crystals

MDPI

Article

Automated Optical Inspection Method for Light-Emitting Diode Defect Detection Using Unsupervised Generative Adversarial Neural Network

Che-Hsuan Huang [1,2,†], Pei-Hsuan Lee [2,†], Shu-Hsiu Chang [3], Hao-Chung Kuo [2,*], Chia-Wei Sun [2], Chien-Chung Lin [3], Chun-Lin Tsai [2] and Xinke Liu [1]

[1] Guangdong Research Center for Interfacial Engineering of Functional Materials, College of Materials Science and Engineering, Shenzhen University, Shenzhen 518060, China; seadeep08281226@gmail.com (C.-H.H.); xkliu@szu.edu.cn (X.L.)

[2] Department of Photonics and Institute of Electro-Optical Engineering, College of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan; ph63.c@nycu.edu.tw (P.-H.L.); chiaweisun@nctu.edu.tw (C.-W.S.); asd123456789626@gmail.com (C.-L.T.)

[3] Institute of Photonic System, National Yang Ming Chiao Tung University, Tainan City 71150, Taiwan; frank.chang@leda-creative.com (S.-H.C.); chienchunglin@faculty.nctu.edu.tw (C.-C.L.)

* Correspondence: hckuo@faculty.nctu.edu.tw

† These authors contributed equally to this work.

Abstract: Many automated optical inspection (AOI) companies use supervised object detection networks to inspect items, a technique which expends tremendous time and energy to mark defectives. Therefore, we propose an AOI system which uses an unsupervised learning network as the base algorithm to simultaneously generate anomaly alerts and reduce labeling costs. This AOI system works by deploying the GANomaly neural network and the supervised network to the manufacturing system. To improve the ability to distinguish anomaly items from normal items in industry and enhance the overall performance of the manufacturing process, the system uses the structural similarity index (SSIM) as part of the loss function as well as the scoring parameters. Thus, the proposed system will achieve the requirements of smart factories in the future (Industry 4.0).

Keywords: abnormal detection; GAN; LED; machine learning; unsupervised learning

check for updates

Citation: Huang, C.-H.; Lee, P.-H.; Chang, S.-H.; Kuo, H.-C.; Sun, C.-W.; Lin, C.-C.; Tsai, C.-L.; Liu, X. Automated Optical Inspection Method for Light-Emitting Diode Defect Detection Using Unsupervised Generative Adversarial Neural Network. *Crystals* **2021**, *11*, 1048. https://doi.org/10.3390/cryst11091048

Academic Editor: Shujun Zhang

Received: 1 July 2021
Accepted: 24 August 2021
Published: 31 August 2021

1. Introduction

After data input and data preprocessing, an ordinary AOI system will generally use a basic algorithm [1–3] or a supervised model [4–6] as the inspection method. These methods require customization or huge amounts of training data and are only suitable for processing a predefined type of data. Unfortunately, with such a wide variety of products and production rates in a general factory setting, it is difficult to obtain sufficient anomaly data to train supervised models. This in turn makes it excessively laborious and time-consuming to customize each algorithm or to train each set of weights of the supervised model for a specific anomaly type.

Unsupervised generative neural networks [7,8] have been developed rapidly in recent years. Among them, the application models of anomaly detection series are provided with inspection abilities for processing various kinds of data. Since unsupervised generative neural networks only require a small amount of normal data for training, it is more suitable to deploy unsupervised neural networks in factories rather than supervised networks for the AOI applications [9] in the future.

In 2015, the variational autoencoder-based anomaly detection method [10], which is a reconstruction-based anomaly detection model, achieves the purpose of image generation by adding the Gaussian distribution of the original data as the output restriction to the latent vector in the autoencoder.

A Generative Adversarial Network (GAN) is used for anomaly detection and consists of a generative network and a discriminator. The generative network learns to create fake images which are similar to known sample images. The discriminator learns to distinguish real images from these synthetic fake images, and identifies abnormal pixels [10]. Therefore, GANs can recognize unknown defects and different types of new samples via unsupervised learning. GANs are divided into three types: AnoGAN, Efficient-GAN and GANomaly models [11].

In 2017, Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs published a generative adversarial network (GAN) based unsupervised anomaly detection model named AnoGAN [12]. It is a reconstruction-based model with a score function constructed using mapping from the image space to the latent space. AnoGAN is applied to retinal optical coherence tomography (OCT) image detection.

To achieve a better performance on images and network intrusion datasets than other published methods, and significantly shorten the test time, efficient GAN-based anomaly detection (EGBAD) models, raised in 2018, learn encoders which map inputs to latent representations in training, and thus EGBAD models do not have to recover latent representations at test time [13].

In the same year, Samet Akcay, Admir Atapour-Abargghouei, and Toby P. Breckon proposed generative-based model GANomaly [11] and Skip-GANomaly [14]. These anomaly detection models are autoencoder-based adversarial autoencoders (AAEs) and they can be applied to MNIST [15], CIFAR-10 [16], UBA, and FFOB [17] datasets. Due to the great inspection abilities and the high arithmetic speed of GANomaly, this model will be adopted in this paper.

With the development of industrial intelligence, manufacturing technology has evolved to High Variety Low Volume production. The highly adaptable deep learning algorithms have gradually been adopted in AOI systems. The attributes of industrial scene data are quite different from the attributes of data from public databases. Compared to public databases, most data in industrial scenes are unblemished and look alike; the defect samples are fewer and fewer nowadays. However, a lack of defect data makes it hard to detect objects precisely in object detection or to create labels correctly for segmentations. To resolve this problem, we propose a GANomaly technique which finds abnormal pixel characteristics by learning from normal samples. Furthermore, we use loss functions to improve the performance of the whole GANomaly model.

2. Experiment and Architecture of the System

In this section, we introduce the concept of our AOI system, which contains the dataset, the algorithm, the scoring method, and the settings.

2.1. Introduction of the AOI System

In this paper, we propose a human-machine interaction AOI System which can respond rapidly to different data inputs. Before applying an algorithm or a supervised defect detection model in the assembly line, we deploy a GANomaly anomaly detection model, as seen in Figure 1. Since the GANomaly model is highly effective at identifying anomalies for a variety of different samples, and only requires a small amount of the normal training data, it is fast in processing data and easily achieves a fairly good detection effect.

We employ GANomaly to distinguish newly encountered defects from existing data, as the model can quickly notify the developer that there is a need to operate a new abnormal data analysis when a new abnormal data distribution occurs. After confirming that a new type of abnormal data has been identified, the developer can use the pre-labeled method proposed in this paper for GANomaly to generate labeled data automatically. These labeled data will be provided to the supervised neural network for training and to adjust the weights. Therefore, our AOI system not only improves the overall yield rate significantly by reducing the impact caused by new incoming defects, but the system also cuts down the labor and time costs for labeling new anomaly data [18–20].

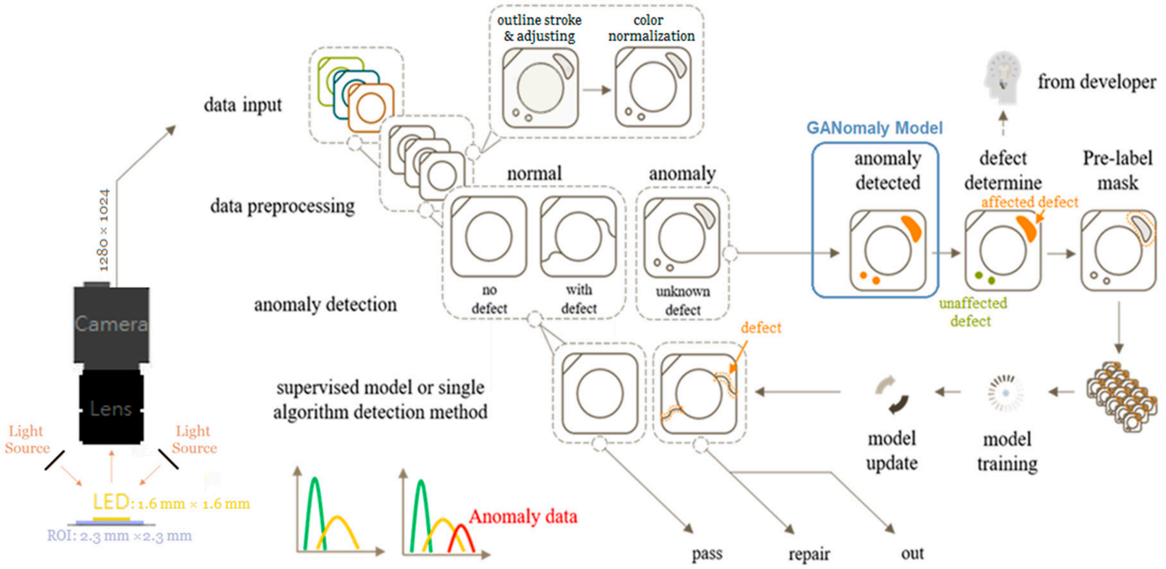

Figure 1. Illustration of the AI-AOI system.

2.2. Dataset

Our dataset is a collection of packaged LED surface images, where the LED size is 1.6 mm × 1.6 mm, and the region of interest is 2.3 mm × 2.3 mm. There are 5 different packaging and wiring methods, as well as three categories of defects: plate rift defects, unclean, and glue overflow defects. These defects are individually distributed on the surfaces of different types of LEDs. Besides, not all types of LEDs will be confronted with all kinds of defects. As shown in Figure 2, plate rift defects do not occur in Type 3, Type 4, and Type 5; however, unclean defects do not occur in Type 4 and Type 5.

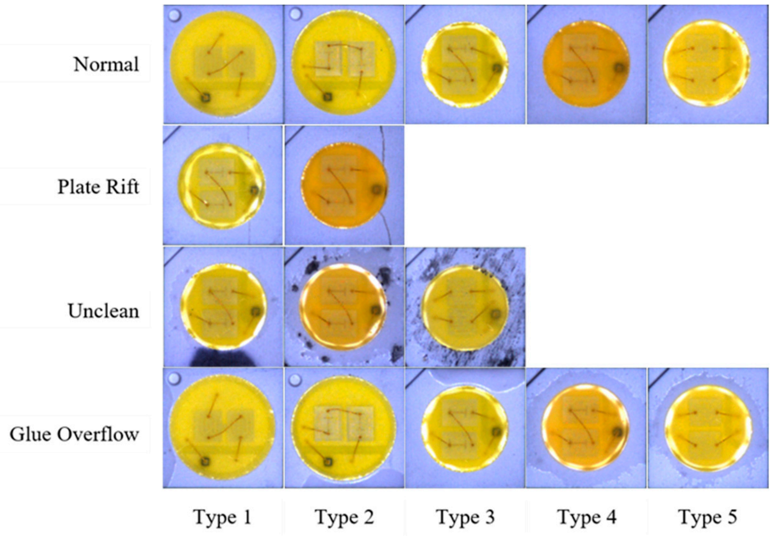

Figure 2. Five types of LED surface image samples, including three different categories of defects: (1) Plate rift defects; (2) Unclean defects; (3) Glue overflow defects.

Coaxial lights were used to evenly illuminate the packaged LEDs and a CMOS industrial camera was used to acquire images of which the resolutions are 1280 × 1024 pixels. To recognize the insides and the outsides of LEDs, there are two image preprocessing steps that need to be complete before training. First, gray-scaling the image, obtaining the LED profile data by a specific threshold, and then getting the full-page LED surface image map by rotating and capturing the image based on the profile of the LED is completed. Second, color normalization is performed using Gamma Correction on the inside and the outside of the LED surface separately. Without color normalization, it is hard to recognize the insides and the outsides of LEDs, and the system will waste time detecting defects on unnecessary parts. After color normalization, we can obtain an image map, as shown in Figure 2, which is helpful for feature extraction in the neural network.

2.3. *The Contextual Loss Function and the Score Function*

The backbone of our algorithm, as previously discussed, is GANomaly, which is an unsupervised generative confrontation anomaly detection model based on autoencoder and imported into a confrontation network. By using a conditional generative confrontation network, our algorithm converts high-dimensional training data images into latent space representations and uses the concept of the encoding/decoding model to reconstruct the images which contain the features of learning data. After that, the reconstructed images are graded by a scoring function, and the abnormal data are identified by delineating the normal zones. To help GANomaly focus on adjusting the data to normal, three loss functions are employed to standardize the operations of GANomaly. These three types of functions are: contextual loss, encoder loss, and adversarial loss. As shown in Figure 3 (drawn based on [11]), z is the latent space, and the contextual loss function compares the differences between the fake image generated from the autoencoder and the original image by L1 function (Option 1 in Figure 3). The encoder loss function compares the differences between two latent representations. One of the latent representations is generated by the encoder during the auto-encoding, and the other is generated by the same encoder of which the input is a fake image. The adversarial loss function compares the differences between the two latent representations generated by the encoder. One is generated from the real image and the other one is generated from the fake image. The function then operates the mini-max operation with the two intermediate latent representations and calculates the distance between these two function outputs. With the three loss functions mentioned above, the model can generate the features of normal data by training with only a few normal images.

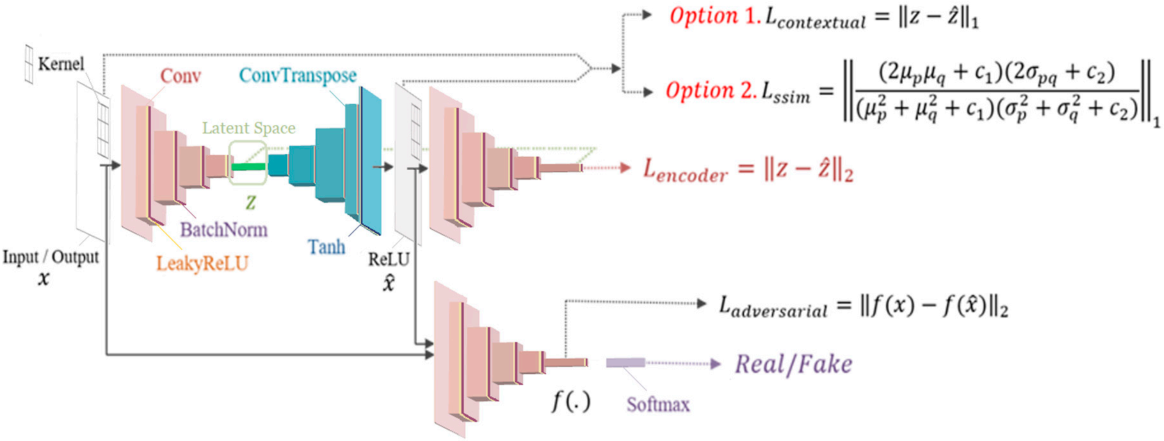
Option 1. $L_{contextual} = \|z - \hat{z}\|_1$
Option 2. $L_{ssim} = \left\| \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)} \right\|_1$
Kernel
Conv
ConvTranspose
Latent Space
z
BatchNorm
LeakyReLU
Tanh
ReLU
$\hat{x}$
Input / Output
x
$L_{encoder} = \|z - \hat{z}\|_2$
$L_{adversarial} = \|f(x) - f(\hat{x})\|_2$
Real/Fake
$f(.)$
Softmax

Figure 3. Model training under two different contextual losses: (**1**) L1 loss; (**2**) SSIM loss.

When a model generates a fake image, it defines a score function to compare the differences between the fake and the real images, as with Equation (1) [5].

$$A(x_i) = \lambda R(x_i) + (1 - \lambda) L(x_i) \tag{1}$$

where x_i represents the ith input, $A(x_i)$ is the anomaly score, $R(x_i)$ is the L1 score produced from the comparison between the fake and the real images, $L(x_i)$ is the L2 score based on the latent representation of real and fake images, and λ is the weight between the contextual and encoder scores. The obtained score will be normalized by Equation (2) [9], and the normalized result is the final score we are interested in.

$$\hat{A}(x_i) = \frac{A(x_i) - \min(A)}{\max(A) - \min(A)} \tag{2}$$

where A is the anomaly score vector, x_i is the ith vector, $A(x_i)$ is the score vector of x_i, and $\hat{A}(x_i)$ is the normalized score vector. After that, we can use all the obtained images to calculate the area under the curve (AUC) value, which represents the overall performance of the model. The AUC value is the most important index of an anomaly detection model, where the main curve is the Receiver Operation Characteristic (ROC) curve. The ROC curve is a graphical representation of how well a classification model performs. The curve is plotted on the TPR-FPR (True Positive Rate-False Positive Rate) plane and consists of data calculated from the thresholds which can be set, which is similar to each sample score. The AUC represents the ability to detect anomalies. “AUC = 0.5” implies no ability of anomaly detection, and “AUC = 1” implies the complete ability of anomaly detection. To apply the AOI system to the industry, the required AUC is at least 0.85 or even 0.9.

There are 2 comparison functions in a GANomaly model, one is the loss function used in training and the other is the loss function using in the comparisons between real sample images and generated fake normal sample images. For the purpose of convenience, the models are named with the following format: GANomaly-[training loss function]-[comparison loss function].

Our experiments contain three architectures. One is GANomaly-L1-L1, which provides us with outputs from the original GANomaly model of which inputs are the LED data set. Another is GANomaly-SSIM-L1, employing the mSSIM [9] loss function instead of the L1 loss function as the contextual loss function in GANomaly, and test models with L1 score function to improve the performance. The other is GANomaly-SSIM-SSIM, replacing the L1 score with the SSIM score as the contextual loss function and the score function.

$$A_{ssim}(x_i) = \lambda SSIM(x_i) + (1 - \lambda)L(x_i) \tag{3}$$

It is worth noting that the GANomaly architectures of GANomaly-SSIM-L1 and GANomaly-SSIM-SSIM are the same, so the same parameters will be used in the parameter optimizations process mentioned in the next section. As shown in Figure 4, two different comparison methods, L1 and mSSIM scores, are used to compare the unknown sample images and the generated fake normal sample images.

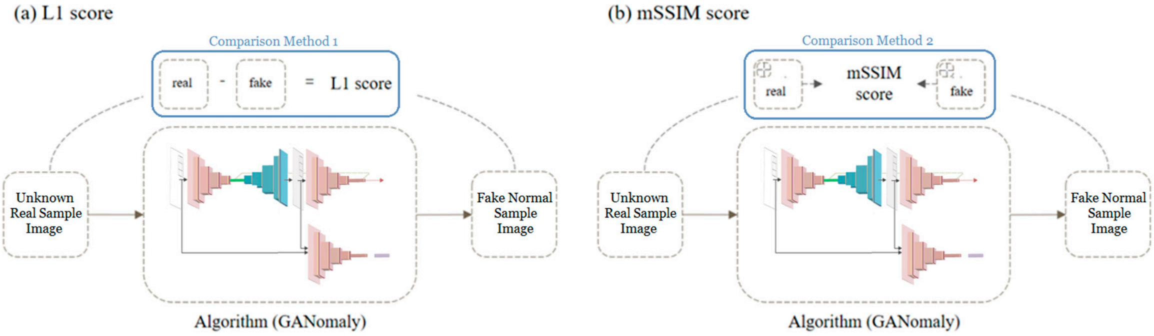

Figure 4. Scoring for each sample under two different methods: (**a**) L1; (**b**) mSSIM.

This study proposes the use of a pre-labeled method to extract the profile of the abnormal data, which implies the defect positions on LEDs detected by GANomaly, and converts the detection results into labeled data. The unsupervised feature of GANomaly is then employed to automatically generate labeled data, which also has a positive effect on the supervised models in AOI systems. This experiment compares the effectiveness of the system with and without the pre-labeled method, based on the quality and the time it took to complete the labelling.

2.4. Parameter Optimizations

The three algorithm architectures mentioned in the previous section will go through a series of parameter optimizations for our data set, before the performance is tested. The adjustments for GANomaly are described below.

- The weights of three loss functions are tested separately at intervals of 10, and the tests range from 1 to 100. Each of the functions has 11 parameters, thus there are a total of 113 possibilities under the interactive test. Since it is inefficient to calculate these one by one, we use a 3-stage random sampling approximation method to obtain the best solution.
- The sizes of latent representations are 64, 100, 128, 256, 512, and 1024. The tests are used to ensure the model will not be confronted with the bottleneck effect, which is caused by the excessive amount of image pixels.
- The image resolutions are 64 × 64, 128 × 128, and 256 × 256 pixels. To ensure the image features are retained, it is vital to select the most suitable model for subsequent image scoring processing under different image sizes and sufficient model convergence (good generation ability).
- For architecture 2 and architecture 3, the parameters of the SSIM size are optimized. The intervals are set to 1 and ranges from 8 to 14.

The parameter adjustments for the score function are described below.

- Optimizing the scoring weight λ. To find the best ratio for the score function, based on the difference between the model contextual and the encoder, we set the interval to 0.1 and the range from 0 to 1.
- For architecture 2 and architecture 3, the SSIM score in the score function is optimized with the kernel size, and the interval is set to 1 and range from 8 to 14.

After the parameter optimizations, the optimized parameters of GANomaly-L1-L1, GANomaly-SSIM-L1, and GANomaly-SSIM-SSIM are listed in Table 1. After sufficient convergence, which is mentioned in the next section, the system uses optimized parameters and then selects the best result from the outputs.

Table 1. Optimized parameters.

Parameter	**GANomaly-L1-L1**	**GANomaly-SSIM-L1**	**GANomaly-SSIM-SSIM**
Contextual loss	1	50	50
Encoder loss	1	1	1
Adversarial loss	20	1	1
Latent size	100	100	100
Image size	64	128	128
Kernel size	x	11	11
Score weight (λ)	0.5	0.8	0.8

2.5. Hardware and Software Configurations

We used Intel® core™ i7-5820k CPU and Nvidia GTX 1080 Titan GPU to train the model and to test the performance of our algorithms, and used PyTorch to perform the data pre-processing, GANomaly, SSIM, and Pre-labeled method. The training parameters are listed in Table 2.

Table 2. Training parameters.

Parameter	Value
Training epochs	200
Learning rate	0.0001
Earning policy	lambda
Batch size	1
Momentum beta	0.5

Since the data amount from the factory is limited, we use 500 normal images to train the model. There are 100 training images per LED type. The number of testing (validation) images is 600, including 300 normal data and 300 abnormal data. There are 60 normal data for each LED type, and 100 abnormal data for each defect. Due to the limited data amount, the number of defect categories under different LED types do not follow any specific distribution.

3. Results and Discussion

3.1. Performance of GANomaly-L1-L1

After the parameter optimizations, the adversarial loss function dominates the generation data flow in algorithm GANomaly-L1-L1. The unnecessary details may cause the scoring function to over-grade, resulting in an excessive overlap in the score distribution. Therefore, after the image is generated, the image resolution is resized to 64 × 64 pixels to reduce the number of unnecessary details.

Under the framework of this algorithm, the best solution obtained by our calculation is the anomaly detection with AUC = 0.8395, as shown in Figure 5a. As seen in Figure 5b, the overlap of normal and abnormal data is quite large. The reasons for this are described below.

- The ability of the model to generate original data is limited. In other words, the generated image cannot sufficiently retain the normal data features of the original real image, and the fake image restoration is excessively consistent, so the features images are not good enough to identify abnormalities in the subsequent scoring step.
- Under the assumption that the first problem does not exist, the abilities of the previous scoring functions are too compendious to distinguish images, which is the main problem of the L1 score function.

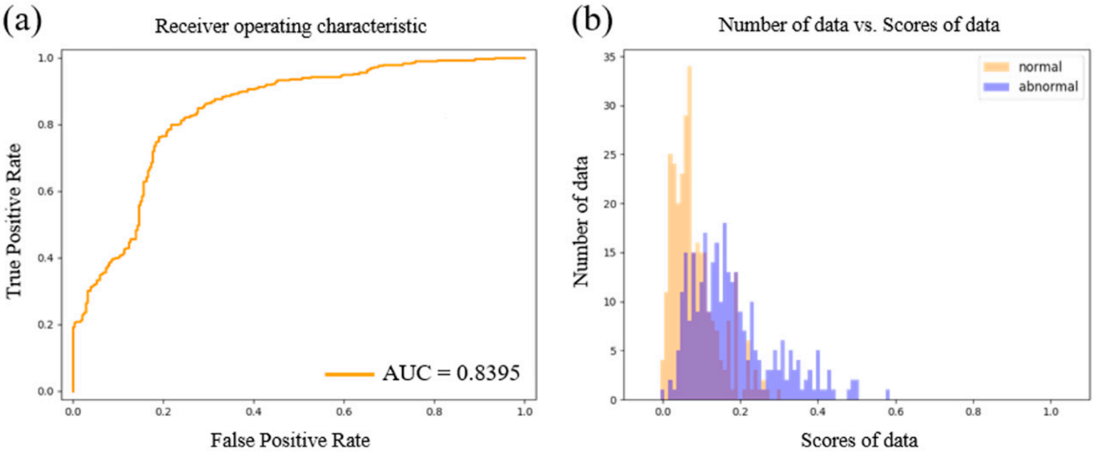

Figure 5. (**a**) The AUC performance using GANomaly-L1-L1; (**b**) The scores distribution under GANomaly-L1-L1.

The data distribution jitters in Figure 5a,b are caused by the limited data amount. This does not affect the overall performance.

3.2. Performance of GANomaly-SSIM-L1

Under GANomaly-SSIM-L1, mSSIM is used as the contextual loss function of the model. To improve the requirements for the image similarity through SSIM, based on the characteristics of image brightness, contrast, and structural differences, we resized the image resolution to 128 × 128 pixels and used mSSIM loss function to compare images after the parameter optimizations.

After the sufficient convergence, we obtained the better performance of the detection ability of AUC = 0.8801, as shown in Figure 6a. As seen in Figure 6b, the normal and anomalous score distributions are narrower than GANomaly-L1-L1, but the overlapping scores are lower. From these results, the impact of Problem 1, mentioned in Section 3.1, is reduced when the score function remains unchanged but is operating under a different model. The narrower score distribution is due to the better preservation of the individual normal features of the real image by the model so that the score distribution will be closer to the lower score during subsequent re-scoring. Compared with GANomaly-L1-L1, it is easier to identify abnormal blocks and generate fewer misjudgment events. Therefore, the overall performance is improved.

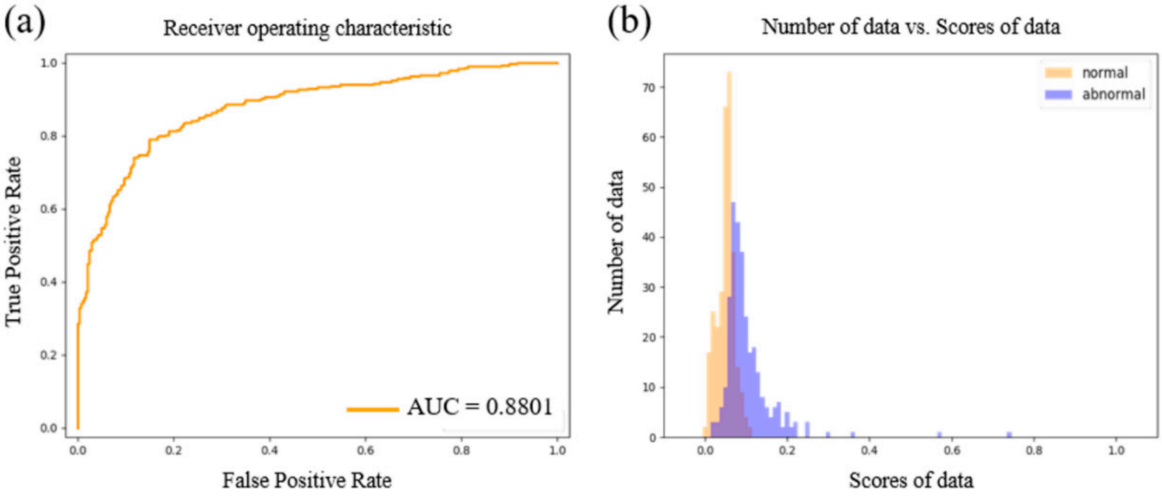

Figure 6. (**a**) The AUC performance using GANomaly-SSIM-L1; (**b**) The scores distribution under GANomaly-SSIM-L1.

3.3. Performance of GANomaly-SSIM-SSIM

Improvements were made in GANomaly-L1-SSIM, using the comparison ability of SSIM in the scoring function in this section. This was achieved by retaining the overall optimized parameters and only replacing the L1 score in the scoring function with mSSIM scores. We obtained a better solution, AUC = 0.9524, as shown in Figure 7a. It can be seen from Figure 7b that, compared with GANomaly-L1-SSIM, the overall score distribution becomes wider, and the overlapped part drops significantly. This phenomenon explains that SSIM compares the three factors the most, and the score judgment will be more delicately distinguished. This results in the score distribution widening toward the high score, making the positive and abnormal data more obvious. It also enhances the recognition ability.

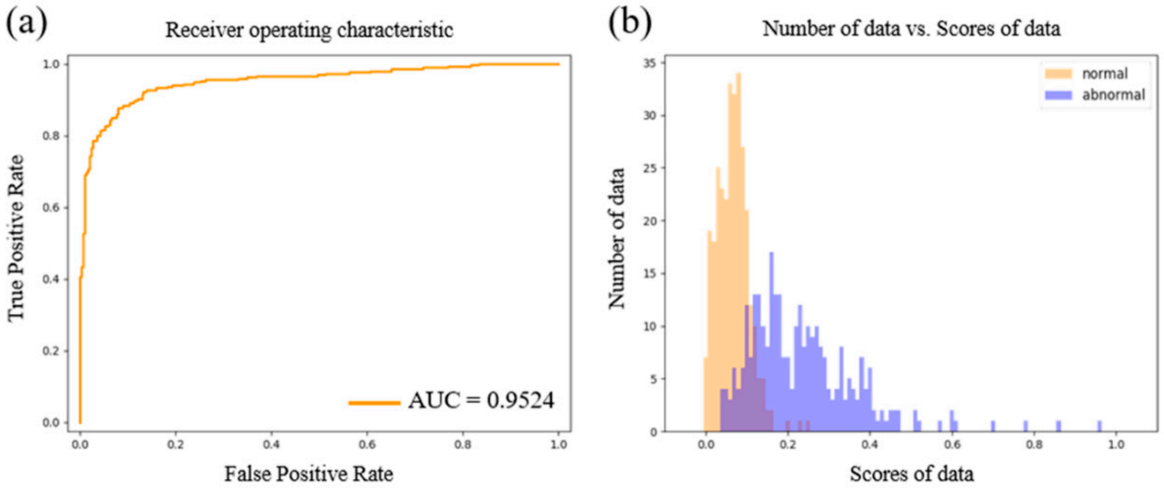

Figure 7. (**a**) The AUC performance using GANomaly-SSIM-SSIM; (**b**) The scores distribution under GANomaly-SSIM-SSIM.

3.4. Comparisons of Algorithms

As analyzed above, the restoration ability of GANomaly-L1-L1 is limited, and the restored images of L1 fake images are too consistent to make perfect judgments during the comparison. In comparison with GANomaly-L1-L1, GANomaly-SSIM-L1 and GANomaly-SSIM-SSIM not only restore the normal images but also retain the normal image features individually, which improves the accuracy.

To directly explain the performance by generated images, Figure 8 lists the real images, L1 fake images, L1 comparison results, SSIM fake images, and SSIM comparison results of the three defects.

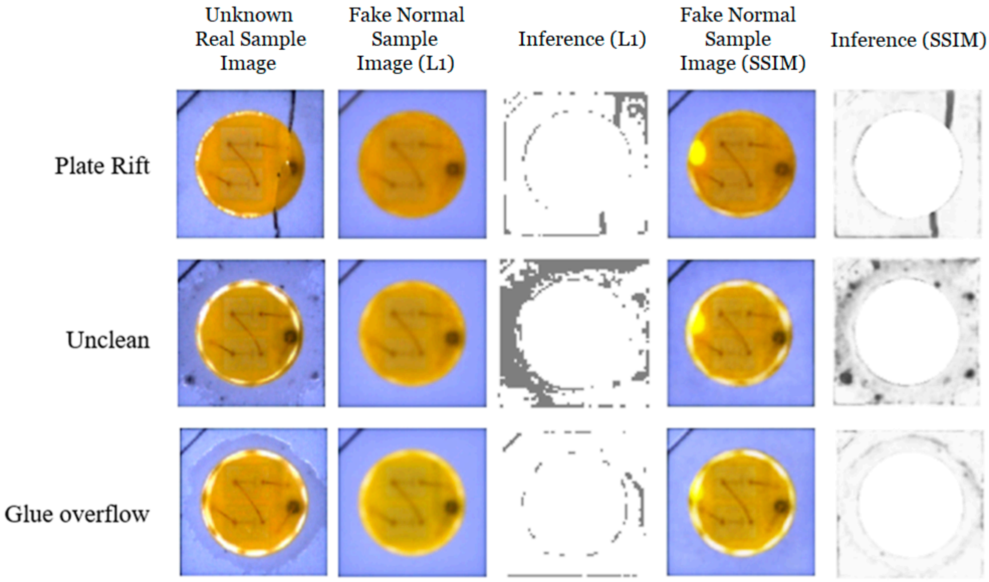
Unknown Real Sample Image
Fake Normal Sample Image (L1)
Inference (L1)
Fake Normal Sample Image (SSIM)
Inference (SSIM)
Plate Rift
Unclean
Glue overflow

Figure 8. The reconstruction and inference result of three algorithms.

After training the GANomaly model, pre-processing the sample image acquired from the CMOS camera, and inputting the image into the generative network, the generative network generates fake normal sample images depending on the loss functions (L1 or SSIM) they used.

Those real sample images of three different defect types (plate rift, unclean, and glue overflow) can be used to generate fake normal images, labeled "Fake Normal Image (L1)" and "Fake normal Image (SSIM)" in Figure 8, corresponding to the first loss functions (L1 or SSIM) that the generative networks adopted. The discriminators then use the second loss functions to compare the fake normal sample images and the real sample images, and identify the abnormal pixels. Following this, the algorithms output L1 comparison results, labeled Inference (L1) in Figure 8.

When it comes to the plate rift and unclean defects, it can be observed that the noises in Inference (SSIM) are fewer than in Inference (L1). Although the L1 comparison function can catch the defect positions when board cracks and stain defects are targeted, the over grabbing phenomenon of the non-defect positions is also quite serious. What is worse, the different values of pixels in sample images with defect glue overflow are very small, which makes the L1 difficult to detect. However, SSIM comparison function can perfectly grasp the defect locations when it comes to board cracks, stains, and glue overflow defects. That is, SSIM can effectively judge abnormal information during the scoring.

In addition to the ability to identify anomalies in these comparison algorithms, the calculation speed is an important indicator of the AOI system. The general industrial inspections require an image identification speed that is faster than 0.03 s per frame. All of the algorithms mentioned above can satisfy the requirements and, as shown in Table 3, the best performance is 0.0076 s per frame.

Table 3. The AUC and the operation time performances of three algorithms.

Algorithm	Size (Pixels)	AUC	AUC Improvement (%)	Inference (s)
GANomaly-L1-L1	64	0.8395	0	0.0027
GANomaly-SSIM-L1	128	0.8801	4.8	0.0075
GANomaly-SSIM-SSIM	128	0.9524	13.4	0.0076

3.5. Performance of Pre-Labeled Method

We propose the pre-labeled method to convert the comparison images generated by GANomaly into labeled files suitable for the supervised model. In this experiment, we provided the same unlabeled images and pre-labeled files to seven testers, and recorded the time they took to label the images, after the labor marking training. A tester can mark 1.06 unlabeled images per minute and 2.33 pre-labeled images per minute on average. As shown in Table 4, the marking speed can be increased by 21 times via the pre-labeled method.

Table 4. The labeling speed under labor labeling and GANomaly-SSIM-SSIM.

Labeling	Number of Labels per Minute	Enhancement
Labor	3	–
GANomaly-SSIM-SSIM	63	21×

According to Figure 9, different markers possess different marking qualities for the same image. For instance, Tester 1 marked quite completely as soon as possible, Tester 2 marked too much area and abbreviated, Tester 3 incorrectly marked the defect location, and Tester 4 marked the wrong mark. The unstable marking quality directly affects the inspection ability of the supervised model. With the pre-labeled method, the labeling personnel only has to make a quick judgment and minor repairs based on the labeling data generated by GANomaly, and then feed the model training, thereby reducing the labeling labor costs. Moreover, the labeled data generated by GANomaly are consistent and the unstable marking quality will not occur, improving the ability of the subsequent supervised model to identify defects.

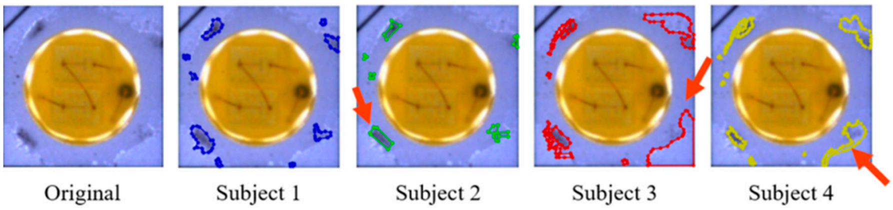

Figure 9. The labeling condition within four different subjects, the red arrows imply bad labeling by labor.

4. Conclusions

In this paper, we proposed an AOI defect detection system with a supervised model assisted by an unsupervised model. The model adopts GANomaly and imports structural similarity indicators to strengthen the original model. The detection rate is increased by 13.4%, and the AUC reaches 0.9524. We also proposed the pre-labeled method for GANomaly to automatically generate labeled data files to assist in supervised model training. Compared with direct labeling, the generation speed is increased by 21 times. Moreover, the stability and the completeness of data are guaranteed, which enhances the training effectiveness of the supervised model. To achieve the purpose of smart factories, we integrated the algorithms and the pre-labeled method into a complete AOI system, which can handle the detection of LED surface defects and can also provide abnormal alarm prompts and rapid responses.

Author Contributions: Conceptualization, C.-H.H. and S.-H.C.; Data curation, C.-H.H. and C.-L.T.; Formal analysis, C.-H.H.; Methodology, C.-H.H.; Project administration, C.-H.H.; Resources, S.-H.C.; Software, C.-H.H. and C.-L.T.; Supervision, H.-C.K., C.-W.S., C.-C.L. and X.L.; Validation, C.-H.H., P.-H.L., S.-H.C. and H.-C.K.; Visualization, S.-H.C.; Writing—original draft, P.-H.L. and C.-L.T.; Writing—review and editing, P.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We would like to acknowledge the support from Leda Technology Corporation Limited, Yung Chi Tsao, Tony Yang, and Takhellambam Gautam Meitei.

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

References

1. Perera, P.; Patel, V.M. Learning deep features for one-class classification. *IEEE T Image Process* **2019**, *28*, 5450–5463. [CrossRef] [PubMed]
2. Kim, Y.K.; Lee, S.Y.; Seo, S.; Lee, K.M. Fuzzy logic-based outlier detection for bio-medical data. In Proceedings of the 2014 International Conference on Fuzzy Theory and Its Applications (iFUZZY2014), Kaohsiung, Taiwan, 26–28 November 2014; pp. 117–121.
3. Rabiner, L.R. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [CrossRef]
4. YOLOv3: An Incremental Improvement. Available online: http://arxiv.org/abs/1804.02767 (accessed on 27 July 2020).
5. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
6. Liu, W. SSD: Single shot multiBox Detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV2016), Amsterdam, The Netherlands, 8–16 October 2016; 9905, pp. 21–37.
7. Goodfellow, I.J. Generative adversarial networks. In Proceedings of the twenty-eighth Conference on Neural Information Processing Systems (NeurIPS2014), Montréal, Canada, 8–13 December 2014; pp. 2672–2680.
8. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
9. Bergmann, P.; Löwe, S.; Fauser, M.; Sattlegger, D.; Steger, C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *Visigrapp* **2019**, *5*, 372–380.
10. Jinwon, A.; Sungzoon, C. Variational autoencoder based anomaly detection using reconstruction probability. *Spec. Lect. IE* **2015**, *20*, 404–406.

11. Akcay, S.; Abarghouei, A.A.; Breckon, T.P. GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training. In Proceedings of the 14th Asian Conference on Computer Vision (ACCV2018), Perth, Australia, 2–6 December 2018; 11363, pp. 622–637.
12. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *Lect. Notes Comput. Sci.* **2017**, *10265*, 146–147.
13. Zenati, H.; Foo, C.S.; Lecouat, B.; Manek, G.; Chandrasekhar, V.R. Efficient GAN-based anomaly detection. *arXiv* **2018**, arXiv:1802.06222.
14. Akcay, S.; Atapour-Abarghouei, A.; Breckon, T. Skip-GANomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; p. 8851808.
15. MNIST Handwritten Digit Database, Yann LeCun, Corinna Cortes and Chris Burges. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 27 August 2020).
16. CIFAR-10 and CIFAR-100 Datasets. Available online: https://www.cs.toronto.edu/~{}kriz/cifar.html (accessed on 27 August 2020).
17. Threat Image Projection Library Software—GOV.UK. Available online: https://www.gov.uk/government/publications/threat-image-projection-library-software (accessed on 27 August 2020).
18. Rau, H.; Wu, C.H. Automatic optical inspection for detecting defects on printed circuit board inner layers. *Int. J. Adv. Manuf. Technol.* **2005**, *25*, 940–946. [CrossRef]
19. Cheng, S.H.; Perng, D.B. Automatic optical inspection system for IC molding surface. *J. Intell. Manuf.* **2016**, *27*, 915–926.
20. Hecht, O.; Dishon, G. Automatic optical inspection (AOI). In Proceedings of the 40th Conference Proceedings on Electronic Components and Technology, Las Vegas, NV, USA, 20–23 May 1990; pp. 659–661.