown in Figure 1, effectively reducing the collapse phenomenon in generative models during training.

The generative models produce generated defective regions at the same resolution as the real ones, but these regions remain similarly small after cropping. We hypothesize that generating the defect in its original form after preprocessing steps, such as resizing, generation, and transformation, is challenging, especially due to the small size of the defects. To address this issue, we cropped the real defective region to 96 × 96 × 3 pixels for model input, ensuring that the generated defect dataset match this size, thereby mitigating any loss in resolution.

3.2. Image Generation

The DDPM [7] is a generative model that restores input images based on a sequential process of adding noise to the images and then performs the inverse transformation of the additional diffusion process. While the GAN performs image generation, based on a generator and a discriminator, the DDPM restores the image by noising (diffusion process) and denoising (inverse process), as shown in Figure 1.

During the training process of the DDPM, the real defective region x_0 undergoes a progressive noising process, where noise with variance β_t is added incrementally from time t = 0 until t = T. At each step, the previous sample x_{t-1} is scaled by $(1 - \beta_t)^{0.5}$. This noising process is defined by Equation (1), which represents the forward nosing process as a product of conditional distributions $q(x_t \mid x_{t-1})$. Each subsequent sample x_t depends on the previous sample x_{t-1} with noise added at each step:

$$q(x_{1:T}|x_0) \ := \prod\nolimits_{t=1}^{T} q(x_t|x_{t-1}) \tag{1}$$

Equation (2) further specifies that $q(x_t \mid x_{t-1})$ follows a Gaussian distribution, with a mean of $(1 - \beta_t)^{0.5}{}_{\times} x_{t-1}$ and variance β_t scaled by the identity matrix I. As the time steps increase, the image x_t becomes progressively noisier until, at t = T, x_{T} follows a Gaussian distribution:

$$q(x_t|x_{t-1}) := N\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right) \tag{2}$$

The reverse of the noising process is the denoising stage p, which starts from a Gaussian sample x_{T} and attempts to recover x_0. This process is modeled by the function $p_\theta(x_{t-1} \mid x_t)$, as shown in Equation (3), where θ represents the parameters of a neural network estimating p. The purpose of the DDPM is to learn this inverse process, so the parameters of the Gaussian Markov chain are learned through parameterization:

$$p_\theta(x_{0:T}) := p(x_T)\prod\nolimits_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{3}$$

The conditional distribution $p_\theta(x_{t-1} \mid x_t)$ in the reverse denoising process is defined in Equation (4). It assumes a Gaussian distribution with a mean $\mu_\theta(x_t, t)$, which is a function of the current noisy sample x_t and the time step t, and a variance $\alpha^2{}_t$ scaled by I. The mean

μ_θ is predicted by the neural network, allowing the model to gradually denoise the image as it moves backward through the time steps, eventually recovering the original image x_0:

$$p_\theta(x_{t-1}|x_t) := N\left(x_{t-1}; \mu_\theta(x_t, t), \alpha_t^2 I\right) \tag{4}$$

These two processes aim to minimize the discrepancy between the real image and the output, producing an image that closely resembles the input. The deep model used for the DDPM is a U-Net. The U-Net is trained to predict the noise added to the input image during the forward diffusion process. The input to the U-Net is the noisy image x_t, obtained by gradually adding Gaussian noise to the original image x_0 over time steps t. The ground truth is the actual noise ϵ applied to the image at that specific time step. The U-Net learns to predict this noise by minimizing the mean squared error (MSE) between the predicted noise and the real noise, helping the model denoise images step by step.

3.3. Image Synthesis

The process of image synthesis is shown in Figure 1. Inspired by the inpainting method, we augment dataset diversity by reinserting the generated defective regions into defect-free backgrounds, which are randomly sampled from the set of selected golden templates. By placing generated defective regions at their original crop locations and blending them with defect-free backgrounds, this technique not only increases the variety of generated samples but also retains some of the original image information, leading to more realistic results. This method enables the model to generate defective samples that better simulate real-world conditions, where defects may occur against various backgrounds. Moreover, since the surrounding pixels of the generated defective regions are white, their corresponding generated masks can be easily produced through automatic binarization.

The FID score is a widely used metric for evaluating generative quality of generated defective images [22]. The FID score measures the distance between the distributions of generated and real images by extracting features from the Inception neural network, reflecting discrepancies in their sample distributions:

$$FID(r,g) = \|\mu_r - \mu_g\|^2 + Tr\left(\sum r + \sum g - 2\left(\sum r \sum g\right)^{1/2}\right) \tag{5}$$

where $\mathrm{N}(\mu_r, \sum r)$ and $\mathrm{N}\left(\mu_g, \sum g\right)$ are the Gaussian respectively fitted to the real image and the generated image. $Tr(\sum r + \sum g - 2(\sum r \sum g)^{1/2})$ denotes the trace of a matrix, capturing the difference between the covariances of the feature vectors of the real and generated images. The reliability of this metric is based on how informative the features provided by the Inception network are regarding image quality and the appropriateness of assuming a Gaussian distribution. A lower FID score indicates a smaller difference between the real and generated image distributions, signifying the higher quality of the generative model. Typically, FID scores below 10 are considered unbiased and indicative of strong generative capabilities [23].

3.4. Defect Inspection

As shown in Figure 1, the structure of the YOLOv8 generally incorporates several key components, each with a specific function in the process of making predictions from input images.

The YOLOv8 backbone uses a series of convolutional layers and cross-stage partial bottleneck (C2f) blocks to progressively extract features from input images. Each C2f block splits the input feature map, processes one part through a bottleneck of convolutions and activations, and then concatenates it with the bypassed features. This structure allows the model to retain detailed information while improving computational efficiency. Additionally, the spatial pyramid pooling-fast (SPPF) layer is at the end of the backbone, helping to capture multi-scale features by pooling information at different levels, which enhances the model's ability to detect objects of various sizes. As the backbone processes the image, it reduces its resolution and increases the number of feature channels, capturing both fine and abstract features for accurate predictions.

Next, the neck of YOLOv8 takes the extracted features from the backbone and further processes them. The neck is designed to refine these features and may include specialized mechanisms such as feature pyramid networks (FPN) or path aggregation networks (PAN). These structures are particularly adept at merging features from multiple scales, ensuring that the model is able to utilize both fine and coarse information for accurate predictions.

At the end of this process lies the head of YOLOv8, which is tasked with generating the final predictions with three primary prediction targets: YOLOv8-cls for image classification, YOLOv8-det for object detection, and YOLOv8-seg for instance segmentation. YOLOv8-cls, the simplest of the three, focuses on classifying the defect embedding in an entire image into one of several predefined classes. The output of this classifier is a single class label with accuracy serving as the evaluation metric. YOLOv8-det is used for defect detection, a more complex task that involves identifying both the locations and classes of defects within an image. The detector outputs bounding boxes that enclose the identified defects, along with class labels and corresponding confidence scores for each box. Performance for this task is commonly measured using the mean average precision for bounding boxes (box mAP). YOLOv8-seg goes a step beyond defect detection by performing instance segmentation, which not only identifies defects in an image but also segments them from the rest of the image at the pixel level. The performance of YOLOv8-seg is similarly evaluated using mAP, but this time for segmentation masks, called mask mAP.

4. Experiment Results

4.1. Expanded Dataset Description

The dataset comprises 2060 samples, all standardized to a resolution of $720 \times 720 \times 3$ pixels. Aiming to expand the wafer defect dataset, this research combines the generated defect dataset produced by generative models with the real defect dataset to increase the sample size of defect patterns for defect inspection models. The generated images, produced by generated models, have defective regions with an image size of $96 \times 96 \times 3$ pixels. After obtaining the generated defective regions from generative models, we need to perform a synthesis image stage. The generated defective regions are combined with the defect-free background. We then prepare the corresponding generated masks, which can be used for the tasks of defect classification, defect detection, and defect segmentation.

For the defect classification task in YOLOv8-cls, we use only defective images as input to the model, along with their corresponding class labels. For the defect detection task in YOLOv8-det, we transform the real or generated masks into labels in YOLO format, which include the coordinates of the center, as well as the width and height of the bounding box for each defective region. The bounding box is derived by enclosing the mask with the minimum and maximum coordinates that define its outer boundaries. The YOLOv8-det uses paired images and labels to detect defects within bounding boxes. For the defect segmentation task in YOLOv8-seg, we transform the real and generated masks into labels in YOLO format. These labels contain multiple point coordinates that accurately represent the shape of the defective region. The polygon points for these labels are extracted from the mask by tracing its contours. The contour detection algorithm identifies the boundary points that form a polygon around the mask, allowing for precise labeling of the defective region. The YOLOv8-seg uses paired images and labels to segment defects within masks.

For quality evaluation, the dataset is grouped by the ratio between real and generated images to assess the enhancement capability of generated images for defect inspection performance. We use four levels of data expansion for all the quality experiments in this research. The first level of data augmentation uses a ratio of 1-0, with 150 real defective images for most defect types, except for defect type “Passivation”, which has only 114 images due to the scarcity of defective patterns. The total number of images in the dataset with a ratio of 1-0 is 864 images. The second level of data augmentation uses a ratio of 1-1, where we maintain the real dataset quantity from the 1-0 ratio and add an equal number of generated images. For the third level of data augmentation, with a ratio of 1-2, we double the quantity of generated images, and, for the fourth level, with a ratio of 1-3, we triple the quantity of generated images. The ratio of training, validation, and testing in the dataset will be distributed as 80-10-10 across all levels of data augmentation. We use two expanded datasets generated from GANs and the DDPM in parallel to compare the performance enhancement between the two models.

4.2. Quantitative and Qualitative Comparisons of Image Generation Quality among GANs and the DDPM

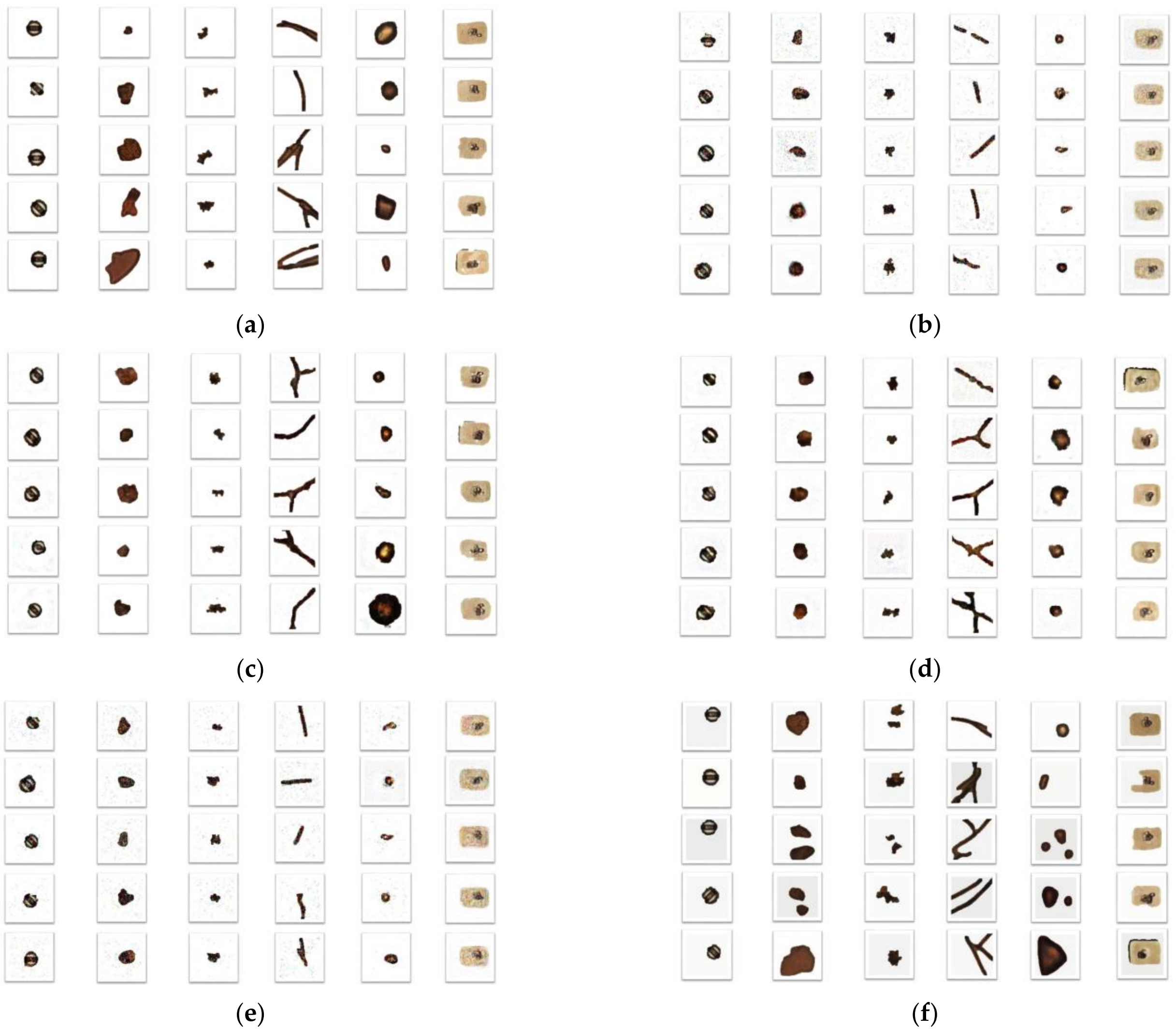
To evaluate the quality of defect dataset generated by different generative models, we use the FID score as the primary metric. Table 1 compares the FID scores among real, vanilla GAN [8], DCGAN [24], ACGAN [25], BGAN [26], and the DDPM [7] for different kinds of defects. The second column represents the baseline FID scores calculated from the real defective images themselves, serving as a reference for comparison against the FID scores of the generated images. The FID score for all kinds of real defects is close to zero. Lower FID scores for the DDPM indicate that its generated images are closer in quality to the real defects, whereas the GAN's higher FID scores reflect a greater deviation from the real patterns. In our comparison, the DDPM consistently outperformed GANs across all defect types, achieving significantly lower FID scores, with all values below 10, indicating superior image generation quality. This strong performance highlights the DDPM's superior capability in generating defect patterns with higher fidelity. Moreover, the DDPM's consistent results across various defect types underscore its robustness and effectiveness in generating realistic defect images.

Table 1. The FID score evaluation for real and generated images.

Image Sources **Kinds of Defects**	**Real**	**Vanilla GAN [8]**	**DCGAN [24]**	**ACGAN [25]**	**BGAN [26]**	**DDPM [7]**
Passivation hole	0.38	23.10	14.25	22.16	22.85	8.21
Removable particle	0.57	31.66	18.61	29.33	31.89	8.91
Metal bump corrosion	0.35	23.00	11.68	21.03	18.05	7.61
Metal poor	0.44	24.86	15.29	23.71	22.99	9.05
Passivation	0.41	25.36	12.42	23.38	24.74	8.48
PAD discoloration	0.29	14.50	9.53	14.16	15.33	6.97

To evaluate the defect data augmentation of generative models, we also assess the image generation ability visually. The description of the real defective region and the generated defective regions produced by vanilla GAN, DCGAN, ACGAN, BGAN, and the DDPM are shown in Figure 2a–f, respectively. We observed that both DCGAN and the DDPM can generate defective regions with the signature features of each type of defect. We believe that the generation performance of generative models is due to the preprocessing step, which removes the background and concentrates on generating the defective region only. This method effectively guides the generative models to focus on defect characteristics such as shape, size, texture, location, and intensity.

However, upon closer examination, it becomes clear that the vanilla GAN and BGAN still introduce noticeable noise in the generated images. While they capture some characteristics of the defects, the results often suffer from unwanted artifacts and a lack of sharpness. ACGAN, on the other hand, performs better but occasionally generates blurry images with noise in specific defect types, particularly in complex patterns. Compared with the vanilla GAN, ACGAN, and BGAN, DCGAN performs significantly better by producing clearer images with fewer artifacts, especially when generating complex defect structures. However, DCGAN mostly preserves the shape and size of the real defects, and many results appear blurry and do not maintain the same degree of sharpness as the originals when compared with the DDPM. The DDPM impressively generates the most natural-looking features while properly maintaining the identity of the real defects.



(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

(**f**)

Figure 2. Defective regions for each class: (**a**) real images; (**b**) generated by vanilla GAN; (**c**) generated by DCGAN; (**d**) generated by ACGAN; (**e**) generated by BGAN; (**f**) generated by DDPM.

We further counted the number of defective regions generated in each image. We found that the DDPM outperforms DCGAN and other GANs in generation ability by generating more than two defective regions in one generated image. Although the input images for both DCGAN and the DDPM are the same, with each input image containing only one defective region, the DDPM demonstrated an impressive ability in image generation. The DDPM not only learns the features of defects from the input dataset but also estimates the number of defects that can be generated for each type of defect. In contrast, DCGAN and other GANs can generate only one defective region per image, similar to the real dataset.

4.3. Quantitative and Qualitative Comparisons of Defect Inspection Performance between DCGAN and DDPM Associated with YOLOv8

To evaluate the inspection performance of the expanded datasets from DCGAN and the DDPM, we chose to test with YOLOv8 in three version classification, object detection, and object segmentation and in all four expansion levels of datasets.

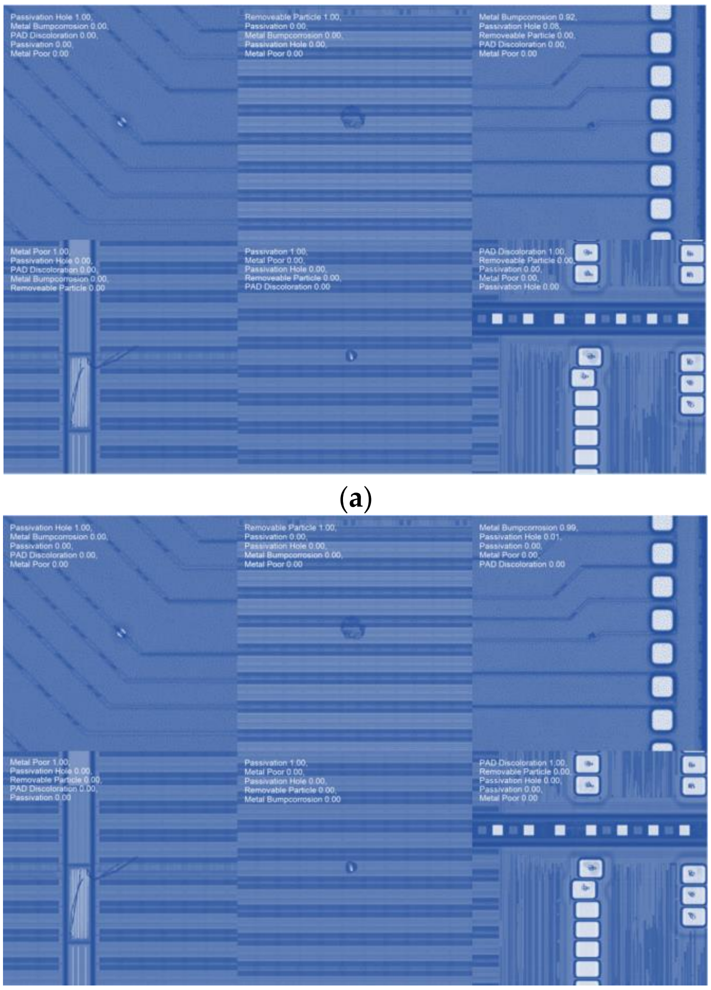
The overall result of YOLOv8 on the expanded dataset is shown in Table 2. In summary, the DDPM demonstrates superior overall performance in classification and competitive performance in detection and segmentation tasks, making it a robust choice for defect image analysis. DCGAN, while slightly less accurate in classification, excels in detection tasks, particularly with more augmented data. Both models significantly enhance defect detection and segmentation over the original dataset.

Table 2. The inspection performance of YOLOv8 on dataset with different data augmentation ratio.

Data Augmentation Ratio	**1-0**	**1-1**		**1-2**		**1-3**	
Method	**Original**	**DCGAN**	**DDPM**	**DCGAN**	**DDPM**	**DCGAN**	**DDPM**
YOLOv8-cls (Accuracy)	0.960	0.975	0.958	0.979	0.985	0.985	**0.987**
YOLOv8-det (Box mAP)	0.923	0.902	0.955	0.561	**0.958**	0.615	0.925
YOLOv8-seg (Mask mAP)	0.911	0.937	0.945	0.939	**0.957**	0.597	0.938

The bolded values indicate the best performance for each method under varying data augmentation ratios.

For the defect classification task, the classification results of YOLOv8-cls on the augmented dataset generated by DCGAN and the DDPM are shown in Figures 3a and 3b, respectively. As shown in Table 2, we observed that the classification performance was clearly enhanced in both DCGAN and the DDPM when the level of generated images was increased. This means generative models can improve inspection accuracy in the classification task. Specially, the DDPM achieved a higher accuracy than DCGAN across all ratios with the highest accuracy of 98.7% for ratio 1-3 (level 4), indicating its superior performance in this task.

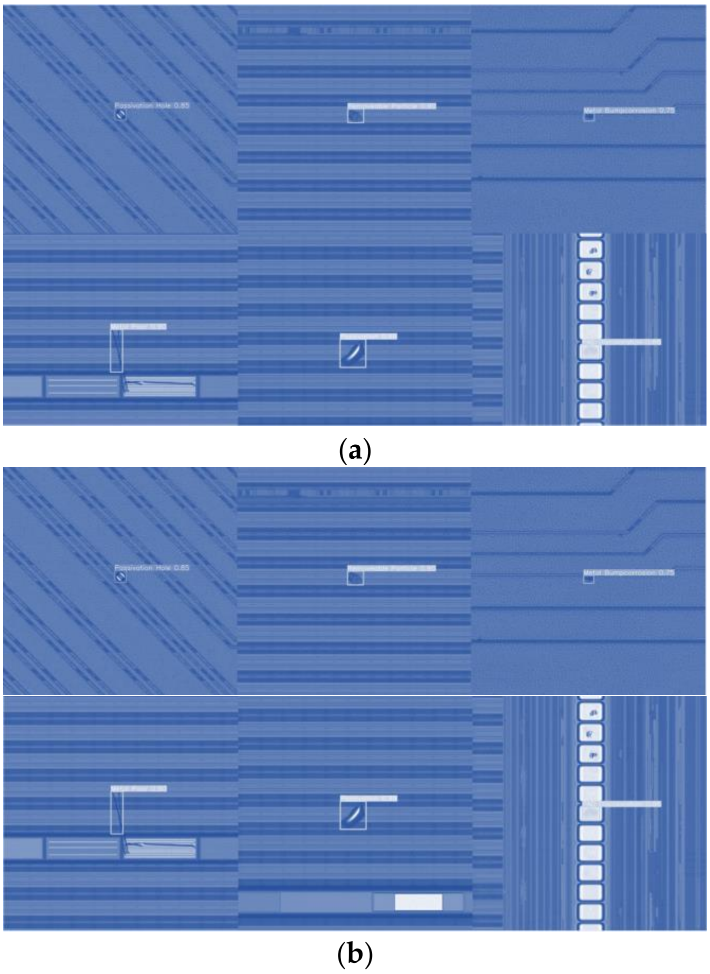


(**a**)

(**b**)

Figure 3. Defect classification results: (**a**) from DCGAN; (**b**) from DDPM.

For the defect detection task, the detection results of YOLOv8-det on the augmented dataset generated by DCGAN and the DDPM are shown in Figures 4a and 4b, respectively. The box mAP result for the dataset augmented by DCGAN and the DDPM are shown in Table 2. The DDPM demonstrates its superior performance with the highest box mAP of 95.8%. In contrast, the DCGAN model exhibits a collapse phenomenon in the object detection task, with a box mAP of 56.1% at a ratio of 1-2 and 61.5% at a ratio of 1-3. Although the DDPM experiences a slight decrease in performance, its values of box mAP consistently remain higher than those of the original dataset.

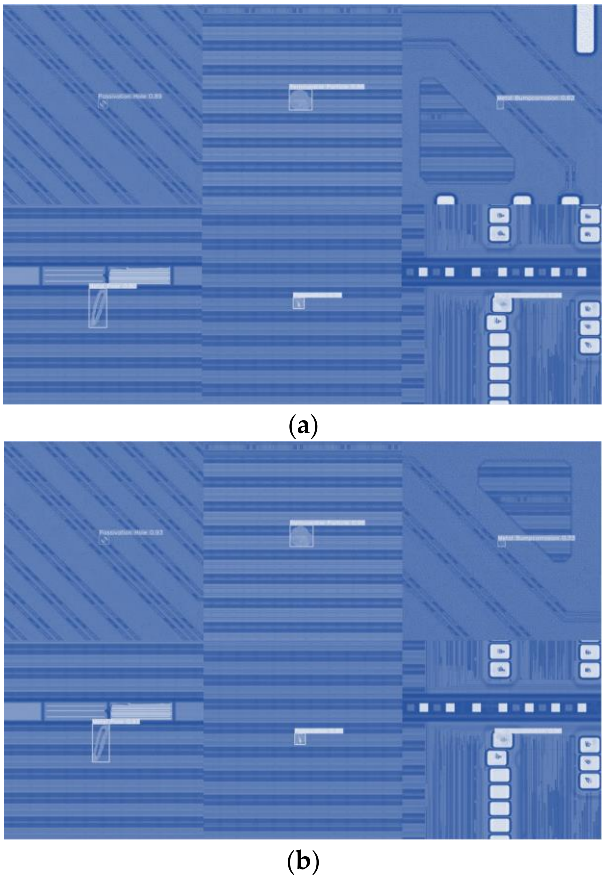


(**a**)

(**b**)

Figure 4. Defect detection results: (**a**) from DCGAN; (**b**) from DDPM.

For the defect segmentation task, the result of YOLOv8-seg on the augmented dataset generated by DCGAN and the DDPM are shown in Figures 5a and 5b, respectively. From the mask mAP prediction results of augmented dataset generated by DCGAN and the DDPM in Table 2, we observed that generative models significantly enhanced segmentation performance compared with the original dataset. The DDPM consistently outperformed DCGAN across all levels, achieving the highest mask mAP of 95.7% at a ratio of 1-2. This indicates the DDPM's robustness and superior predictive performance in object segmentation tasks. Conversely, DCGAN experienced a collapse at a ratio of 1-3, with its mask mAP dropping to 59.7%.



(**a**)

(**b**)

Figure 5. Defect segmentation results: (**a**) from DCGAN; (**b**) from DDPM.

5. Conclusions

Recognizing the unique challenges in the IC industry due to the scarcity of wafer defect pattern samples, this research involves using a DDPM to generate defective images of wafer datasets for defect inspection experiments. By utilizing an augmented dataset of real and generated images, the inspection performance improved in the tasks of defect classification, detection, and segmentation, tested with YOLOv8. The augmented dataset outperformed previous benchmarks when tested with YOLOv8, achieving an accuracy of 98.7% for YOLOv8-cls, a box-mAP of 95.8% for YOLOv8-det, and a mask-mAP of 95.7% for YOLOv8-seg. The results demonstrate that images generated by the DDPM can effectively enrich wafer defect datasets and enhance wafer defect inspection performance in the real world. The results also demonstrate that the GAN may experience collapse, where the generator produces limited variations of samples, ignoring the full diversity of the training data.

In this study, we used generative models for wafer defect inspection without hyperparameter tuning, relying on default settings. This property shows the DDPM's ability to generate realistic defect samples without major adjustments, making it practical for industrial use. The model's out-of-the-box performance is especially useful for quick deployment or when resources are limited.

Author Contributions: Conceptualization, P.-H.W. and S.-H.C.; methodology, P.-H.W., Y.-T.C., Y.-W.L., C.-H.K., Y.-C.C., S.-Z.L. and A.P.M.; software, Y.-T.C., Y.-W.L., C.-H.K., Y.-C.C. and A.P.M.; validation, P.-H.W. and S.-H.C.; formal analysis, T.P.H., S.-Z.L. and S.-H.C.; data curation, P.-H.W.; writing—original draft preparation, T.P.H. and S.-H.C.; writing—review and editing, S.-Z.L., S.-H.C. and A.P.M.; supervision, P.-H.W.; project administration, S.-H.C.; funding acquisition, S.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council (NSTC), grant number 113-2221-E-131-031-, and the APC was funded by Ming Chi University of Technology.

Data Availability Statement: The data used in this study contains commercial privacy and must not be disclosed.

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

References

1. Kim, T.; Behdinan, K. Advances in machine learning and deep learning applications towards wafer map defect recognition and classification: A review. *J. Intell. Manuf.* **2023**, *34*, 3215–3247. [CrossRef]
2. Chien, J.C.; Wu, M.T.; Lee, J.D. Inspection and classification of semiconductor wafer surface defects using CNN deep learning networks. *Appl. Sci.* **2020**, *10*, 5340. [CrossRef]
3. Niu, S.; Li, B.; Wang, X.; Lin, H. Defect image sample generation with GAN for improving defect recognition. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1611–1622. [CrossRef]
4. Ma, J.; Zhang, T.; Yang, C.; Cao, Y.; Xie, L.; Tian, H.; Li, X. Review of wafer surface defect detection methods. *Electronics* **2023**, *12*, 1787. [CrossRef]
5. Chen, S.H.; Kang, C.H.; Perng, D.B. Detecting and measuring defects in wafer die using gan and yolov3. *Appl. Sci.* **2020**, *10*, 8725. [CrossRef]
6. Megahed, F.; Camelio, J. Real-time fault detection in manufacturing environments using face recognition techniques. *J. Intell. Manuf.* **2012**, *23*, 393–408. [CrossRef]
7. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
8. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2014**, *63*, 139–144. [CrossRef]
9. Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; Yang, M.H. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.* **2023**, *56*, 1–39. [CrossRef]
10. Dhariwal, P.; Nichol, A. Diffusion models beat GANs on image synthesis. *Adv. Neural Inf. Process. Syst.* **2021**, *11*, 8780–8794.
11. Gao, H.; Zhang, Y.; Lv, W.; Yin, J.; Qasim, T.; Wang, D. A deep convolutional generative adversarial networks-based method for defect detection in small sample industrial parts images. *Appl. Sci.* **2022**, *12*, 6569. [CrossRef]
12. He, X.; Luo, Z.; Li, Q.; Chen, H.; Li, F. DG-GAN: A high quality defect image generation method for defect detection. *Sensors* **2023**, *23*, 5922. [CrossRef] [PubMed]
13. Chen, S.H.; Lai, Y.W.; Kuo, C.L.; Lo, C.Y.; Lin, Y.S.; Lin, Y.R.; Kang, C.H.; Tsai, C.C. A surface defect detection system for golden diamond pineapple based on CycleGAN and YOLOv4. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 8041–8053. [CrossRef]
14. Zhang, H.; Pan, D.; Liu, J.; Jiang, Z. A novel MAS-GAN-based data synthesis method for object surface defect detection. *Neurocomputing* **2022**, *499*, 106–114. [CrossRef]
15. Hao, D.; Yaermaimaiti, Y. Improved DCGAN for solar cell defect enhancement. *Distrib. Gener. Altern. Energy J.* **2023**, *38*, 1383–1401. [CrossRef]
16. Tang, S.; Jin, Z.; Zhang, Y.; Lu, J.; Li, H.; Yang, J. A timestep-adaptive-diffusion-model-oriented unsupervised detection method for fabric surface defects. *Processes* **2023**, *11*, 2615. [CrossRef]
17. Liu, W.; Liu, C.; Liu, Q.; Yu, D. Assigned MURA defect generation based on diffusion model. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Vancouver, BC, Canada, 17–24 June 2023; pp. 4395–4402.
18. Semchyshyn, P. Automated Visual Inspection in the Industrial Setup via Deep Learning. Bachelor's Thesis, Department of Computer Sciences and Information Technologies, Ukrainian Catholic University, Lviv, Ukraine, 2023.
19. Huang, T.; Gao, Y.; Li, Z.; Hu, Y.; Xuan, F. A hybrid deep learning framework based on diffusion model and deep residual neural network for defect detection in composite plates. *Appl. Sci.* **2023**, *13*, 5843. [CrossRef]
20. DeRidder, V.; Dey, B.; Halder, S.; VanWaeyenberge, B. SEMI-DiffusionInst: A diffusion model based approach for semiconductor defect classification and segmentation. In Proceedings of the 2023 International Symposium ELMAR, Zadar, Croatia, 11–13 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 61–66.
21. Wu, P.H.; Lin, S.Z.; Chang, Y.T.; Lai, Y.W.; Chen, S.H. A self-training-based system for die defect classification. *Mathematics* **2024**, *12*, 2415. [CrossRef]
22. 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. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
23. Chong, M.J.; Forsyth, D. Effectively unbiased fid and inception score and where to find them. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6070–6079.
24. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.

25. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
26. Ferdowsi, A.; Saad, W. Brainstorming generative adversarial network (BGAN): Towards multiagent generative models with distributed data sets. *IEEE Internet Things J.* **2024**, *11*, 7828–7840. [CrossRef]

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.