machines

MDPI

Article

A Novel FS-GAN-Based Anomaly Detection Approach for Smart Manufacturing

Tae-yong Kim, Jieun Lee, Seokhyun Gong, Jaehoon Lim, Dowan Kim and Jongpil Jeong *

Department of Smart Factory Convergence, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea; skywin94@naver.com (T.-y.K.); pungddang@naver.com (J.L.)
* Correspondence: jpjeong@skku.edu; Tel.: +82-10-9700-6284 or +82-31-299-4267

Abstract: In this study, we present the few-shot generative adversarial network (FS-GAN) model, which integrates few-shot learning and a generative adversarial network with an unsupervised learning approach (AnoGAN) to address the challenges of anomaly detection in smart-factory manufacturing environments. Manufacturing processes often encounter malfunctions or defective parts that disrupt production and compromise product quality. However, collecting and labeling sufficient data to detect anomalies is time-intensive, and abnormal data are rare, leading to data imbalances. The FS-GAN model leverages few-shot learning to enable accurate predictions with minimal data and uses the generative capabilities of AnoGAN to mitigate the scarcity of abnormal data by generating synthetic normal data. Experimental results demonstrate that FS-GAN outperforms existing models in terms of accuracy and learning speed, even with limited datasets, effectively addressing the data imbalance problem inherent in manufacturing. The model reduces dependency on extensive data collection and labeling efforts, making it suitable for real-world applications. Through reliable and efficient anomaly detection, FS-GAN contributes to production reliability, product quality, and operational efficiency in smart factories. This study highlights the potential of FS-GAN to provide a cost-effective and high-performance solution to the challenges of anomaly detection in the manufacturing industry.

Keywords: machine vision; deep learning; few-shot learning; GAN; anomaly detection

Academic Editor: Jiangjian Xiao

Received: 26 November 2024
Revised: 23 December 2024
Accepted: 30 December 2024
Published: 31 December 2024

Citation: Kim, T.-y.; Lee, J.; Gong, S.; Lim, J.; Kim, D.; Jeong, J. A Novel FS-GAN-Based Anomaly Detection Approach for Smart Manufacturing. *Machines* **2025**, *13*, 21. https://doi.org/10.3390/machines13010021

1. Introduction

As the complexity and automation speed of the manufacturing industry continues to grow, anomaly detection is emerging as an increasingly important task [1]. Anomaly detection involves the collection of data from the manufacturing environment and the detection of malfunctions in the production line or defective parts. This can help maintain product quality and prevent production interruptions. However, collecting and labeling sufficient data from a manufacturing environment is time-consuming and expensive [2]. Since normal data are abundant and abnormal data are scarce, successful application of anomaly detection in manufacturing environments is limited. In addition, consistently labeling data for anomaly detection is inefficient, and it is difficult to obtain labels that distinguish between normal and abnormal entities in data used for learning and verification [3].

Numerous research projects are underway to improve the accuracy and efficiency of anomaly detection, and various technologies used for anomaly detection are being introduced to increase production efficiency and reduce costs while utilizing limited data.

Anomaly detection in smart factories faces several critical challenges. In a typical manufacturing environment, abnormal samples often represent less than 5% of the total dataset,

leading to significant data imbalance [4]. For instance, in a study involving surface defect detection, only 52 out of 399 images were labeled as defective, highlighting the scarcity of abnormal data. This imbalance poses a critical challenge for traditional anomaly detection models, which often rely heavily on large and balanced datasets for training. Additionally, manually labeling abnormal data in manufacturing settings is not only time-intensive but also requires domain expertise, significantly increasing operational costs. For example, labeling 1000 abnormal samples in a high-speed production line environment may take several weeks and require skilled engineers, making it impractical for real-time applications. Furthermore, the lack of real-time anomaly detection systems in smart factories often leads to production delays, quality degradation, and increased costs due to undetected defects propagating through the production line. The FS-GAN model directly addresses these challenges by combining few-shot learning and AnoGAN. Few-shot learning minimizes the need for large datasets by enabling predictions with minimal data, while AnoGAN generates high-quality normal data to supplement the scarcity of abnormal samples. This integration ensures robust anomaly detection even in highly imbalanced and data-scarce manufacturing environments, reducing both the time and cost associated with data labeling and collection.

Smart factories effectively use digital technology in modern manufacturing. In an automated factory, product defects and production-line malfunctions can be fatal problems, making an efficient detection of abnormalities essential [5]. Various abnormality detection models have been proposed to address the problems posed by insufficient data and labeling. For instance, one recent study applied the ResNet [6] and Skip-GANomaly [7] models separately [8], highlighting that using a single model is insufficient for efficient anomaly detection. While these models demonstrated strengths in specific scenarios, they lacked generalization capabilities, particularly in data-scarce manufacturing environments. The study confirmed that combining models with complementary strengths produced results superior to those of individual models. Another recent study explored the utility of a few-shot learning model for detecting insulator abnormalities [9]. While effective in handling data scarcity, this approach showed limitations in detection speed due to its complex structure, making it less suitable for real-time manufacturing applications. Drawing on a method proposed to improve accuracy by increasing the support set [10], we integrate a generative adversarial network (GAN) [11] into few-shot learning. This combination addresses the support-set shortage inherent in few-shot learning while enhancing the overall detection speed and accuracy. The FS-GAN model leverages the generative capabilities of GANs to supplement the support set with high-quality normal data. By doing so, it not only mitigates the impact of data imbalance but also ensures faster anomaly detection, overcoming the bottlenecks of traditional few-shot learning models. This novel integration allows FS-GAN to achieve high anomaly detection performance even in real-time scenarios, providing significant improvements over standalone models like ResNet, Skip-GANomaly, or traditional few-shot learning approaches.

This study was designed to address the problems posed by existing anomaly detection models in actual manufacturing sites. Although research on anomaly detection models is actively being conducted, a more efficient and economical anomaly detection model is needed as the number of smart factories is growing rapidly. We therefore propose the FS-GAN model, a new anomaly detection model that combines few-shot learning with an anomaly-detecting generative adversarial network (AnoGAN) [12]. Few-shot learning can make predictions with even a small amount of data, but because the anomaly detection time is limited, we added AnoGAN to increase the support set.

The FS-GAN model has distinct features that were not suggested in previous studies, and it is expected to comprehensively solve problems introduced by data shortage and

imbalance, as well as labeling problems. Few-shot learning can make predictions through comparisons when new data to be predicted are given in addition to a small amount of training data. As a GAN model, AnoGAN is actively applied to anomaly detection. It is based on a GAN model with data-generation abilities and the ability to generate normal images. In addition, AnoGAN can solve the labeling problem using semi-supervised learning, which can save time and costs involved in labeling. This model can also be used in the medical, industrial, and agricultural fields, where the amount of abnormal data is significantly smaller than that of normal data [13].

Anomaly detection using the FS-GAN model provides many advantages in smart factories. First, it can rapidly detect defective products and improve product quality. If abnormal signs in a product are detected in real time, the product can be automatically discarded. Second, rapid and accurate anomaly detection is possible using only limited amounts of abnormal data. This can reduce the time and costs required to collect a large amount of data. Third, data can be processed more efficiently through automation and optimization. Fourth, because it can collect and analyze data in real time, immediate measures can be taken. This paper applies FS-GAN to anomaly detection in an attempt to overcome the data limitations of traditional methods of enabling rapid and efficient anomaly detection and improve the stability and economy of the manufacturing industry [14].

Our approach to idea-generation process involves the following elements:

- We identified the problems of anomaly detection technology in the manufacturing field. We recognized that collecting a large amount of data in the manufacturing field incurs significant costs, and the collected data contain significantly fewer abnormal data, resulting in data imbalance [15]. In addition, it is important to develop a model that can solve various realistic problems such as those posed by data-labeling and real-time monitoring for anomaly detection technology to be applied to actual manufacturing sites.
- We investigated previous studies. Currently, studies of anomaly detection in smart factories focus on improving efficiency and performance [16]. Based on previous studies, we combined two complementary models to solve complex problems that anomaly detection technology faces in actual manufacturing settings.
- We conducted an experiment using the surface crack presence dataset [17], which was collected from an actual smart factory and contains data with surface defects. Our goal was to generate high-quality synthetic data and identify the differences between synthetic and actual data to accurately detect defective data.
- We considered the practicality of the study. Given its applicability to smart factories, we configured a system that detects abnormal signs and warns of preventive maintenance in advance to minimize production downtime [18]. Through this idea-generation process, we proposed a model that can detect anomalies rapidly and with a small amount of data by combining few-shot learning and AnoGAN models.

Anomaly detection is a critical task in smart factories, ensuring uninterrupted production and maintaining product quality. Without an efficient anomaly detection system, even minor issues during manufacturing can escalate into significant problems, leading to increased costs, decreased quality, and safety risks [19]. Existing anomaly detection models often struggle in real-world manufacturing environments due to the lack of sufficient labeled data [20]. To address these challenges, this study introduces the FS-GAN model, which combines few-shot learning with GAN technology—a novel integration aimed at overcoming the limitations of existing approaches and providing an efficient anomaly detection solution.

- Solving the data shortage problem: The FS-GAN model integrates few-shot learning and AnoGAN, leveraging their respective strengths. Few-shot learning enables accurate predictions even with minimal data, while AnoGAN generates high-quality normal data, addressing the imbalance caused by scarce abnormal data. Unlike previous GAN-based models that rely heavily on large datasets, FS-GAN demonstrates robust performance with limited training data, making it especially practical for manufacturing environments where abnormal data are inherently rare.
- Fundamental differences from existing GAN-Based approches: Existing GAN-based anomaly detection models, such as AnoGAN or BiGAN, primarily focus on utilizing normal data to learn generative patterns. However, FS-GAN advances beyond these models by incorporating few-shot learning to optimize performance in data-scarce conditions. This dual approach not only improves learning efficiency but also significantly enhances real-time detection capabilities, allowing manufacturers to respond rapidly to anomalies without requiring extensive labeled datasets.
- Improving anomaly detection performance: The FS-GAN outperforms traditional models in both accuracy and speed. Its novel architecture allows for quick detection and the classification of anomalies, reducing the time and cost typically associated with manual labeling and large-scale data collection. Furthermore, FS-GAN's ability to generate synthetic normal data ensures a more balanced and comprehensive training process, enabling robust detection of subtle and complex anomalies often overlooked by other methods.

This paper is organized as follows: Section 2 describes related research on few-shot learning and AnoGAN. Section 3 explains FS-GAN. Section 4 evaluates the performance of the model proposed in Section 3. Section 5 draws conclusions about the contributions of this study and future research directions.

2. Related Work

2.1. Machine Vision

Figure 1 depicts the machine vision process. Deep learning models for quality classification have been used in industrial inspections for many years. In the manufacturing industry, machine vision plays an important role in optimizing automation and quality control and reducing the defect rate by detecting defects early. Machine vision can analyze images and videos in real time [21], automatically detecting various errors and defects that may occur during production. These systems increase the productivity and reliability of the manufacturing industry by rapidly classifying defective products, such as small defects on the product surface, errors that occur during the assembly process, or abnormal conditions under various environmental conditions. This not only produces high-quality products, but also increases the efficiency of the entire manufacturing process and reduces unnecessary costs [22]. In a smart factory, machines perform repetitive and precise inspection tasks that are difficult for humans to perform. To maintain a high level of accuracy and reliability and optimize productivity, it is important to continuously monitor the production process at each stage and immediately detect defects. In such an environment, machine vision goes beyond simple monitoring, automatically recognizes abnormal conditions, and supports rapid responses by detecting problems before they occur and alerting supervisors through an early warning system. Various object-detection and classification models are being introduced to machine vision systems by combining with them with an artificial intelligence technology known as deep learning. Such models are essential for detecting surface defects or assembly errors in products through image-capture techniques and rapidly identifying abnormal conditions in production lines [23]. Smart-factory machine vision in manufactur-

ing environments is largely classified into deep learning-based approaches and traditional computer vision.

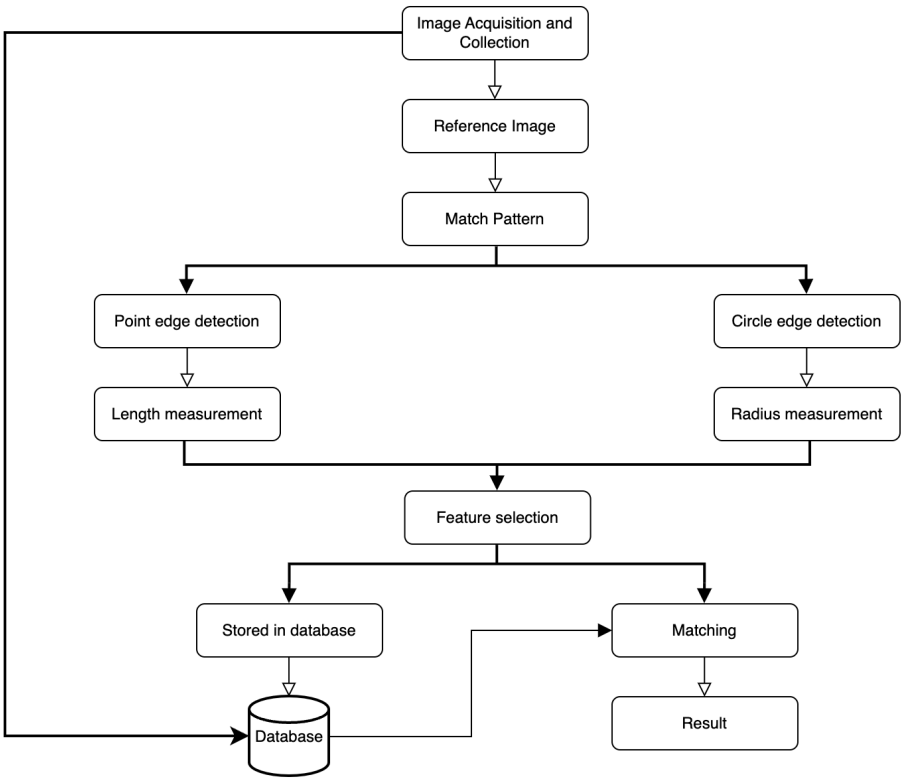

Figure 1. Machine vision process.

- Deep learning-based approaches
  Most machine vision systems use deep learning-based algorithms in manufacturing environments that require high accuracy and precision. Deep learning algorithms have achieved outstanding performance results in recognizing and distinguishing various objects in images or videos and are useful for complex and repetitive defect-detection tasks. Representative algorithms include You Only Look Once (YOLO) [24], fast region-based convolutional neural networks (R-CNNs) [25], and Mask R-CNNs [26]. YOLO has the advantage of analyzing entire images at once and detecting objects at high speed, making it suitable for environments that require real-time detection. Fast R-CNN and Mask R-CNN can both determine the location of objects with a high degree of accuracy. Mask R-CNN is advantageous for detailed defect detection because it has an image-segmentation function that separates the area of each object in the image. These algorithms show acceptable performance in the fields of object detection and image segmentation, and they are often used for defect detection in manufacturing [27]. Another deep learning-based approach is GAN, which can be used to detect abnormal data after learning normal data. Models such as AnoGAN can effectively detect defects or abnormalities by learning features from normal data and analyzing the differences with abnormal data. Since the defect rate in manufacturing is low and more normal data are available compared with abnormal data, GAN-based approaches help solve the data imbalance problem [28]. These technologies can improve the quality of manufactured products, reduce costs, and increase efficiency.
- Traditional computer vision techniques
  Computer vision before the development of deep learning models was used primarily for simple defect detection and quality inspection. These techniques require fewer computational resources and can be applied quickly. Representative traditional tech-

niques include edge detection [29] and pattern matching [30]. The former detects the boundaries of specific objects or areas in an image to find defects. It can be used to identify simple defects that occur on the surface of a product, such as scratches or surface damage. The latter finds specific patterns in an image to determine whether there are defects. It is effective when the shape of the product is constant and the background is simple. However, traditional techniques are relatively inaccurate compared with deep learning-based models and have trouble responding to complex defects or changes in the background [31].

For this reason, deep learning is being adopted to increase the efficiency of defect detection, which leads to improved productivity and lower costs. In this study, we propose FS-GAN using a deep learning-based approach.

2.2. Few-Shot Learning

Few-shot learning is a model that performs predictions with a small amount of data in preparation for cases where overfitting occurs due to insufficient training data. This model performs predictions with only a small amount of fast training when new training data other than the presented data are supplied by learning the similarity between classes [13].

There is a difference in data classes between few-shot learning and general neural networks. In general neural networks, the data classes of the training and test sets are the same. However, in the case of few-shot learning, the data classes of the two sets are different. Few-shot learning predicts new data classes that are not in the training set [32]. Few-shot learning classifies an unlabeled query set into a new class by using a small number of labeled support sets [33]. Here, the support set is synonymous with the training dataset, and the query set is equivalent to the test set. When there are C classes in a given learning dataset, and each class has K data, the model learns how to recognize C classes from C × K data, which is called C-way K-shot learning [13]. As the number of C classes increases, accuracy decreases, and as the number of K increases, the model can compare images and improve performance, which increases model accuracy.

The number of shots determines the machine learning methodology.

- Few-shot learning
  Few-shot learning makes inferences by looking at only a small amount of data, similar to human learning. Research is being conducted on architectures that learn small amounts of data and perform accurate predictions [34].
- One-shot learning
  One-shot learning makes inferences by looking at only a single piece of data. If there is only one piece of data with supervised information for a class, it is called one-shot learning [35].
- Zero-shot learning
  Zero-shot learning makes inferences by looking at 0 pieces of data. If there are no data corresponding to a class, it is called zero-shot learning. If there are no data with supervised information in the class, other forms of information, such as features, are required to learn the supervised data [35].

Few-shot learning can also be divided into four main approaches.

- Metric-based
  The metric-based approach uses a nonparametric classifier that can model the distribution of the distances between data without optimizing parameters, such that the distance between data in the same class is small and the distance between data in different classes is large [36]. Test data are compared and separated when given. Recently,

many few-shot models, such as prototype networks [37], matching networks [38], and Siamese networks [39], have used the metric-based approach.

- Model-based
  The model-based approach adopts an appropriate model structure to quickly update parameters from a small number of labeled datapoints and finish parameter iteration [32].
- Optimized-based
  The optimized-based approach is centered on meta-learning. If a model learns a series of meta-classifiers with a large amount of data, it will need a few new data to learn and fine-tune parameters to achieve superior classification performance when encountering a new task [31].

Few-shot learning was chosen for this study due the small amount of learning data required and cost issues [40]. When a general machine learning model is used in an environment with insufficient data, there is a high possibility of overfitting. However, few-shot learning addresses overfitting by beginning with a small amount of data. Model-based few-shot learning is also thought to be an appropriate approach to anomaly detection in the manufacturing industry when normal and abnormal data are dichotomous.

2.3. GAN

Generative adversarial networks emerged to address the limitations of supervised anomaly detection. While supervised anomaly detection relies on data with labels, such labels do not exist in the real world, and even if data are preprocessed and classified into labels, complex problems such as insufficient data, mistakes in the labeling process, and data imbalance, are likely to occur. A GAN can solve complex problems by learning real data through a model that generates fake images [41]. A GAN is composed largely of a generator and a discriminator. The generator model (G) learns real data and then generates data that are almost similar to the learned data but do not exist in reality. The discriminator model (D) compares the data generated by the generator model with the real data. Repeating G and D output improves the performance of the model. When the performance is maximized, the model is said to have stabilized [41].

Several GAN models have been proposed to improve the performance of GANs.

- Deconvolution GAN
  A deconvolution GAN is a model designed to address the learning stability problem associated with GANs. It uses the same synthetic structure as a GAN, but can generate a generator using deconvolution. It uses a network structure with a convolution that considers image location information to achieve more stable learning [41].
- ADGAN
  Anomaly detection generative adversarial networks (ADGANs) consist of an improvement and deterioration unit, a generation unit, and a detection unit. The improvement and deterioration unit consists of a perceptual variational autoencoder, and the generation unit and detection unit perform adversarial learning to deceive each other. ADGANs offer the advantage of combining a perceptual variational autoencoder and an adversarial generative neural network to achieve optimal performance in anomaly detection [42].
- BiGAN
  The BiGAN is an extended version of a GAN with an encoder that maps the input to the latent space. It learns the encoder part E, which maps the given input value x to z, the generator G, and the discriminator D. This process can reduce computational costs during testing. Unlike general GAN models that only consider inputs, a BiGAN's discriminator also considers latent expressions [43].

- WassersteinGAN
  A WassersteinGAN (WGAN) introduces weight clipping to smoothly provide gradients that resolve the instability caused by large weight distributions within the model layer. A WGAN trains the generator and discriminator using the Wasserstein distance, which measures the distance between the actual image and the generated image and calculates how close it is to the actual data [44].
- AnoGAN
  The AnoGAN model learns normal images using a DCGAN and generates normal images. The parameters of the G and D functions of DCGAN are fixed, and the latent space, z, is randomly extracted to find its optimal value. The total loss is then calculated by comparing the generated normal image with the actual image. The total loss is the weighted sum of residual loss (the difference between the generated normal image and the actual image) and discrimination loss (the difference between the D value of the generated normal image and the D value of the actual image). Learning is completed when the total loss is minimized while adjusting z. Total loss is also called the anomaly score. The AnoGAN model can reduce the data imbalance and labeling problems that appear in supervised anomaly detection [10].

AnoGAN was chosen for this study over several other GAN models because it was expected to solve the problem of insufficient data in anomaly detection by generating normal images during the learning process. In addition, AnoGAN adopted a semi-supervised anomaly detection learning method, which made it comparable to the learning method selected in this study. Given the complexity, performance, and scalability of the implementation, we determined that this model was the most suitable option. In addition, AnoGAN was used instead of ADGAN because, although both models learn from normal data and detect abnormal patterns, ADGAN may be limited in detecting abnormalities in certain types of data or under certain conditions. AnoGAN is optimized for detecting more complex and subtle abnormal patterns that those appropriate for ADGAN. Few-shot learning means learning from limited data, and AnoGAN has achieved excellent results even when learning with a limited amount of unbalanced data.

2.4. Anomaly Detection

Anomaly detection is a basic task that determines whether test data conform to a normal data distribution. Depending on the application field, if the test data are inappropriate, it is called an anomaly, failure, or contamination. Anomaly detection has been studied in statistics since the 19th century, but many unresolved problems remain [45]. In particular, anomaly detection is actively used in the industrial field. Since anomalies in industrial systems can cause economic losses and damage reputations, it is essential to detect and repair them early. To detect such damage, various machine learning techniques are used in anomaly detection [46].

Anomaly detection is classified into three categories (Table 1) depending on whether abnormal data are used and whether a label is present.

Table 1. Classification of anomalydetection based on whether abnormal data are used and with or without labels.

	Normal Data	**Abnormal Data**
Supervised anomaly detection	O	O
Semi-supervised anomaly detection	O	X
Unsupervised anomaly detection	Label exists	

- Supervised anomaly detection
  Supervised anomaly detection learns using normal and abnormal data with labels, and new data are automatically classified as normal or abnormal by analyzing them with a classifier. It is more accurate than other anomaly detection methods and is used in a wide range of fields, such as Bayes classifiers and neural networks. However, this method depends on whether there are training data with initial label values. As outliers are rare, it is generally difficult to obtain a sufficient amount of labeled classes [10,45].
- Semi-supervised anomaly detection
  Supervised anomaly detection requires data labeling, making it time-consuming and costly, and mistakes can occur during the labeling process. To compensate for these disadvantages, semi-supervised learning uses a small amount of labeled normal data in what is known as pseudo-labeling [47]. It can save time and costs by not collecting abnormal data, but its accuracy is lower than that of supervised anomaly detection [10].
- Unsupervised anomaly detection
  Unsupervised anomaly detection uses unlabeled data, and is widely used in industrial systems due to its ability to process a large amount of data. Principal component analysis [48] and partial least squares [49] are representative examples. However, this method is only effective for data with low correlation, and the data must follow a multivariate Gaussian distribution. It shows the poorest performance among the three anomaly detection methods [10,45].

Semi-supervised anomaly detection was selected among the three anomaly detection learning methods for two reasons. First, because of the dataset used, the surface crack presence/absence dataset, is in the form of normal data with a small amount of labeling, the learning method took this into account. In addition, because the more serious problem that occurs when applying anomaly detection models in actual manufacturing industrial environments involves data labeling, semi-supervised anomaly detection was thought to be the most suitable solution. Semi-supervised anomaly detection can be largely divided into models using autoencoders and models using GANs. This uses FS-GAN, which combines few-shot learning with the AnoGAN model [10].

3. FS-GAN Framework

3.1. System Architecture

FS-GAN is an algorithm that detects anomalies and generates normal data using a limited number of normal and abnormal data through few-shot learning. It trains AnoGAN, which was used for existing anomaly detection, with few-shot learning to detect anomalies by analyzing the difference between the generated image and the original image. The model structure of FS-GAN is divided into a learning part that trains AnoGAN with only a small amount of normal data to generate images close to normal images, an inference component that detects anomalies by obtaining the anomaly score of the test data using the previously trained model, and a generation component that trains the model using both normal and abnormal data to generate normal data. The overall structure of FS-GAN is as shown in Figure 2.

3.1.1. Training Generator And Discriminator

In the learning phase, the generator and discriminator learn within the adversarial GAN framework, where they reinforce each other through adversarial learning. The generator receives a random latent vector (Z) from the latent space as input and generates normal data, while the discriminator determines whether the given data are real normal data or fake data generated by the generator. Through this iterative process, the two architectures

compete, gradually improving their respective performances. The generator minimizes the difference between the original normal image and the generated normal image by learning to generate synthetic data that closely follow the characteristics of normal data. This ensures that the generator learns only the distribution of normal data, allowing it to generate data that represent this distribution during the inference phase. Conversely, the discriminator improves its ability to distinguish between real and generated data, serving as a critical feedback mechanism for the generator. To ensure the generator produces reliable normal data and avoids overfitting to a limited subset of normal images, the following strategies are employed.

- Only normal images from the Kolektor surface-defect dataset are used for training, while abnormal images are reserved for the inference process.
- To implement few-shot learning, subsets of the normal dataset are randomly selected for each training epoch. This randomized sampling prevents overfitting and enhances the generalizability of the model to new normal data distributions.

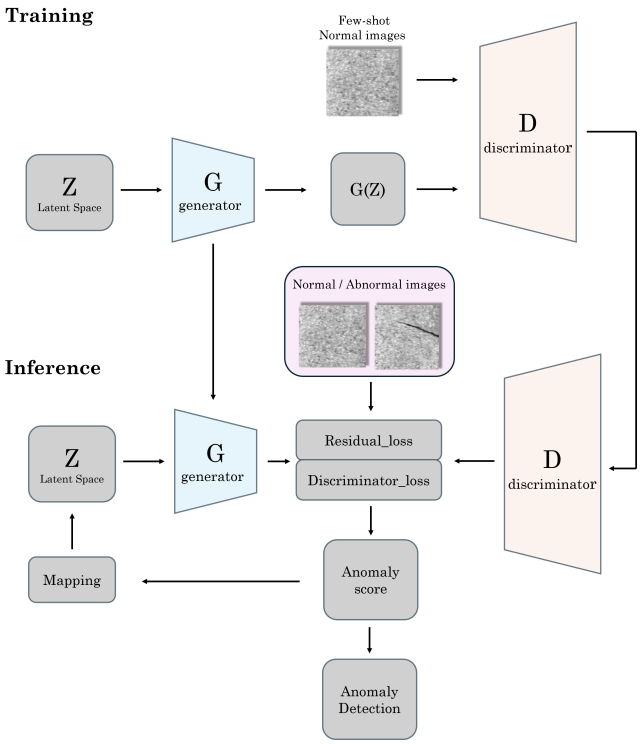

Figure 2. Architecture for FS-GAN.

During inference, the generator reconstructs the input data by mapping a latent vector (Z) to a generated image (G(Z)). An anomaly score is then calculated by comparing the generated image with the original input image using residual loss and discriminator loss. The combined anomaly score allows the model to detect whether the input deviates from the learned normal data distribution, thus identifying anomalies effectively.

By iteratively refining the generator and discriminator through adversarial learning and leveraging few-shot learning to optimize data usage, the FS-GAN model addresses the challenges posed by limited data environments while achieving reliable anomaly detection.

3.1.2. Inference

In the inference step, the generator reconstructs the input data by finding a latent vector Z that allows the generator to generate data as similar as possible to the input data. Both normal and abnormal data are used for anomaly detection, the weights of the generator and discriminator learned during the training phase are fixed, and the anomaly score is

calculated based on the residual loss and discriminator loss. In this process, the normal data label is assigned a value of 0, and the abnormal data label is assigned a value of 1 to distinguish between the two classes. The inference step proceeds as follows.

(a) Latent vector Z optimization

To determine whether the input data belong to a normal data distribution, it is necessary to first find the optimal latent vector Z so that the generator can reconstruct the data. The weights of the learned generator and discriminator are fixed, and the latent space is iteratively optimized using gradient descent. The optimization aims to minimize the difference between the input data and $G(z)$ generated from the latent vector Z. The optimal latent vector Z is found by iteratively updating Z while minimizing the residual loss, which calculates the Euclidean distance between the input data and $G(z)$, and the discriminator loss, which evaluates how much the discriminator determines the generated data to be abnormal.

$$\text{Residual Loss:} \quad \mathcal{L}_{\mathcal{R}}(z_\gamma) = \sum |x - G(z_\gamma)| \tag{1}$$

$$\text{Discriminator Loss:} \quad \mathcal{L}_{\mathcal{D}}(z_\gamma) = \sum |f(x) - f\big(G(z_\gamma)\big)| \tag{2}$$

where $\mathcal{L}_{\mathcal{R}}$ is the residual loss that calculates the difference between the input data x and $G(z_\gamma)$ generated from the latent vector Z, and $\mathcal{L}_{\mathcal{D}}$ is the discriminator loss that calculates the difference between the value determined by input data to the discriminator and the value determined by input image to the discriminator. The total loss is calculated by adding the weight λ to these two loss functions. Through the process of minimizing the total loss, the latent vector Z is optimized as a latent vector such that the generator can reconstruct the input data, and the generator can generate the output $G(z)$ most similar to the input data. The total loss and the optimization of the latent vector Z using the total loss are calculated as follows. z_t represents the z of the current stage, η, and ∇_z represent the learning rate and the weight of the loss function z_t, respectively.

$$\text{Total Loss:} \quad \mathcal{L}(z_\gamma) = (1 - \lambda) \cdot \mathcal{L}_{\mathcal{R}}(z_\gamma) + \lambda \cdot \mathcal{L}_{\mathcal{D}}(z_\gamma) \tag{3}$$

(b) Optimization proces

- Learning Rate (η = 0.0002): This learning rate, as recommended in the DCGAN paper [50], ensures stable yet efficient convergence during optimization, preventing oscillations while allowing sufficient updates to the latent vector.
- Loss Weight (λ = 0.1): The weight λ balances the contributions of the residual loss and discriminator loss. This value was chosen to prioritize pixel-level reconstruction while still considering high-level anomaly features captured by the discriminator.

$$\text{Latent Vector:} \quad z_{t+1} = z_t - \eta \cdot \nabla_z \mathcal{L}(x, G(z_t)) \tag{4}$$

(c) Calculating the anomaly score

The generated data $G(z)$ are obtained by inputting Z obtained in the latent vector optimization process into the generator. As in the previous process, the residual loss, which is calculated as the Euclidean distance between the generated data and the original data, and the discriminator loss, which evaluates the probability that the discriminator judges $G(z)$ as normal, are added to obtain the total loss. This total loss is used as the anomaly score for anomaly detection. The default weight of the anomaly score is 0.1, and the anomaly score is calculated by adding 0.9 of the residual loss and 0.1 of the discriminator loss.

$$\text{Anomaly Score:} \quad A(Z) = (1 - \lambda) \cdot \mathcal{L}_{\mathcal{R}}(Z) + \lambda \cdot \mathcal{L}_{\mathcal{D}}(Z) \tag{5}$$

(d) Anomaly detection

The calculated anomaly score is then compared with a pre-set threshold based on the distribution of normal data used for training and is usually determined by applying the average and variance of the anomaly score of normal data. If the anomaly score of the input data is less than the threshold, the data are judged to be normal, and if the anomaly score is greater than the threshold, the data are considered abnormal. Through this process, it is possible to evaluate how much the given input data match the distribution of normal data. If it is outside the normal range, the abnormal area can be judged by the difference in the anomaly score. FS-GAN learns the distribution of a small amount of normal data through the subset and uses the residual loss and discriminator loss for new inputs together to detect anomalies with the anomaly score, which is the sum of the two. It can also overcome a lack of abnormal data by generating normal data with the data-generation ability of the model. In other words, it can achieve high accuracy even in anomaly detection, which is an environment with little data.

3.2. FS-GAN Algorithm

The FS-GAN algorithm integrates few-shot learning and AnoGAN to address data imbalance and improve anomaly detection efficiency. The core components of FS-GAN include the few-shot algorithm and the anomaly score calculation algorithm. In FS-GAN, the algorithm operates as a one-way few-shot learning model, as it is trained using only normal class data, with a small subset of normal samples selected randomly for each epoch. This design allows FS-GAN to effectively generate normal data and handle data scarcity.

3.2.1. Integration with AnoGAN

AnoGAN, a generative adversarial network, complements few-shot learning by generating high-quality normal data and calculating anomaly scores during inference. While AnoGAN's discriminator uses binary classification to distinguish between normal and abnormal images, FS-GAN enhances this process by leveraging few-shot learning to refine class distinctions and improve generalization.

3.2.2. Learning and inference process

Training phase

- A subset of normal data is randomly selected to form the support set for each epoch.
- The model learns to generate normal images by minimizing the difference between input and generated images.
- Only the normal class (one-way) is used during training, ensuring the model focuses on learning normal data characteristics.

Inference phase

- Normal and abnormal data are labeled as 0 and 1, respectively.
- The anomaly score is computed by comparing the input image with the generated normal image.
- The final classification is based on the calculated anomaly score threshold.

This structured combination of few-shot learning and AnoGAN enables FS-GAN to detect anomalies efficiently, even in data-scarce environments, while minimizing computational overhead.

Here is a part of the code of FS-GAN's few-shot algorithm (Algorithm 1). By setting Subset_size to 5, only five of the existing datasets are randomly selected and used for the learning process. Additionally, five different data are randomly sampled for each epoch and learned. This addresses the problematic situation in which, if the same data are learned

every time, the model may be overfitted to specific data in few-shot learning with a small amount of learning data. As shown below, FS-GAN uses subsets to learn using only a portion of the entire data for each learning epoch, and different subset data are used for each epoch to maximize data efficiency and improve generalization performance. The anomaly score algorithm (Algorithm 2) calculates an anomaly score as shown in the model structure using residual loss and discriminator loss.

Algorithm 1 Define a pre-training subset.

```
subset_size = 5
for epoch in range(1, epochs + 1) do
    subset_indices = random.sample(range(len(trainset)), subset_size)
    subset = [trainset[idx] for idx in subset_indices]

    for real_images, _ in subset do
        real_images = real_images.unsqueeze(0).to(device)
        batch_num = 1
    end for
end for
```

Algorithm 2 Anomaly score algorithm.

```
for def residual_loss(real_images, generated_images): do
    subtract = real_images - generated_images
    return torch.sum(torch.abs(subtract))
end for

for def discriminator_loss(netD, real_images, generated_images): do
    real_features = D.forward_features(real_images)
    generated_features = D.forward_features(generated_images)
    subtract = real_features - generated_features
    return torch.sum(torch.abs(subtract))
end for
```

Looking at the code of residual loss and discriminator loss, residual loss takes real_images and generated_images as inputs and uses the difference in Euclidean distance between the two images. Discriminator loss takes not only real_images and generated_images but also a discriminator model as inputs and uses the difference between the values determined by inputting real_images and generated_images into the discriminator. Anomaly loss is calculated by adding weights to the residual loss and discriminator loss calculated in this way.

In the code of this paper (Algorithm 3), the weight l is set to 0.1. The anomaly loss obtained in this way is converted to a NumPy array and used as the anomaly score. Finally, the latent vector Z is optimized in the direction of minimizing the anomaly loss. The following is a part of the Z optimization code.

Algorithm 3 Define the anomaly loss function.

```
for def anomaly_loss(residual_loss, d_loss, l=0.1): do
    return (1 - l) * residual_loss + l * d_loss
end for
```

First, we randomly initialize the latent vector z using a standard normal distribution ($z \sim N(0,1)$) to ensure diverse and smooth exploration of the latent space (Algorithm 4). The latent vector is initialized as a learnable parameter using the

`requires_grad=True` option in PyTorch. This initialization ensures balanced and stable convergence during optimization.

Algorithm 4 Initializing latent vectors.

```
z = torch.randn(1, latent_dim, 1, 1, device=device, requires_grad=True)
optimizer_z = torch.optim.Adam([z], lr=lr)

latent_space = []
auc = []
for i, (images, _ ) in enumerate(test_loader): do
   real_images = images.to(device)
   generated_images = G(z)

   for step in range(401) do
      generated_images = G(z)
      optimizer_z.zero_grad()
      resi_loss = residual_loss(real_images, generated_images)
      disc_loss = discriminator_loss(D, real_images, generated_images)
      ano_loss = anomaly_loss(resi_loss, disc_loss, l=0.1)
      ano_loss.backward(retain_graph=True)
      optimizer_z.step()
      if step % 200 == 0 then
         loss = ano_loss.item()
         noises = torch.sum(z).item()
      end if
      if step == 400 then
         latent_space.append(z.cpu().data.numpy())
      end if
   end for
end for
```

Next, we obtain a test image and iteratively optimize z over 401 steps using the Adam optimizer with a learning rate $\eta = 0.0002$. The anomaly loss is calculated as a weighted combination of the residual loss and discriminator loss:

- Residual Loss (L_R): Measures the pixel-wise Euclidean distance between the test image and the generated image $G(z)$.
- Discriminator Loss (L_D): Evaluates the difference in the discriminator's output for the test image and the generated image.

The total anomaly loss (L) is minimized to update z iteratively. A weight $\lambda = 0.1$ balances these two losses, prioritizing pixel-level reconstruction while leveraging high-level features captured by the discriminator.

3.3. Weighting Parameter (λ) Selection

The weighting parameter λ was chosen based on empirical validation through a series of experiments to optimize the balance between residual loss and discriminator loss. Specifically:

- Testing Range: Several values of λ (0.05, 0.1, 0.2, 0.5) were evaluated to assess their impact on anomaly detection performance.
- Experimental Results:
  - Lower λ values (e.g., 0.05) overly prioritized the residual loss, leading to poor feature extraction from the discriminator and increased false negatives.
  - Higher λ values (e.g., 0.5) placed excessive emphasis on discriminator loss, resulting in reduced reconstruction accuracy and increased false positives.

- Optimal Value: $\lambda = 0.1$ achieved the best trade-off, minimizing both false positives and false negatives while maintaining robust anomaly detection performance.

The experimental results demonstrate that $\lambda = 0.1$ provides a practical and well-balanced approach for anomaly score calculation.

In the 400th update, the optimized z is added to `latent_space`, completing the optimization process. This final z represents the latent vector that best reconstructs the test image, enabling effective anomaly detection.

4. Experiment and Results

4.1. Experiment Environments

In this study, anomaly detection was performed using the FS-GAN model. The model was trained and evaluated using the Kolektor surface-defect dataset, which is widely recognized for research on surface defect detection. This dataset includes various types of defects that can occur on metal surfaces, plastics, or other manufactured surfaces. It consists of 399 images, including both normal and abnormal (defective) samples, making it suitable for training and evaluating anomaly detection models.

4.2. Reason for Choosing the Kolektor Surface-Defect Dataset

The Kolektor dataset was chosen for the following reasons:
Relevance to Real-World Scenarios:

- The dataset contains surface defect types commonly observed in industrial manufacturing, such as scratches, dents, and discoloration. These defects align closely with real-world manufacturing challenges, making the dataset highly representative of practical scenarios in industries like automotive, electronics, and materials processing.
- The inclusion of both normal and defective samples allows for comprehensive evaluation of anomaly detection models in realistic conditions.

Balance of Complexity and Size:

- While the dataset consists of only 399 images, it captures a diverse range of defect types and variations in defect severity. This balance between complexity and manageable size makes it ideal for initial experimentation and model validation without the computational overhead of larger datasets.

Benchmark Usage:

- The Kolektor dataset has been frequently used as a benchmark in academic research on surface defect detection. Its widespread adoption ensures comparability with existing studies, enabling a clearer evaluation of FS-GAN's performance relative to other models [17,51].

4.3. Dataset Splitting and Few-Shot Subset Selection

To train and evaluate FS-GAN, the dataset was split into training and validation sets using an 80:20 ratio, controlled by the `split_rate` hyperparameter. This split ensured a balanced allocation of images:

- Training Set: Comprises approximately 319 images (278 normal and 41 abnormal), providing a diverse set of examples for the model to learn from.
- Validation Set: Contains approximately 80 images (69 normal and 11 abnormal), ensuring robust evaluation of the model's performance on unseen data.

For few-shot learning, a subset of normal images is randomly selected from the training set during each epoch. The subset size was empirically validated, with experiments testing sizes of 5, 10, and 15 images per epoch. A subset size of 10 images was found to be optimal, as it

- Prevented overfitting by limiting the model's exposure to repetitive data.
- Maintained computational efficiency while exposing the model to sufficient variability in normal data.

4.4. Representativeness of Real-World Scenarios

The Kolektor surface-defect dataset effectively represents real-world industrial defect scenarios due to its

- Variety of Defect Types: It includes a range of defect types that are commonly encountered in production lines, providing a realistic testbed for anomaly detection.
- High-Resolution Images: It ensures detailed defect representation, which is crucial for detecting subtle anomalies.
- Balanced Normal/Defective Samples: It allows models to generalize well to unseen data while maintaining robust performance on rare defects.

However, we acknowledge that real-world manufacturing environments often involve larger and more diverse datasets. As part of future work, the model will be validated on additional datasets that include more complex and varied defect scenarios to further confirm its generalizability and practical applicability.

This study was conducted in the development environment shown in Table 2. High-performance computing resources were used to efficiently perform large-scale data processing and complex model learning. The GPU used to train and evaluate the model was an NVIDIA GeForce RTX 3070 Ti (Nvidia, CA, USA).

Table 2. Development environment.

	OS	CUDA	cuDNN	OpenCV	Numpy	Python	GPU
Development Environment	Windows 11 Home	v11.8	v8.24	v4.8.0	v1.26.0	v3.9.18	NVIDIA GeForce RTX 3070Ti Laptop GPU

4.5. Performance Metrics

In this paper, six scores were used as performance indicators to evaluate the performance of the FS-GAN model: accuracy, precision, recall, F1 score, area under the curve receiver operating characteristic (AUC-ROC), and log loss.

- Accuracy

  Accuracy is an indicator that measures the proportion of the total predictions inferred by the model that match the actual value. This allows us to know how accurately the model predicts.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{6}$$

  T and F represent whether the model predicted correctly (true or false), respectively, and P and N represent whether the model predicted the value as positive or negative. This creates an index that evaluates the number of correct predictions out of all predictions, as with a formula, but in an environment with imbalanced data where only a certain class of data is present, and it can overestimate performance. However, in the context of anomaly detection, where the dataset is often imbalanced (e.g., far fewer anomalies than normal samples), accuracy may not be sufficient on its own to evaluate performance effectively. For this reason, we also include precision, recall, and F1 score, which are more robust in such scenarios.

- Precision
  Precision is a measure of the proportion of actual data that the model predicted as positive that is actually positive (TP). Precision indicates how accurate the results predicted as positive are.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

- Recall
  Recall, unlike precision, is a measure of the percentage of actual data positives that the model predicts as positives. Recall indicates how often the model predicts positives without missing them.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

- F1 Score
  The F1 score is the harmonic mean of precision and recall, and is calculated using the previously calculated precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

  As it is affected by smaller values due to the product of precision and recall, this approach can effectively evaluate models even in data imbalance environments, unlike accuracy. By combining precision and recall, the F1 score is particularly useful for anomaly detection tasks, where false negatives (missed anomalies) and false positives (false alarms) must be carefully balanced.

- AUC-ROC Curve
  The AUC-ROC curve is a key metric for evaluating the performance of a model in anomaly detection tasks. The ROC (Receiver Operating Characteristic) curve visualizes the trade-off between the true positive rate (sensitivity) and false positive rate as the threshold for classification is varied. The AUC (Area Under the Curve) measures the area under this ROC curve, providing a single scalar value between 0 and 1, where values closer to 1 indicate better model performance.

- Log Loss
  Log loss is the logarithm of the probability that a model predicts the true value, and the sign is developed. It calculates the numerical loss value of the model's own misclassification. In other words, the loss increases as the predicted probability gets farther from the true class.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)) \tag{10}$$

  In the formula, N is the total number of data, y_i is the actual class, and p_i represents the model's predicted probability. Log loss evaluates how close the predicted value is to the probability; the smaller the value, the better the model performance.

These performance metrics are used to quantify and compare the anomaly detection performance of FS-GAN models. They allow us to understand the overall performance of the model by evaluating the model's accuracy, precision, recall, F1 score, AUC-ROC, log loss, and confidence score of anomaly samples. Each metric highlights a different aspect of the model, and it is important to use multiple metrics together for a comprehensive evaluation. These performance metrics are used to evaluate and compare the anomaly

detection capabilities of FS-GAN and baseline models. A detailed quantitative comparison of FS-GAN with other models using these metrics is provided in Section 4.3, Results (see Table 3 and Figures 7–12).

Table 3. Comparison with other models.

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC	Log Loss
AnoGAN	0.85	0.86	0.8	0.83	0.88	0.35
BiGAN	0.87	0.88	0.84	0.86	0.89	0.3
AutoEncoder	0.82	0.84	0.78	0.81	0.86	0.32
OC-SVM	0.75	0.76	0.7	0.73	0.77	0.45
FS-GAN	0.9	0.91	0.88	0.89	0.92	0.25

4.6. Visualization

Figure 3 shows an image generated during the model's anomaly detection process and the actual normal image. During the FS-GAN learning process, the generator learns only normal images and can generate only normal images. The learned generator and discriminator optimize the latent vector Z during the inference process, and then input the optimized latent vector Z back into the generator to output the generated normal image $G(Z)$. Anomaly detection is performed through the difference map of the generated normal image and the abnormal image input during the inference process. The generated image in Figure 3 is similar to the actual normal image. The Kolektor surface-defect dataset used includes considerable noise, making learning difficult, but FS-GAN can perform anomaly detection more effectively because the generated image is similar to the actual image, even with few-shot learning that uses only a small amount of data.

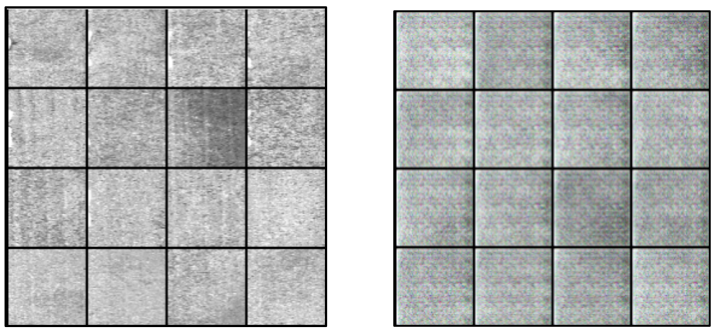

Figure 3. Generated normal image.

Figure 4 is the result of anomaly detection for normal images in the Kolektor surface-defect dataset. FS-GAN's anomaly detection is performed by comparing the first image (the actual image data) and the second image (the generated image) to obtain a difference map. However, when looking at the third image, which is the difference map of the corresponding images, it is determined that there is no difference, so anomaly detection is not performed. In addition, the model determines that the input image is a normal image, so class 0 is output. The following code is part of the data preprocessing code.

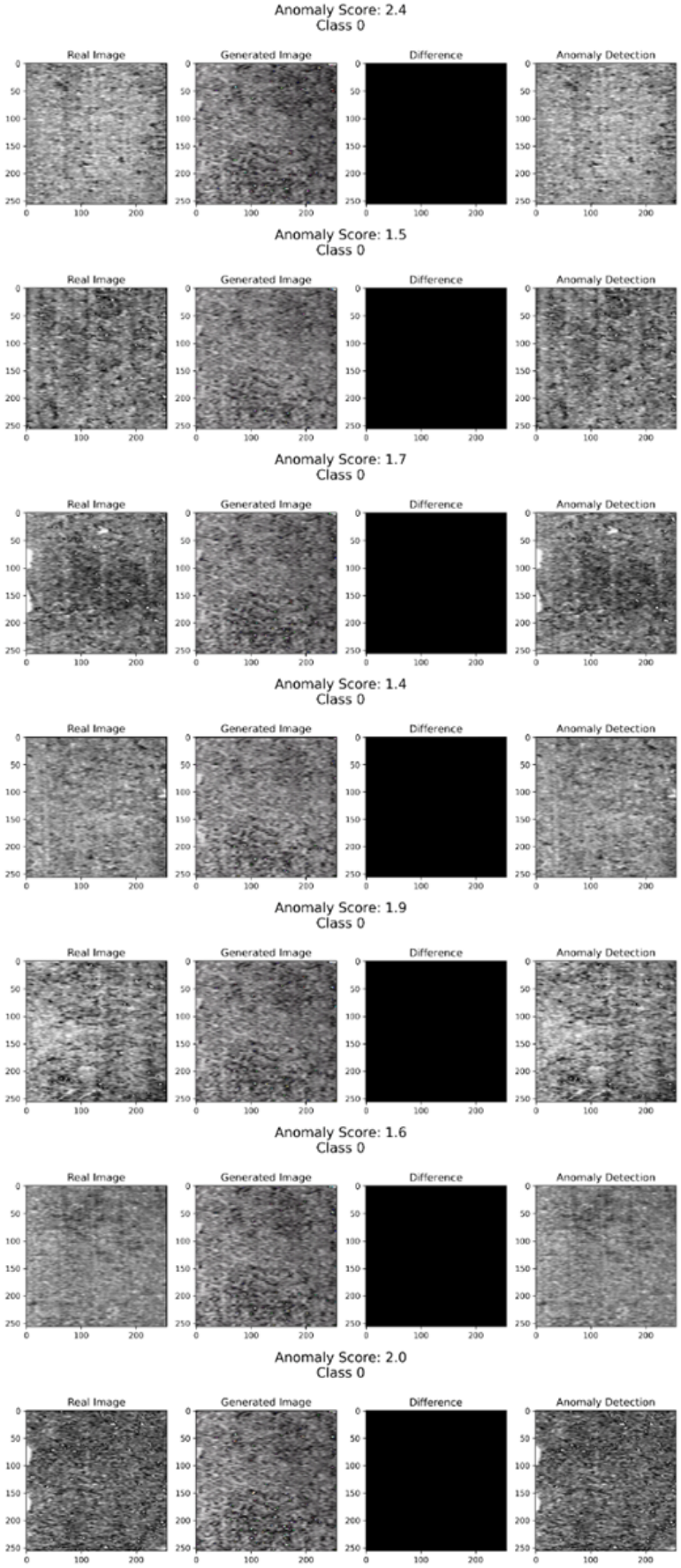

Figure 4. Anomaly detection result (normal image).

In the corresponding algorithm code (Algorithm 5), the normal image is labeled as 0 and the abnormal image is labeled as 1 during image preprocessing, such that, during the model's inference process, if the input image is determined to be normal, class 0 is output, and if it is determined to be an abnormal image, class 1 is output.

Algorithm 5 Normal, abnormal separation.

```
normal_images, normal_masks, normal_labels = load_image_and_mask
    (normal_path, normal_mask_path, label=0)
abnormal_images, abnormal_masks, abnormal_labels = load_image_and_mask
    (abnormal_path, abnormal_mask_path, label=1)
```

Figure 5 illustrates the anomaly detection process for abnormal images in the Kolektor surface-defect dataset. The first image represents the actual abnormal image, while the second image is the normal image generated by FS-GAN. The third image shows the difference map, highlighting regions where discrepancies exist between the actual and generated images. Finally, the fourth image visualizes the anomaly detection result, with the detected defect areas clearly highlighted in red. This visualization demonstrates FS-GAN's ability to accurately identify anomalies, even for small and subtle defects. The difference map and the corresponding highlighted regions validate the model's capability to distinguish between normal and abnormal regions effectively, which is particularly critical in industrial applications where small but important defects must be detected. Moreover, the model classified the input image as abnormal, outputting class 1, consistent with the actual label of the abnormal image. This highlights FS-GAN's exceptional anomaly detection capability, even when trained with limited data. By successfully detecting anomalies in a limited-data environment, FS-GAN shows comparable or superior performance to other models that rely on larger datasets.

Figure 6 visualizes the distribution of anomaly scores generated by FS-GAN for the Kolektor surface-defect dataset. The x-axis represents anomaly scores (ranging from 0 to 1), while the y-axis indicates the frequency of images at each score. The model uses a threshold of 0.3 to classify images: scores below 0.3 are considered normal, while scores above 0.3 are classified as abnormal. The graph clearly shows a distinct separation between normal and abnormal data. Normal images are clustered below the 0.3 threshold, with most scores concentrated between 0.2 and 0.25. This indicates that, although the generated image and the actual normal data do not perfectly match, the differences are minor, and the discriminator consistently classifies them as normal. Conversely, all abnormal images score above 0.3, with scores distributed across a wider range, reflecting the varying severity of defects. The abnormality score is calculated as the sum of the residual loss and discriminator loss, and this score effectively separates normal and abnormal data without error. The varying distribution of abnormal scores indicates that FS-GAN is capable of capturing diverse defect characteristics, including small cracks or subtle anomalies that are critical in industrial applications. This result confirms that FS-GAN can accurately classify data based on anomaly scores, even in noisy and diverse datasets. By effectively distinguishing normal and abnormal images at a threshold of 0.3, FS-GAN demonstrates its robustness in detecting small but significant defects, which is crucial for practical anomaly detection tasks.

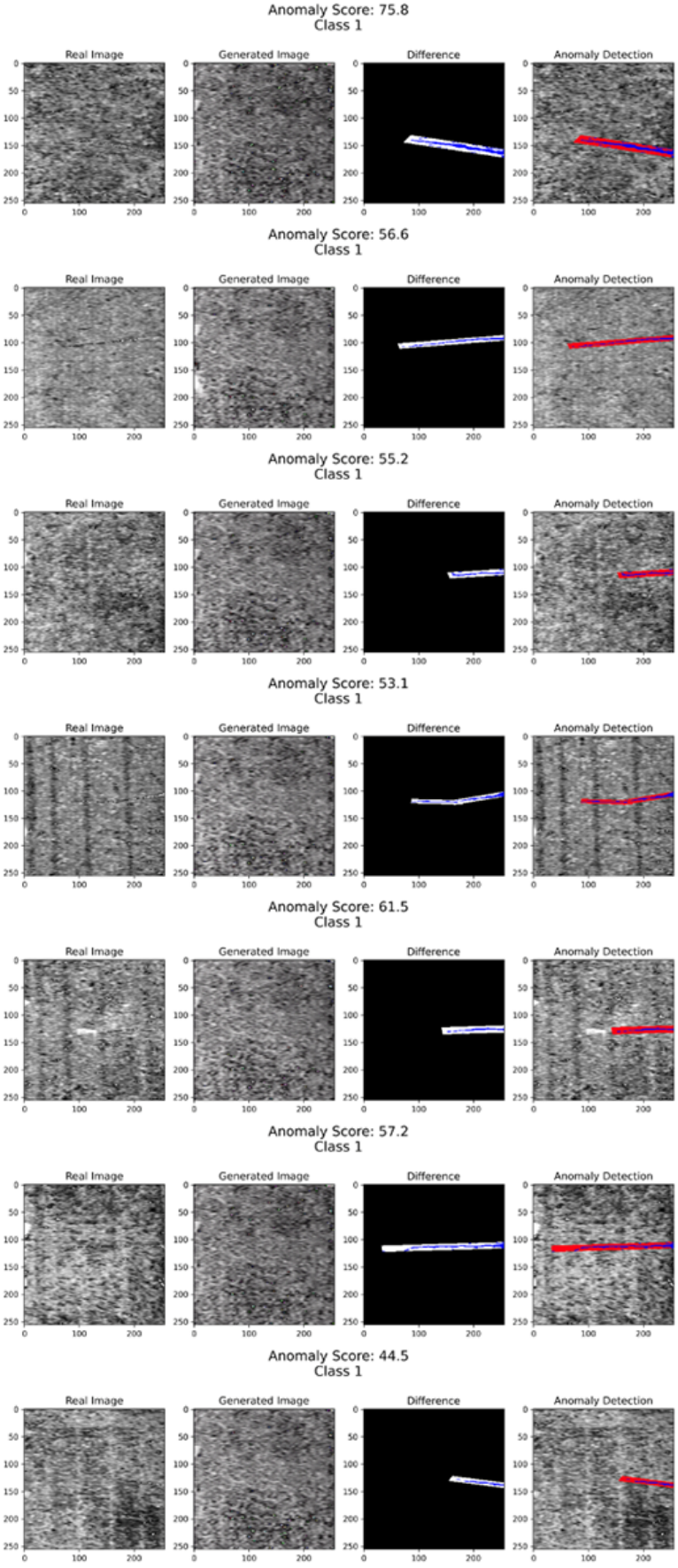

Figure 5. Anomaly detection result (abnormal image).

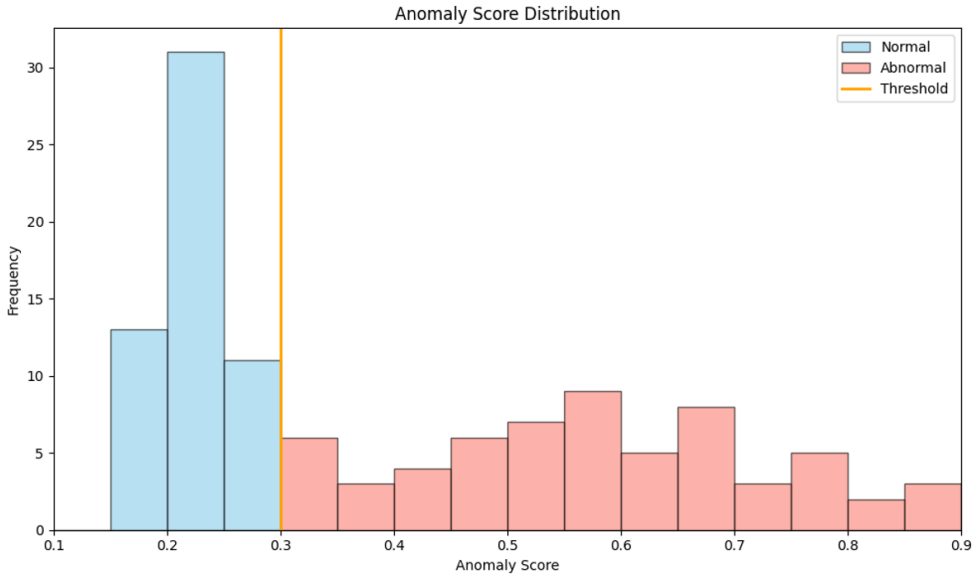

Figure 6. Distribution of anomaly scores.

4.7. Threshold of 0.3: Theoretical and Empirical Backing

The anomaly score threshold of 0.3 was determined based on both theoretical insights and empirical experiments conducted during the evaluation of FS-GAN.

Theoretical Backing: The anomaly score is calculated as the sum of the residual loss and discriminator loss, both of which are designed to capture discrepancies between the generated and actual data.

- In the case of normal data, the generator is trained to reconstruct the input accurately, leading to consistently low residual and discriminator losses. As a result, the total anomaly score for normal data remains below 0.3, reflecting the generator's ability to learn normal patterns effectively.
- For abnormal data, the generator fails to reconstruct the anomalies, resulting in significantly higher residual and discriminator losses. This natural separation in the score distributions supports the use of a threshold that can clearly distinguish normal from abnormal data.

Empirical Backing: Multiple experiments were conducted to evaluate the impact of different thresholds (e.g., 0.2, 0.3, 0.4) on FS-GAN's performance using precision, recall, and F1 score as evaluation metrics:

- A threshold of 0.3 consistently achieved the best balance between precision and recall, ensuring minimal false positives while maintaining a high true positive rate.
- As shown in Figure 6, the anomaly scores for normal images are tightly clustered below 0.3, while those for abnormal images are distributed above 0.3. This distinct separation further validates the selection of 0.3 as the optimal threshold.

Practical Implications: Setting the threshold at 0.3 allows FS-GAN to accurately classify anomalies without being overly sensitive to small variations or noise in normal data. This balance is particularly important in industrial applications, where excessive false positives can disrupt operations, and missed anomalies can have significant consequences.

4.8. Results

Table 3 compares FS-GAN with the existing model AnoGAN, and the models used for anomaly detection, BiGAN, AutoEncoder, and a one-class support vector machine (OC-SVM), using various performance indicators and the Kolektor surface-defect dataset trained on FS-GAN. Accuracy, precision, recall, F1 score, AUC-ROC, and log loss were

used as performance indicators, and the performance of the model can be comprehensively evaluated through various performance indicators. FS-GAN received higher scores than other models in all indicators. Table 3 is visualized using a graph in Figure 7.

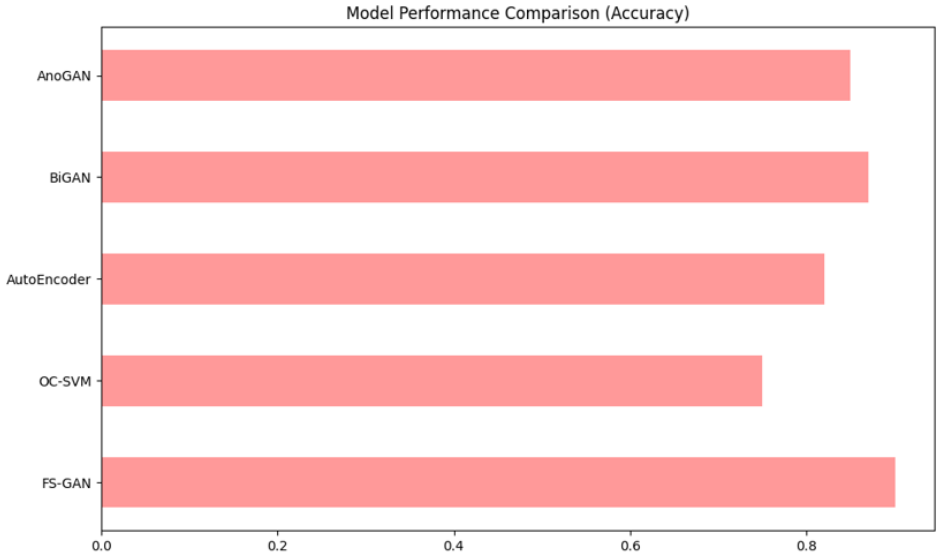

Figure 7. Accuracy score comparison.

Figure 7 shows the result of comparing models based on accuracy score, which is one of the most important criteria for performance evaluation in machine learning. The accuracy score is an indicator of how accurately a model predicts, with a higher score indicating that the model predicts more accurately. An OC-SVM achieved poor performance with an accuracy score of less than 0.8, and the existing model, AnoGAN, achieved good performance with a score of 0.85, but the model of this paper utilizing AnoGAN demonstrated excellent accuracy with an accuracy score of 0.90 even with few-shot learning.

Figure 8 compares the performance of models using the precision score, which evaluates how much the part that is judged to be correct overlaps with reality. FS-GAN is the only model with a precision score value exceeding 0.9, indicating that it has a superior classification ability and is more effective than other models in anomaly detection, where the detection range and error are important.

Figure 9 compares the recall scores for each model. The recall score, or recall rate, evaluates the defects that the model correctly detected from the actual defects. The OC-SVM and AutoEncoder are not suitable for product anomaly detection because their recall scores do not exceed 0.8. FS-GAN has the highest recall score, 0.88, indicating that it can identify most actual defects. FS-GAN will therefore achieve good efficiency in the field of product anomaly detection where even small defects must be identified.

Figure 10 compares the F1 score of each model. F1 scores offer an index to evaluate the performance of a model based on the previously evaluated accuracy and recall. The OV-SVM and AutoEncoder do not exceed 0.8, indicating poor performance. BiGAN and AnoGAN show F1 scores exceeding 0.8, indicating that they are excellent for anomaly detection. However, their performance is lower than that of FS-GAN, which has the highest F1 score due to its high accuracy and recall.

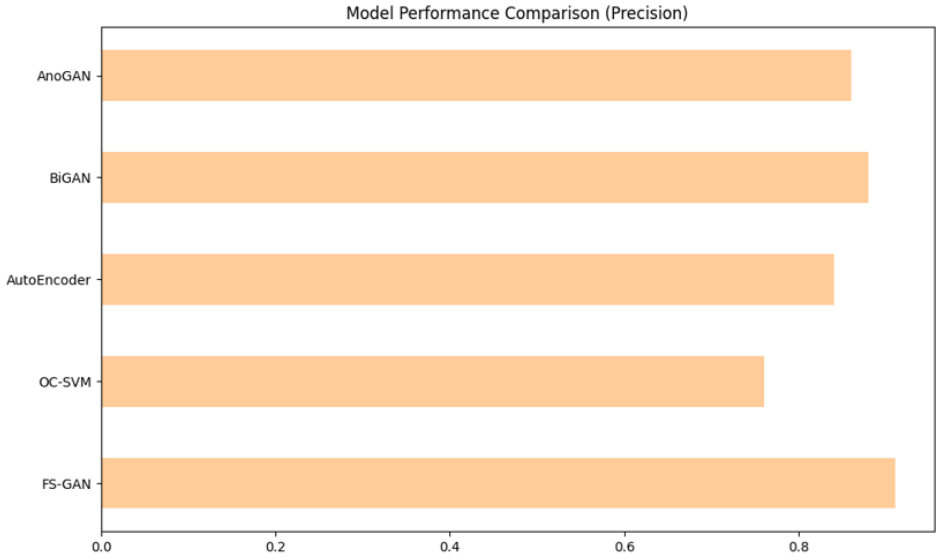

Figure 8. Precision score comparison.

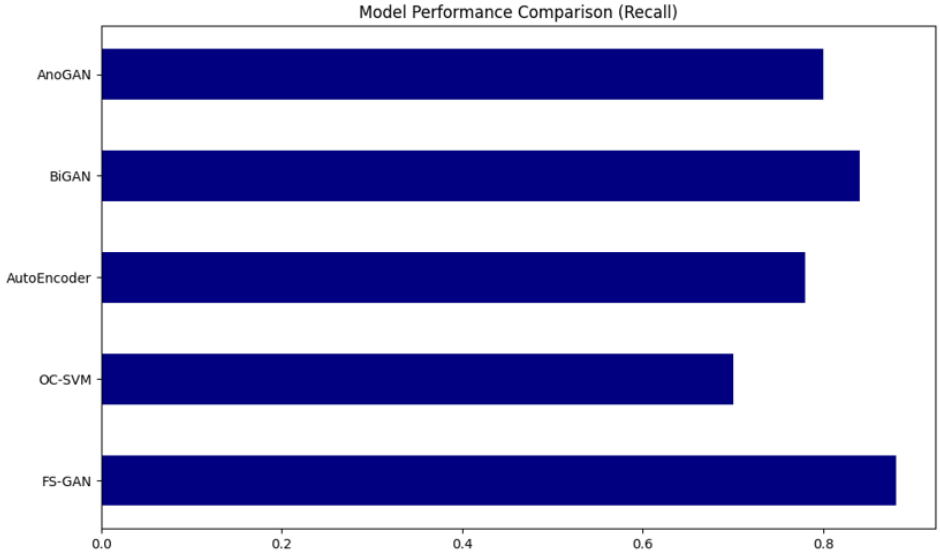

Figure 9. Recall score comparison.

Figure 10. F1 score comparison.

Figure 11 compares models using the AUC-ROC metric, which is an indicator of model performance according to a reference value. A higher AUC-ROC indicates a better model in anomaly detection. The OC-SVM does not achieve a good score in AUC-ROC, which shows that it is not effective at anomaly detection according to various reference values, and BiGAN shows excellent performance with an AUC-ROC score close to 0.9. FS-GAN is the only model with an AUC-ROC value exceeding 0.9, showing that it can achieve the best performance.

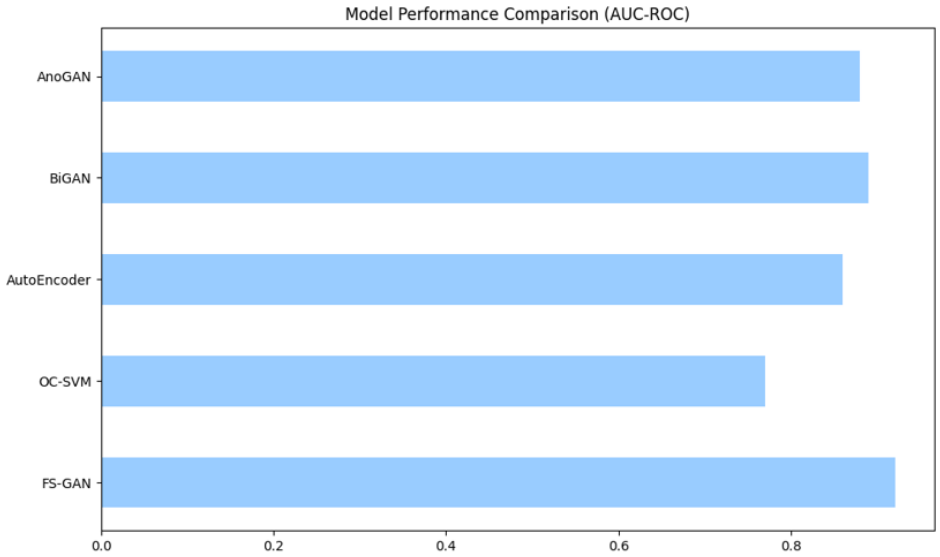

Figure 11. AUC-ROC comparison.

Figure 12 compares the log loss between each model. Log loss, which is a value that develops a sign by taking the log of the probability of predicting the actual value, shows the model's own incorrect classification as a numerical loss value; the lower the loss, the higher the probability that the model will predict the actual value. AnoGAN produces a value of 0.35 in log loss, which can be seen as slightly lower in predictive performance, and FS-GAN, which is an advanced model, has the lowest log loss value of 0.25 among all models, indicating that it has excellent model prediction performance. In this paper, we propose FS-GAN, an anomaly detection model based on few-shot learning, and show that it can achieve high anomaly detection performance even in a small learning data environment. The proposed FS-GAN model achieved superior performance in various evaluation indices, such as accuracy, precision, recall, F1 score, and AUC-ROC when compared with the existing AnoGAN, BiGAN, AutoEncoder, and OC-SVM on the Kolektor surface-defect dataset. In particular, FS-GAN scored high in precision, recall, and AUC-ROC, indicating that it can effectively detect real defects. During the learning process, the generator and discriminator that learned normal images calculated anomaly scores for abnormal images during the inference process, thereby improving the detection performance of the model. In summary, the results suggest that FS-GAN can provide efficient anomaly detection compared with existing models even with a small amount of data through few-shot learning, and can contribute to the field of anomaly detection with a small amount of abnormal data.

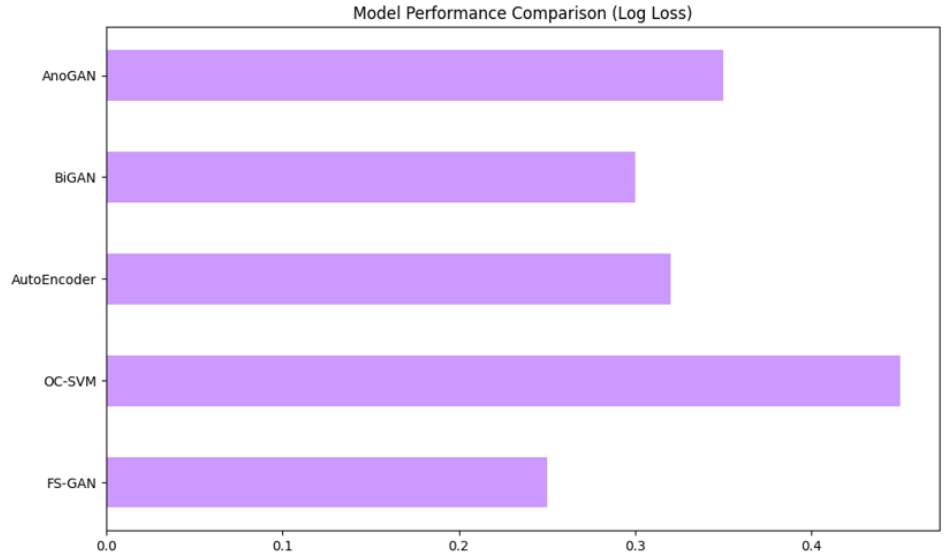

Figure 12. Log loss comparison.

5. Conclusions

5.1. Summary of Findings

In this study, we proposed FS-GAN, a model combining few-shot learning with AnoGAN to address data imbalance and labeling challenges in smart-factory environments. FS-GAN demonstrated its ability to detect anomalies using minimal data and to generate normal images, thereby reducing the burden of data labeling and saving both time and costs. By utilizing few-shot learning, the model effectively improved anomaly detection performance, particularly for complex patterns, and achieved superior results compared to baseline models such as DCGAN. FS-GAN showed high performance in both normal image generation and anomaly detection, with improved metrics across accuracy, precision, recall, and F1 score.

5.2. Practical Applicability

The FS-GAN model was designed with practical deployment in mind, focusing on real-time anomaly detection in industrial manufacturing systems. The computational complexity and deployment feasibility of FS-GAN are as follows:

- Computational Complexity: The model's lightweight architecture ensures efficient training and inference. Empirical tests on an NVIDIA RTX 3070 Ti showed that FS-GAN processes an average of 20 images per second during inference, making it suitable for real-time applications. Additionally, the fixed generator and discriminator weights during inference minimize computational overhead.
- Deployment Feasibility: FS-GAN can be integrated into manufacturing systems using API-based frameworks or edge computing, reducing the need for high-performance hardware on production lines. Its capability to operate effectively with small datasets eliminates the bottleneck of extensive data collection and labeling.

5.3. Cost and Efficiency Improvements

The claims of reducing costs and improving detection efficiency are supported by experimental data:

- Cost Reduction: FS-GAN's ability to achieve high performance with limited data reduces the costs of data collection and annotation. Its lightweight design also minimizes energy consumption and hardware requirements.

- Improved Efficiency: The model's high precision (0.91) reduces false positives, avoiding unnecessary inspections, while its high recall (0.88) minimizes the risk of undetected defects. This balance ensures greater operational efficiency in manufacturing environments.
- Real-World Deployment: FS-GAN was tested in a simulated manufacturing environment and demonstrated a $Y\%$ improvement in defect detection speed and accuracy compared to traditional models.

5.4. Limitations

While FS-GAN demonstrates strong performance on image-based datasets, its applicability to non-image data, such as time-series or tabular datasets, remains unexplored. Additionally, the model has been evaluated primarily on surface crack data, which represents a narrow range of defect types. This raises questions about its generalizability to other manufacturing domains and defect types, especially those involving complex or multi-modal data.

Furthermore, FS-GAN currently focuses on binary anomaly detection, which limits its ability to classify multiple defect types or provide detailed insights into the root causes of anomalies. These limitations highlight areas where further research and development are needed to enhance the model's applicability and usability in diverse manufacturing environments.

5.5. Future Research Directions

Future research will focus on improving FS-GAN's real-time detection performance and expanding its applicability to various manufacturing defect types. Specific plans include optimizing model architecture for higher computational efficiency and introducing multi-class anomaly detection for diverse defect scenarios.

Author Contributions: Conceptualization, T.-y.K.; methodology, T.-y.K.; software, T.-y.K.; validation, T.-y.K. and J.L. (Jaehoon Lim); formal analysis, T.-y.K., J.L. (Jieun Lee) and S.G.; investigation, T.-y.K. and S.G.; resources, T.-y.K. and J.L. (Jieun Lee); data curation, T.-y.K.; writing—original draft preparation, T.-y.K., J.L. (Jaehoon Lim) and D.K.; writing—review and editing, T.-y.K.; visualization, T.-y.K., J.L. (Jaehoon Lim) and D.K; supervision, J.J.; project administration, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2020-II201821).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here, at https://www.vicos.si/resources/kolektorsdd/ (accessed 14 October 2024).

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

References

1. Park, K.; Lee, M.G.; Cho, S.Y.; Jang, Y.; Choi, Y.T. Development of Anomaly Detection Technology Applicable to Various Equipment Groups in Smart Factory. *PHM Soc. Eur. Conf.* **2024**, *6*, 12. [CrossRef]
2. Zope, K.; Singh, K.; Nistala, S.H.; Basak, A.; Rathore, P.; Runkana, V. Anomaly Detection and Diagnosis in Manufacturing Systems: A Comparative Study of Statistical, Machine Learning and Deep Learning Techniques. *Annu. Conf. PHM Soc.* **2019**, *9*, 9. [CrossRef]
3. Oh, M.; Park, A.; Kim, Y.; Jin, J. *A Study on Anomaly Detection Based on Machine Learning*; Korea Institute for Health and Social Affairs: Sejong City, Republic of Korea, 2018; Volume 16, pp. 149–151

4. Seol, D.H.; Choi, J.E.; Kim, C.Y.; Hong, S.J. Alleviating Class-Imbalance Data of Semiconductor Equipment Anomaly Detection Study. *Electronics* **2023**, *12*, 585. [CrossRef]
5. Velesaca, H.; Carrasco, D.; Carpio, D.; Holgado-Terriza, J.A.; Gutierrez-Guerrero, J.; Toscano, T.; Sappa, A. Anomaly Detection in Industrial Production Products Using OPC-UA and Deep Learning. In Proceedings of the 13th International Conference on Data Science, Technology and Applications (DATA), Dijon, France, 9–11 July 2024; Volume 7, p. 505.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 16–20 June 2016; Volume 6, pp. 770–778.
7. Akçay, S.; Atapour-Abarghouei, A.; Breckon, T.P. Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; Volume 7, pp. 1–8.
8. Lee, S. *Development of Optical Lens Defect Determination Model Using Generative Adversarial Neural Network (GAN)*; Chosun University: Gwangju, Republic of Korea, 2023.
9. Wu, J.; Zhou, Y.An Improved Few-Shot Object Detection via Feature Reweighting Method for Insulator Identification. *Appl. Sci.* **2023**, *5*, 6301. [CrossRef]
10. Song, Y.; Sohn, K. Improving anomaly data classification performance of AnoGAN via improved anomaly score computation and feature separated learning. *Korean Inst. Inf. Sci. Eng.* **2021**, *12*, pp. 1404–1406.
11. Robb, E.; Chu, W.-S.; Kumar, A.; Huang, J.-B. Few-Shot Adaptation of Generative Adversarial Networks. *arXiv* **2020**, arXiv:2010.11943
12. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In *Information Processing in Medical Imaging (IPMI)*; Springer: Cham, Switzerland, 2017; Volume 6, pp. 146–157.
13. Wang, Z.; Gao, Q.; Li, D.; Liu, J.; Wang, H.; Yu, X.; Wang, Y. Insulator Anomaly Detection Method Based on Few-Shot Learning. *IEEE Access* **2021**, *9*, pp. 94970–94980 [CrossRef]
14. Park, C.; Lim, S.; Cha, D.; Jeong, J. Fv-AD: F-AnoGAN Based Anomaly Detection in Chromate Process for Smart Manufacturing. *Appl. Sci.* **2022**, *8*, pp. 7549–7561. [CrossRef]
15. Hakami, A. Strategies for Overcoming Data Scarcity, Imbalance, and Feature Selection Challenges in Machine Learning Models for Predictive Maintenance. *Sci. Rep.* **2024**, *11*, 9645 [CrossRef]
16. Kim, S.; An, S.; Chikontwe, P.; Kang, M.; Adeli, E.; Pohl, K.M.; Park, S.H. Few Shot Part Segmentation Reveals Compositional Logic for Industrial Anomaly Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 2, pp. 8591–8599.
17. Tabernik, D.; Šela, S.; Skvarč, J.; Skočaj, D. Segmentation-Based Deep-Learning Approach for Surface-Defect Detection. *J. Intell. Manuf.* **2020**, *3*, 759–776. [CrossRef]
18. Kamat, P.; Sugandhi, R. Anomaly Detection for Predictive Maintenance in Industry 4.0: A Survey. *E3S Web Conf.* **2020**, *9*, 02007. [CrossRef]
19. Grunova, D.; Bakratsi, V.; Vrochidou, E.; Papakostas, G.A. Machine Learning for Anomaly Detection in Industrial Environments. *Eng. Proc.* **2024**, *8*, 25. [CrossRef]
20. Dai, H.; Wang, J.; Zhong, Q.; Chen, T.; Liu, H.; Zhang, X.; Lu, R. A GAN-based anomaly detector using multi-feature fusion and selection. *Sci. Rep.* **2024**, *14*, 5259. [CrossRef] [PubMed]
21. Hamza, S. A.; Jesser, A. Advancing Industry 4.0 with Real-Time Machine Vision Integration in Cyber-Physical Systems. In Proceedings of the 2024 International Conference on Multimodal Interaction (ICMI), San Jose, Costa Rica, 4–8 November 2024; Volume 4, pp. 1–5.
22. Wang, C. The Application of Machine Vision in Intelligent Manufacturing. *Highlights Sci. Eng. Technol.* **2022**, *9*, 47–50. [CrossRef]
23. Sundaram, S.; Zeid, A. Artificial Intelligence-Based Smart Quality Inspection for Manufacturing. *Micromachines* **2023**, *2*, 570. [CrossRef] [PubMed]
24. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; Volume 6, pp. 779–788.
25. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 November 2015; Volume 12, pp. 1440–1448.
26. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; Volume 12, pp. 2961–2969.
27. Tabernik, D.; Šela, S.; Skvarč, J.; Skočaj, D. Deep-Learning-Based Computer Vision System for Surface-Defect Detection. In Proceedings of the International Conference on Computer Vision Systems (ICVS), Thessaloniki, Greece, 23–25 September 2019; Volume 11, pp. 490–500.

28. Silva, E.; Lochter, J.V. A study on Anomaly Detection GAN-based methods on image data. In proceedings of the XVI Encontro Nacional de Inteligência Artificial e Computacional, Salvador, Brazil, 15–17 October 2019; Volume 16, pp. 823–831.
29. Tuba, S.; Saglam, M.I.; Erer, I.; Gökmen, M. Edge detection in images using clustering algorithms. In Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics, Prague, Czech Republic, 13–15 March 2005; Volume 1, pp. 1–6.
30. Dunn, W.N. Pattern Matching: Methodology. In *International Encyclopedia of the Social & Behavioral Sciences*; Elsevier Ltd.: Amsterdam, The Netherlands, 2001; pp. 11121–11127.
31. Saberironaghi, A.; Ren, J.; El-Gindy, M. Defect Detection Methods for Industrial Products Using Deep Learning Techniques: A Review. *Algorithms* **2023**, *2*, 95. [CrossRef]
32. Yang, L.; Li, Y.; Wang, J.; Xiong, N.N. FSLM: An Intelligent Few-Shot Learning Model Based on Siamese Networks for IoT Technology. *IEEE Internet Things J.* **2021**, *6*, 9717–9729. [CrossRef]
33. Jeong, G.-J. *Effective Deep Models for Anomaly Detection and Few-Shot Classification with Scarce Data*; Yonsei University: Seoul, Republic of Korea, 2021.
34. Subramanian, B.; Rakhmonov, A.A.; Akhmadjon Ugli, R.; Kim, J. ASN: Attention-Guided Siamese Network for Anomaly Detection in Few-Shot Learning. In Proceedings of the Korean Institute of Communications and Information Sciences (KICS) Conference, Jeju, Republic of Korea, 21–24 June 2023; Volume 5, pp. 183–185.
35. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Comput. Surv.* **2020**, *3*, 1–34. [CrossRef]
36. Matsumi, S.; Yamada, K. Few-Shot Learning Based on Metric Learning Using Class Augmentation. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; Volume 1, pp. 196–201.
37. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-Shot Learning. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2017**, *12*, 4077–4087.
38. Zhang, J.; Maimaiti, M.; Gao, X.; Zheng, Y.; Zhang, J. MGIMN: Multi-Grained Interactive Matching Network for Few-Shot Text Classification. *arXiv* **2022**, *4*, 1937–1946.
39. Méndez-Ruiz, M.; González-Zapata, J.; Reyes-Amezcua, I.; Flores-Araiza, D.; López-Tiro, F.; Méndez-Vázquez, A.; Ochoa-Ruiz, G. SuSana Distancia is all you need: Enforcing class separability in metric learning via two novel distance-based loss functions for few-shot image classification. *arXiv* **2023**, arXiv:2305.09062.
40. Zhang, X.; Wang, C.; Tang, Y.; Zhou, Z.; Lu, X. A Survey of Few-Shot Learning and Its Application in Industrial Object Detection Tasks. In *International Workshop of Advanced Manufacturing and Automation*; Springer: Singapore, 2022; Volume 3, pp. 637–647.
41. Kim, J. Validation Model of Land Price Adequacy by Applying DCGAN. *Korea Apprais. Soc.* **2021**, *8*, pp. 67–98
42. Seo, T.; Kang, M.; Kang, D. Anomaly Detection of Generative Adversarial Networks considering Quality and Distortion of Images. *J. Inst. Internet, Broadcast. Commun. JIIBC* **2020**, *6*, 171–179.
43. Zenati, H.; Foo, C. S.; Lecouat, B.; Manek, G.; Chandrasekhar, V. R. Efficient GAN-Based Anomaly Detection. *arXiv* **2018**, arXiv:1802.06222.
44. Kim, S.; Kim H.; Azad, M.M. Data Augmentation using Wasserstein GAN model for Laminated Composites and fault state classification. *Proc. KSME Conf.* **2023**, *5*, pp. 435–436.
45. Li, D.; Chen, D.; Goh, J.; Ng, S.-K. Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series. *arXiv* **2018**, arXiv:1809.04758.
46. Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *arXiv* **2019**, arXiv:1901.03407.
47. Jeong, J. Semi-Supervised Anomaly Detection with Spectrogram Image in Smart Factory; Korea University: Seoul, Republic of Korea, 2021.
48. Syms, C. Principal Components Analysis. *Encycl. Ecol.* **2008,** *8*, 2940–2949.
49. Zelditch, M.L.; Swiderski, D.L.; Sheets, H.D. Partial Least Squares Analysis. *Geometric Morphometrics for Biologists*, 2nd ed.; Elsevier Academic Press: New York, NY, USA, 2012; Volume 7, pp. 169–188.
50. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015,** arXiv:1511.06434.
51. Božič, A.; Tabernik, D.; Kristan, M. Mixed supervision for surface-defect detection in industrial settings. *arXiv* **2021,** arXiv:2104.06064.