

Article

# Generating Defective Epoxy Drop Images for Die Attachment in Integrated Circuit Manufacturing via Enhanced Loss Function CycleGAN

Lamia Alam <sup>\*</sup>  and Nasser Kehtarnavaz 

Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080, USA  
<sup>\*</sup> Correspondence: lamia.alam@utdallas.edu

**Abstract:** In integrated circuit manufacturing, defects in epoxy drops for die attachments are required to be identified during production. Modern identification techniques based on vision-based deep neural networks require the availability of a very large number of defect and non-defect epoxy drop images. In practice, however, very few defective epoxy drop images are available. This paper presents a generative adversarial network solution to generate synthesized defective epoxy drop images as a data augmentation approach so that vision-based deep neural networks can be trained or tested using such images. More specifically, the so-called CycleGAN variation of the generative adversarial network is used by enhancing its cycle consistency loss function with two other loss functions consisting of learned perceptual image patch similarity (LPIPS) and a structural similarity index metric (SSIM). The results obtained indicate that when using the enhanced loss function, the quality of synthesized defective epoxy drop images are improved by 59%, 12%, and 131% for the metrics of the peak signal-to-noise ratio (PSNR), universal image quality index (UQI), and visual information fidelity (VIF), respectively, compared to the CycleGAN standard loss function. A typical image classifier is used to show the improvement in the identification outcome when using the synthesized images generated by the developed data augmentation approach.

**Keywords:** vision-based die attachment inspection; synthesized defective epoxy drop images; enhanced loss function CycleGAN



**Citation:** Alam, L.; Kehtarnavaz, N. Generating Defective Epoxy Drop Images for Die Attachment in Integrated Circuit Manufacturing via Enhanced Loss Function CycleGAN. *Sensors* **2023**, *23*, 4864. <https://doi.org/10.3390/s23104864>

Academic Editors: Xin Ning and Wenfa Li

Received: 11 April 2023

Revised: 9 May 2023

Accepted: 17 May 2023

Published: 18 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

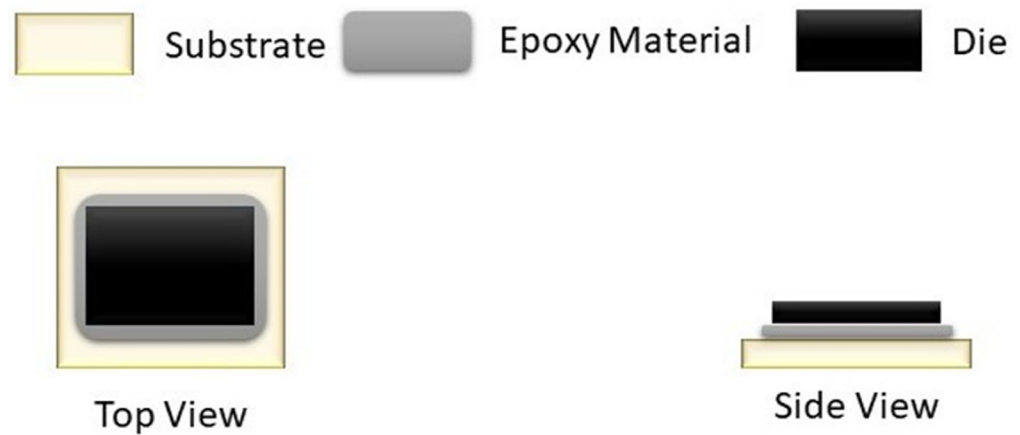
## 1. Introduction

In integrated circuit (IC) manufacturing, vision- or camera-based inspection systems are often used to enable automatic inspection of defects encountered during IC production. There are three primary components associated with such systems: a camera sensor, a computer running the inspection algorithms, and a sorter to separate defective and non-defective ICs. In this paper, defective epoxy drop images for die attachment are studied using die substrate images captured by a camera.

Die attachment is a crucial step in the production of ICs. It is also referred to as die bonding or die mounting and involves attaching a silicon wafer die to a die pad or a substrate. Adhesive die attachments are most widely used due to their low cost [1]. In order to create a bond between a die and a substrate, an adhesive die attachment, as its name implies, uses an adhesive or epoxy. An epoxy die attachment equipment/die bonder is normally used to attach a die to its substrate using epoxy adhesive [2]. As illustrated in Figure 1, the die is set on top of the epoxy drop that is placed on the substrate and is bonded by heating.

A difficult aspect of die attachment is attaching the die to the substrate with the least amount of epoxy possible. An excessive amount of epoxy could cause the die or chip to tilt or overflow the substrate, compromising the stability of the entire IC package. On the other hand, an inadequate amount of epoxy could cause a bond line to become too thin, creating a weak bond with insufficient mechanical strength, which could lead to die cracking or

die lifting. It is therefore necessary to inspect epoxy drops for proper die attachment. Due to the time and cost involved in conducting the inspection manually, the inspection process has been automated in modern IC manufacturing. The inspection process needs to be computationally efficient with high accuracy so that it is deployable in an actual production line.



**Figure 1.** Epoxy die attachment.

A vision-based deep learning system can be utilized to conduct the inspection process automatically by examining images of an epoxy drop to assess whether a proper amount has been used. Depending on predefined quality standards, the system would accept or reject the die in real time. The use of modern deep neural networks requires the availability of a very large number of images with both adequate and inadequate amounts of epoxy. In practice, only a few images with inadequate amounts of epoxy, named defective or rejected epoxy images, are available since such cases occur quite infrequently. A good performance is normally reached when the training data are equally balanced for non-defective and defective image samples. Furthermore, although the lack of defective epoxy images can be mitigated by using conventional data augmentation techniques (such as cropping, rotating, flipping, and translating) [3], these techniques do not provide the image diversity needed for adequate training or testing of deep neural networks. The testing situation arises when only non-defective images are used for training a deep neural network and the generated defective images are used to test the trained network.

Generative adversarial network (GAN) models are being increasingly used for data augmentation purposes [4]. Many variations of GANs have been introduced in recent years. A variation of a GAN called CycleGAN [5] has been used to synthesize realistic images in different applications. In this paper, the CycleGAN standard loss function is modified or enhanced by utilizing learned perceptual image patch similarity (LPIPS) [6] and a structural similarity index metric (SSIM) [7] in order to generate more realistic defective or rejected epoxy images. Different combinations of LPIPS and the SSIM with/without the standard loss function are examined.

A data augmentation method for generating high-quality rejected epoxy images is introduced in this paper by enhancing the standard loss function CycleGAN via incorporating LPIPS and SSIM image quality metrics. This augmentation approach enables vision-based deep neural networks to be trained or tested more effectively by having equal numbers of defective and non-defective images. It is further shown that this data augmentation of synthesized defective images leads to improved identification outcomes.

The rest of the paper is organized as follows. Section 2 provides a review of previous works related to GAN models for data augmentation applications. An overview of CycleGAN is then covered in Section 3 together with the introduced enhanced loss function to address the die attachment problem of interest here. The experimental results in terms of quantitative evaluation metrics are then presented in Section 4 for the CycleGAN standard loss function and the enhanced loss function. The paper is finally concluded in Section 5.

## 2. Previous Works on Data Augmentation Using a GAN

This section provides a summary of data augmentation performed for different applications using a GAN. A GAN was used to produce many versions of an image in [8]. A review of medical image augmentation papers using a GAN was covered in [9]. A pixel-level image augmentation technique was developed in [10] based on image-to-image translation with a GAN. It was trained on a surface defect dataset of magnetic particle images to generate synthesized image samples. A variation of CycleGAN named AttenCGAN was proposed in [11] to synthesize electrical commutators and surface images with artificial defects to increase the number of image samples. A defect transfer GAN (DT-GAN) was developed in [12] to produce realistic surface defect images. The Mask2Defect GAN was suggested in [13] to create surface defect images obtained from an automobile part stamping plan. A region- and strength-controllable GAN for creating synthesized defects in metal surfaces was also proposed in [14] based on the idea of image inpainting. To produce high-quality defect images, a so-called relative mean generative adversarial network (TARGAN) was introduced in [15] using a metal gear surface defect image dataset and a hot-rolling strip defect image dataset. In [16], an MAS-GAN-based model was proposed for the production of industrial defect images by combining an attention mechanism and a data augmentation module. A framework called DefectGAN was introduced in [17] by using a compositional-layer-based architecture to generate realistic defect images. For data augmentation of surface defects on hot-rolled steel strips, three GANs were trained in [18], a new GAN called a contrastive GAN was proposed in [19], and a semi-supervised learning (SSL) defect classification approach based on two different networks of a categorized generative adversarial network (GAN) and a residual network was proposed in [20]. In order to produce defect images using a large number of defect-free images of commutator cylinder surfaces from industrial sites, a generation technique known as the surface defect generation adversarial network (SDGAN) was introduced in [21]. GAN models were also utilized for unsupervised surface inspection in [22], for anomaly detection on structured and arbitrary textured surfaces in [23], for Mura defect classification in thin-film transistor liquid crystal display (TFT-LCDs) in [24], for enhancing the quantity and quality of images of fabric defects in [25], for the autonomous design of architectural shape sketches in [26], and for establishing the probabilistic correlations of quasi-static responses of bridges in [27].

More recently, GANs have been used for IC manufacturing applications. A multi-scale GAN with a transformer (MST-GAN) as a semi-supervised deep learning network was developed for IC metal package samples in [28]. An IC solder joint inspection approach was suggested in [29] based on a GAN model and statistical training. In [30], a GAN model was used to generate pseudo-defective wafer die images from real defective images. A GAN-based image generation technique for organic light-emitting diode (OLED) panel defect images was discussed in [31].

In this paper, our objective is to produce realistic defective or rejected epoxy substrate images based on the few available such images. To meet this objective, we make use of a large number of defect-free or good epoxy substrate images that are available in order to generate a large number of defective or rejected epoxy substrate images by using CycleGAN to translate non-defective or good images to rejected images. For this purpose, we enhanced the CycleGAN cycle consistency loss function by incorporating other loss functions, which are discussed in Section 3.

## 3. Generating Synthesized Defective Images via CycleGAN

The most widely used conditional generative adversarial network for the purpose of unpaired image-to-image translation is called CycleGAN [5]. A typical CycleGAN learns the mapping between two distributions via optimization of an objective function by using two generators and two discriminators. Two losses are incorporated into the CycleGAN optimization framework: adversarial loss and cycle consistency loss. The adversarial loss measures the difference between the generated images and the target images according to the original GAN design [4], and the cycle consistency loss is used to avoid conflicts

between the learnt mappings. In our problem, we generate synthesized defective or rejected epoxy drop substrate images from non-defective or good epoxy drop substrate images. Despite the fact that the CycleGAN generates realistic synthetic images, ambiguity mapping occurs when a domain with rich information (i.e., good epoxy drop substrate images) is translated into a domain with relatively weak information (i.e., rejected epoxy drop substrate images). This ambiguity mapping is addressed in this paper by adding the loss functions of LPIPS and the SSIM to the standard cycle consistency loss function of the generator network. More details of these loss functions are stated later in this section.

Two generator networks and two discriminator networks make up the CycleGAN architecture [5]. Adversarial training is carried out on the networks against one another. The generators' objective is to convert an image from one domain to another. The discriminators' objective is to distinguish between real and synthesized images in their respective domains. Figure 2 demonstrates the CycleGAN framework that we utilized to achieve data augmentation of defective or rejected epoxy drop substrate images. The data augmentation model contains the two mapping functions  $G_{g \rightarrow r} : \text{Good} \rightarrow \text{Rejected}$  and  $G_{r \rightarrow g} : \text{Rejected} \rightarrow \text{Good}$ , and the associated adversarial discriminators  $D_r$  and  $D_g$ . Here,  $D_r$  distinguishes between real rejected epoxy drop substrate images  $\{I_r\}$  and synthesized rejected epoxy drop substrate images  $\{I_r^s\}$  generated from real good epoxy drop substrate images  $\{I_g\}$ , and likewise  $D_g$  distinguishes between  $\{I_g\}$  and  $\{I_g^s\}$ . The total loss function of the CycleGAN can be expressed as a summation of the adversarial losses ( $\mathcal{L}_{advers}$ ) and the cycle consistency loss ( $\mathcal{L}_{cyc}$ ):

$$\mathcal{L}(G_{g \rightarrow r}, G_{r \rightarrow g}, D_g, D_r) = \mathcal{L}_{advers}(G_{g \rightarrow r}, D_r) + \mathcal{L}_{advers}(G_{r \rightarrow g}, D_g) + \mathcal{L}_{cyc}(G_{g \rightarrow r}, G_{r \rightarrow g}) \quad (1)$$

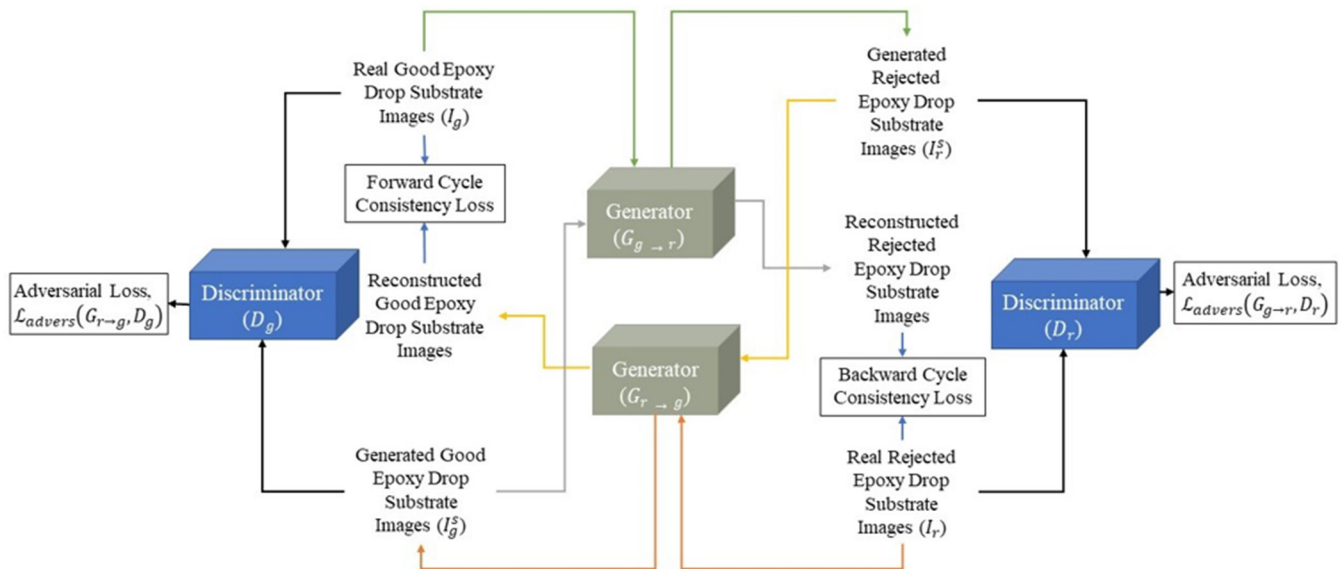


Figure 2. CycleGAN framework used for our data augmentation.

Adversarial losses make sure that the generated images appear realistic, and the cycle consistency loss reflects the difference between the original image and the reconstructed or transformed image. The aim is to solve the following optimization problem

$$G_{(g \rightarrow r)}^*, G_{(r \rightarrow g)}^* = \arg \min_{G_{g \rightarrow r}, G_{r \rightarrow g}} \max_{D_g, D_r} \mathcal{L}(G_{g \rightarrow r}, G_{r \rightarrow g}, D_g, D_r) \quad (2)$$

More details of the loss functions are stated next.

**Adversarial loss function:** For both the mapping functions, the adversarial losses in [4] adopted by the standard Cycle GAN are used. For the mapping function  $G_{g \rightarrow r}$  and its discriminator  $D_r$ , the optimization problem can be written as

$$\mathcal{L}_{advers}(G_{g \rightarrow r}, D_r, I_g, I_r) = \mathbb{E}_{i_r \sim p_{data}(i_r)} [\log D_r(i_r)] + \mathbb{E}_{i_g \sim p_{data}(i_g)} [\log (1 - D_r(G_{g \rightarrow r}(i_g)))] \quad (3)$$

where  $i_r \sim p_{data}(i_r)$  and  $i_g \sim p_{data}(i_g)$  denote the distributions of  $I_r$  and  $I_g$ , respectively, and  $\mathbb{E}$  denotes the expected value over all real data instances. Here,  $G_{g \rightarrow r}$  attempts to generate images  $G_{g \rightarrow r}(i_g)$  that look like  $I_r$  images, while  $D_r$  attempts to distinguish between synthesized images  $I_r^s \approx G_{g \rightarrow r}(i_g)$  and real images  $I_r$ . Similarly, for the mapping function  $G_{r \rightarrow g}$  and its discriminator  $D_g$ , the optimization problem can be written as

$$\mathcal{L}_{advers}(G_{r \rightarrow g}, D_g, I_r, I_g) = \mathbb{E}_{i_g \sim p_{data}(i_g)} [\log D_g(i_g)] + \mathbb{E}_{i_r \sim p_{data}(i_r)} [\log (1 - D_g(G_{r \rightarrow g}(i_r)))] \quad (4)$$

**Cycle consistency loss:** Cycle consistency loss converts images back to their original domain, i.e.,  $i_g \rightarrow G_{g \rightarrow r}(i_g) \rightarrow G_{r \rightarrow g}(G_{g \rightarrow r}(i_g)) \approx i_g$ , known as the forward cycle consistency loss, and  $i_r \rightarrow G_{r \rightarrow g}(i_r) \rightarrow G_{g \rightarrow r}(G_{r \rightarrow g}(i_r)) \approx i_r$ , known as the backward cycle consistency loss. Cycle consistency loss is defined as the combination of the following losses:

$$\mathcal{L}_{Cycle}(G_{r \rightarrow g}, G_{g \rightarrow r}) = \mathcal{L}^{F\_Cycle}(G_{g \rightarrow r}) + \mathcal{L}^{B\_Cycle}(G_{r \rightarrow g}) \quad (5)$$

where  $\mathcal{L}^{F\_Cycle}$  and  $\mathcal{L}^{B\_Cycle}$  represent the forward and backward cycle consistency losses, respectively. The cycle consistency loss makes sure that the features of the input images are preserved in the generated images. For the cycle consistency loss, it matters which loss function is used for the cycle consistency loss. In this work, we consider several loss functions separately and in combination for the cycle consistency loss in order to improve the CycleGAN performance for the die attachment problem of interest here. When combining the loss functions, they are normalized so that their contributions to the combined loss function are made equal. A description of the loss functions considered is presented next.

**$L_1$  loss function:** Additionally, referred to as mean absolute error (MAE) loss, measures the absolute distance between the generated image and the target image. In our case, it is obtained by taking the absolute value of the real good image  $I_g$  and the reconstructed good image  $G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))$ . It can be expressed as follows:

$$\mathcal{L}_{L_1}^{F\_Cycle}(i_g, G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))) = \mathbb{E}_{i_g \sim p_{data}(i_g)} [\|i_g - G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))\|_1] \quad (6)$$

Similarly, the absolute value of the real rejected image  $I_r$  and the reconstructed good image  $G_{g \rightarrow r}(G_{r \rightarrow g}(i_r))$  is obtained as follows:

$$\mathcal{L}_{L_1}^{B\_Cycle}(i_r, G_{g \rightarrow r}(G_{r \rightarrow g}(i_r))) = \mathbb{E}_{i_r \sim p_{data}(i_r)} [\|i_r - G_{g \rightarrow r}(G_{r \rightarrow g}(i_r))\|_1] \quad (7)$$

$L_1$  loss function reduces the absolute difference between the images. It is mostly used to capture low-frequency details or to enforce the accuracy of low frequencies. It has been used to compute the cycle consistency loss in the standard CycleGAN [5]. The quality of the generated images can be improved by combining this kind of loss function with another loss function, as discussed in [32].

**$L_2$  loss function:** Additionally, referred to as mean squared error (MSE), is obtained by squaring the difference between the generated image and the target image. In our case, it can be expressed as follows:

$$\mathcal{L}_{L_2}^{F\_Cycle}(i_g, G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))) = \mathbb{E}_{i_g \sim p_{data}(i_g)} [\|i_g - G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))\|_2^2] \quad (8)$$



$$\mathcal{L}_{L_2}^{B\_Cycle}(i_r, G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))) = \mathbb{E}_{i_r \sim p_{data}(i_r)} \left[ \left\| i_r - G_{g \rightarrow r}(G_{r \rightarrow g}(i_r)) \right\|_2^2 \right] \quad (9)$$

In [33],  $L_1$  and  $L_2$  losses were compared and no discernible difference between them was found. However, according to [34],  $L_1$  loss is preferred over  $L_2$  loss because it promotes less blurring. Both of these losses represent pixel-wise losses. They consider pixel-by-pixel variations between the images. Even though the images being compared are comparable to the human visual system, there exists a loss in value. Since the computation is based on each pixel, it is not significantly affected if one shifts an image by just one pixel. The total loss value gradually rises as a result of the aggregation of each minor difference between the corresponding pixels of two images. Hence, adding some other loss function to the standard loss functions can help to improve the performance of the model.

**Structural similarity index metric (SSIM):** This index has been extensively utilized to assess image quality [7] and has been used as loss function for numerous image processing applications [35,36] as well as for GAN-based solutions [32,37,38]. It was created under the presumption that the human visual system is extremely well suited for sifting through structural data in a visual input. The structural information degradation between a generated image and a corresponding input image is measured by the SSIM. Luminance, contrast, and structure are three sub-indices that make up the SSIM. Luminance is reflected in the local means, contrast in the local standard deviations, and structure in the local Pearson correlation between two images. For an input image  $x$  and a reconstructed image  $y$ , the SSIM is defined as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (10)$$

where  $\mu_x$  and  $\mu_y$  are the mean intensities,  $\sigma_x$  and  $\sigma_y$  are the variances, and  $\sigma_{xy}$  is the covariance of images  $x$  and  $y$ . The constants  $C_1$  and  $C_2$  are used to prevent numerical singularity. More information on this index appears in [7]. Here, the SSIM is included as a loss function to produce visually acceptable images. This index can be expressed as follows:

$$\mathcal{L}_{SSIM}^{F\_Cycle}(i_g, G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))) = [1 - SSIM(i_g, G_{r \rightarrow g}(G_{g \rightarrow r}(i_g)))] \quad (11)$$

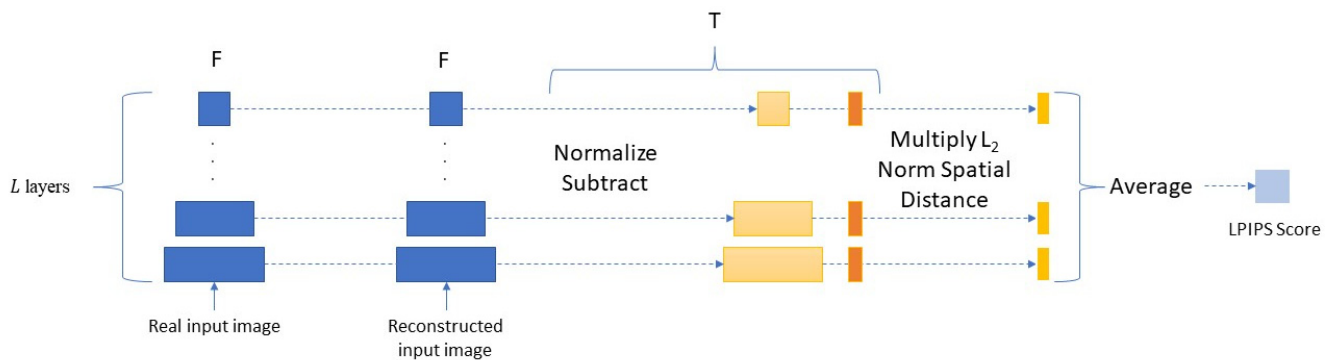
$$\mathcal{L}_{SSIM}^{B\_Cycle}(i_r, G_{g \rightarrow r}(G_{r \rightarrow g}(i_r))) = [1 - SSIM(i_r, G_{g \rightarrow r}(G_{r \rightarrow g}(i_r)))] \quad (12)$$

**Learned perceptual image patch similarity (LPIPS):** LPIPS indicates how similar two images appear to the human eye. In essence, LPIPS determines how comparable the activations of two image patches are for a given network. Therefore, we use it here as a loss function. Figure 3 and Equations (13) and (14) show how a pertained network  $F$  is used to compute the LPIPS score between a real input image and a reconstructed image.

$$\mathcal{L}_{LPIPS}^{F\_Cycle}(i_g, G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))) = \sum_l T^l \left( \left\| F^l(i_g) - F^l(G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))) \right\|_2^2 \right) \quad (13)$$

$$\mathcal{L}_{LPIPS}^{B\_Cycle}(i_r, G_{g \rightarrow r}(G_{r \rightarrow g}(i_r))) = \sum_l T^l \left( \left\| F^l(i_r) - F^l(G_{g \rightarrow r}(G_{r \rightarrow g}(i_r))) \right\|_2^2 \right) \quad (14)$$

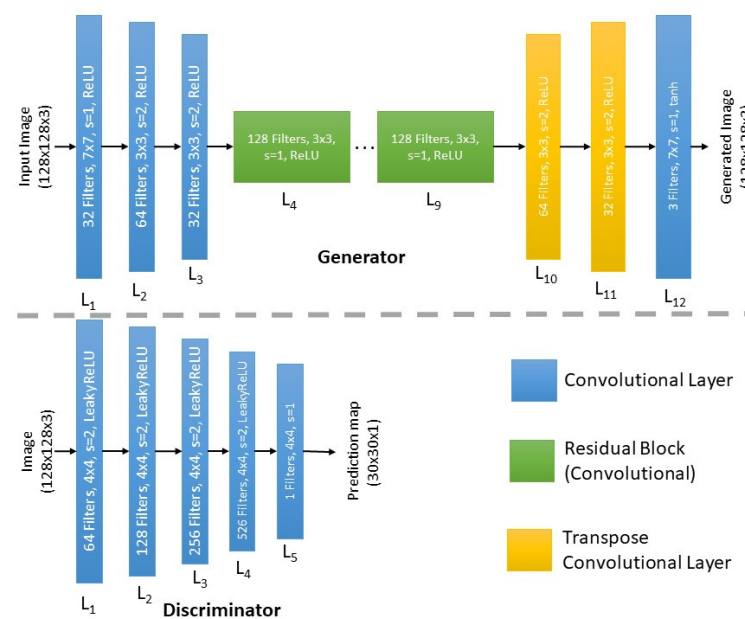
where  $F$  denotes the pertained network with  $l \in L$  layers for feature extraction, and  $T$  normalizes and scales the deep embedding to a scalar LPIPS score. Then, the  $L_2$  distance is computed and averaged across the dimensions and layers of the network. For feature distances, the AlexNet [39] network is used here, which is more in line with the structure of the human visual cortex [6,40].



**Figure 3.** Network to compute LPIPS.

### Model Architecture and Training

For the generator and discriminator, the same architecture described in the standard CycleGAN [5] is used here; see Figure 4. The generator feeds its  $128 \times 128$  input image through three convolutional layers in succession, each of which causes the representation to become smaller with more channels. Afterwards, a set of six residual blocks follows, each with 128 filters. Transpose convolutional layers are used to further enhance the representation for the final image. Apart from the Tanh activation in the last layer for reconstruction, each layer is followed by instance normalization and a rectified linear unit (ReLU) as the activation function. The generated image is  $128 \times 128$  in size. The Markovian discriminator (PatchGAN) [41] is utilized to determine if the image patches are real or synthesized for the discriminator. Five convolutional layers make up the discriminator, which is a fully convolutional network. In order to keep the size of the feature maps at  $1/8$ , the stride is only set to 2 for the first four convolutional layers and the instance normalization along with Leaky ReLU are utilized as the activation function. To preserve the size of the feature maps, the stride of the final output layers is set to 1 and the filter number is set to 1 in order to produce a one-channel prediction map with values ranging from 0 to 1 for every pixel. The discriminator's input is a real or synthesized image having a size of  $128 \times 128$ , and the output is  $30 \times 30$  in size. In order to determine if a patch of the input image is real or synthesized, each output pixel corresponds to a patch of the input image.



**Figure 4.** Architecture of the generator and discriminator of our implemented CycleGAN.

After defining each component of the CycleGAN, the network is trained. A pseudo code of the training is shown in Algorithm 1. CycleGAN offers a clear advantage over utilizing unpaired data. However, our image-generating network is trained using the paired real good epoxy drop substrate images and real rejected epoxy drop substrate images in order to test the generated images in a uniform manner. The batch size is set to 1 with 200 epochs and  $k = 100$ , which is sufficient for convergence. The model is optimized using the Adam optimizer with  $\beta_1 = 0.5$  and an initial learning rate of 0.0002 for the first 100 epochs, decreasing the learning rate linearly to 0 for the last 100 epochs. Our CycleGAN with different cycle consistency losses is implemented in Python (environment version 3.8.13) using the TensorFlow (version 2.9.1) and Pillow (version 9.2.0) libraries.

---

**Algorithm 1.** Cycle GAN Training
 

---

1: for number of epochs do

2: for k iterations do

3: Draw a minibatch of samples  $\{i_g^{(1)}, \dots, i_g^{(m)}\}$  from data distribution,  $p_{data}(i_g)$  domain  $I_g$

4: Draw a minibatch of samples  $\{i_r^{(1)}, \dots, i_r^{(m)}\}$  from data distribution,  $p_{data}(i_r)$  domain  $I_r$

5: Generate m synthetic samples

$$I_g^s : I_g \rightarrow G_{g \rightarrow r}(i_g)$$

$$I_r^s : I_r \rightarrow G_{r \rightarrow g}(i_r)$$

6: Compute adversarial losses

// Combination of discriminator loss on both real and fake images

$$\mathcal{L}_{advers}(G_{g \rightarrow r}, D_r, I_g, I_r) = \mathbb{E}_{i_r \sim p_{data}(i_r)}[\log D_r(i_r)] + \mathbb{E}_{i_g \sim p_{data}(i_g)}[\log(1 - D_r(G_{g \rightarrow r}(i_g)))]$$

$$\mathcal{L}_{advers}(G_{r \rightarrow g}, D_g, I_r, I_g) = \mathbb{E}_{i_g \sim p_{data}(i_g)}[\log D_g(i_g)] + \mathbb{E}_{i_r \sim p_{data}(i_r)}[\log(1 - D_g(G_{r \rightarrow g}(i_r)))]$$

7: Update the discriminators  $D_g$  and  $D_r$

$$\max_{D_g} \mathcal{L}_{advers}(G_{r \rightarrow g}, D_g, I_r, I_g)$$

$$\max_{D_r} \mathcal{L}_{advers}(G_{g \rightarrow r}, D_r, I_g, I_r)$$

8: Generate m cycle samples

$$I_g^{Cycle} : G_{g \rightarrow r}(i_g) \rightarrow G_{r \rightarrow g}(G_{g \rightarrow r}(i_g))$$

$$I_r^{Cycle} : G_{r \rightarrow g}(i_r) \rightarrow G_{g \rightarrow r}(G_{r \rightarrow g}(i_r))$$

9: Compute Cycle Consistency Loss

$$\mathcal{L}_{Cycle}(G_{r \rightarrow g}, G_{g \rightarrow r}) = \mathcal{L}^{F-Cycle}(G_{g \rightarrow r}) + \mathcal{L}^{B-Cycle}(G_{r \rightarrow g})$$

/\* Different loss functions separately and in combination were used to calculate the cycle consistency losses  $\mathcal{L}^{F-Cycle}(G_{g \rightarrow r})$  and  $\mathcal{L}^{B-Cycle}(G_{r \rightarrow g})$  \*/

10: Compute total generator loss

$$\mathcal{L}(G_{g \rightarrow r}, G_{r \rightarrow g}, D_g, D_r) = \mathcal{L}_{advers}(G_{g \rightarrow r}, D_r, I_g, I_r) + \mathcal{L}_{advers}(G_{r \rightarrow g}, D_g, I_r, I_g) + \lambda \mathcal{L}_{Cyc}(G_{g \rightarrow r}, G_{r \rightarrow g})$$

11: Update the generators  $G_{g \rightarrow r}$  and  $G_{r \rightarrow g}$

$$\min_{G_{g \rightarrow r}, G_{r \rightarrow g}} \mathcal{L}(G_{g \rightarrow r}, G_{r \rightarrow g}, D_g, D_r)$$


---

## 4. Experimental Results and Discussion

In this section, first we report our experiments to assess the quality of the generated images based on our introduced enhanced loss function CycleGAN. Then, we report our experiments using a typical image classifier (ResNet18) [42] to show the identification outcomes with and without using the generated synthesized defective or rejected images.

Our experiments were performed on a server with the 64-bit Windows 10 operating system with two Intel® Xeon® 2.40 GHz CPUs and with two NVIDIA Tesla K40 m graphics cards having 256 GB RAM.

### 4.1. Quantitative Evaluation of Generated Images

#### 4.1.1. Evaluation Metrics

Our goal is to create high-quality synthesized rejected epoxy drop substrate images. This requires quantitative quality evaluation metrics of the synthesized images. The metrics that are often used for this purpose include peak signal-to-noise ratio (PSNR) [7], the



universal image quality index (UQI) [43], and visual information fidelity (VIF) [44]. The testing dataset consists of paired images of the same size.

**Peak signal-to-noise ratio (PSNR):** Given its simplicity and ease of use, the PSNR is the most widely used metric for evaluating synthesized images. The PSNR indicates the difference in pixels between a synthesized and a real image. The quality of the resulting image improves with an increasing PSNR. Equation (15) indicates how the PSNR is computed:

$$PSNR = 20 \log_{10} \left( \frac{Peak\ Value^2}{MSE} \right) \quad (15)$$

where *Peak Value* denotes the highest value in the image data, and for an 8-bit unsigned integer data type, it is 255. The mean squared error (MSE) between two images is given by

$$MSE = \frac{1}{NN} \sum_{i=1}^N \sum_{j=1}^N (x(i, j) - y(i, j))^2 \quad (16)$$

with  $x$  and  $y$  representing, respectively, the real and the synthesized images of size  $N \times N$ . Equation (15) reflects the absolute error in dB.

**Universal image quality index (UQI):** The UQI compares generated synthesized and real images in terms of luminance, contrast, and structure, reflecting the characteristics of the human visual system. It corresponds to the special case of the SSIM when  $C_1 = C_2 = 0$  in Equation (10) and can be written as the product of three components of correlation, luminance distortion, and contrast distortion, as follows:

$$UQI = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (17)$$

where  $x = \{x_i | i = 1, 2, \dots, N\}$  and  $y = \{y_i | i = 1, 2, \dots, N\}$  denote the real and the synthesized images, respectively,  $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ ,  $\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$ ,  $\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$ ,  $\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2$ , and  $\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$ . The dynamic range of the UQI is  $[-1, 1]$ . The best value 1 is achieved if and only if for all  $i = 1, 2, \dots, N$ ,  $x_i = y_i$ .

**Visual information fidelity (VIF):** Visual information fidelity (VIF) is a full reference image quality assessment index based on natural scene statistics and the human visual system (HVS). The HVS is used to determine the accuracy of visual information, which includes factors such as the sharpness of edges, the accuracy of color representation, and the ability to detect subtle changes in contrast. VIF measures image fidelity by comparing the information recovered from a real image  $x$  with the information lost in a synthesized image  $y$  using the HVS. It is a straightforward ratio of the real and the generated images with a value between 0 and 1 and is defined as follows:

$$VIF = \frac{HVS(y)}{HVS(x)} \quad (18)$$

#### 4.1.2. Dataset

A dataset of O-shape epoxy drop substrate images was provided to us by Texas Instruments. In the dataset, there were 8850 good epoxy drop substrate images and only 16 rejected epoxy drop substrate images. As explained earlier, this is because defective patterns of epoxy drops rarely occur during production. To ready the dataset for processing, we cropped the region of interest (ROI), having a size of  $128 \times 128$ , from the images, as illustrated in Figure 5 (Figure 5a shows ROI cropping of non-defective or good epoxy drop substrate and Figure 5b shows ROI cropping of defective or rejected epoxy drop substrate). We selected 88 good epoxy drop substrate images with different lighting conditions/backgrounds and paired them with rejected epoxy drop substrate images. Some sample non-defective or good epoxy drop substrate images are shown in Figure 6. Additional defective or rejected epoxy drop substrate images were generated by rotation

and vertical/horizontal flips for the experiments. Figure 7 shows the 16 rejected real epoxy drop substrate images.

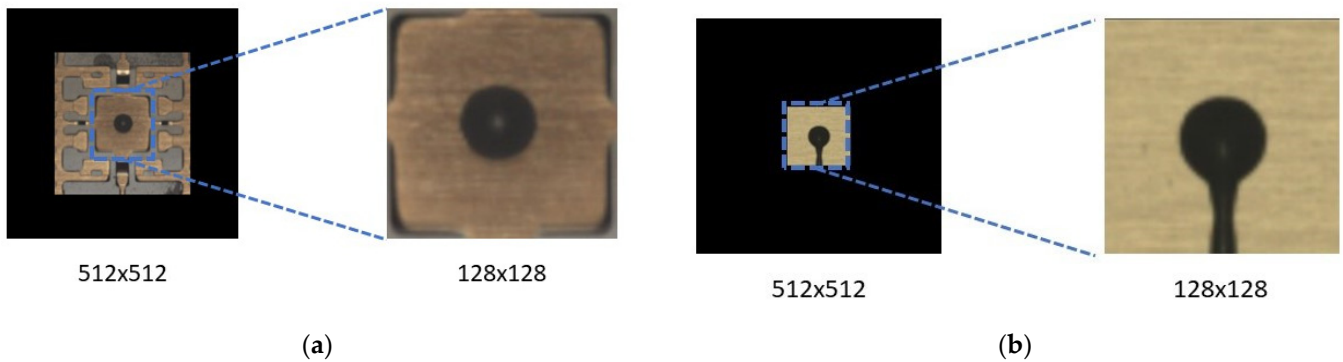


Figure 5. ROI for image generation.



Figure 6. Samples images of non-defective or good epoxy drop substrate.

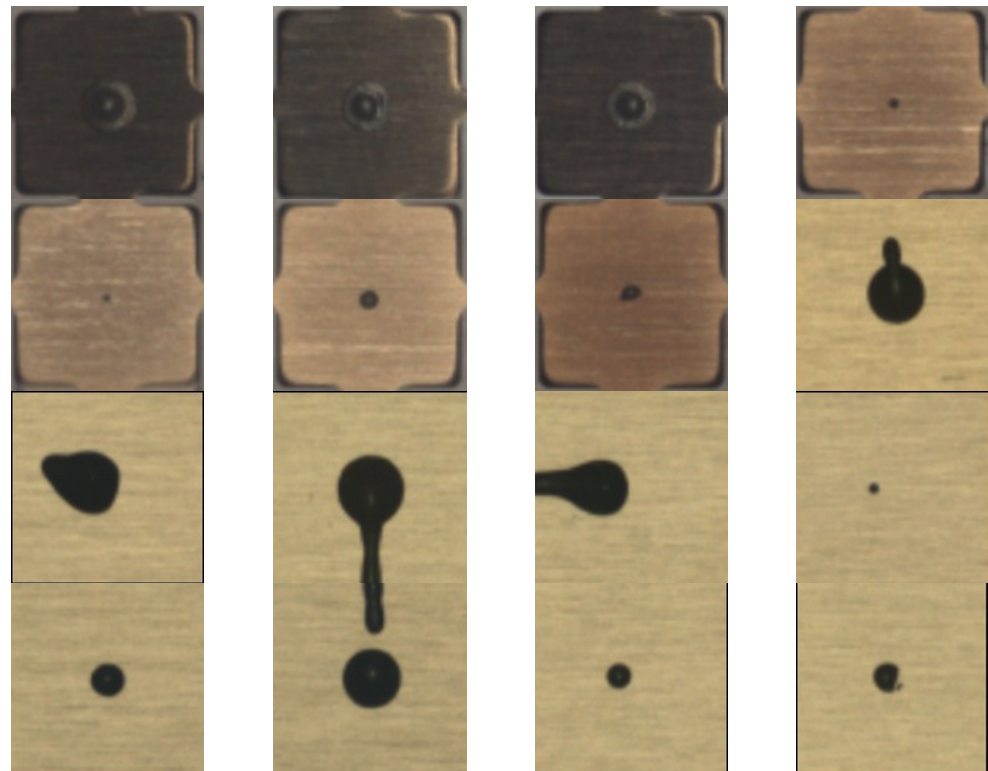


Figure 7. Defective or rejected images of epoxy drop substrate.

#### 4.1.3. Results and Discussion

We started our experiments by training the model using the standard cycle consistency loss (i.e.,  $L_1$ ) along with the other loss functions (i.e.,  $L_2$ , SSIM and LPIPS) separately as well as in combination. Then, we generated realistic synthesized images after training the model.

Figure 8 shows some sample outcomes of the generated synthesized rejected epoxy drop substrate images using different loss functions as the cycle consistency loss, separately and in combination with the CycleGAN standard loss function. Table 1 shows the evaluation metrics for different loss functions as the cycle consistency loss ( $\mathcal{L}_{Cycle}$ ). As the dataset utilized was made up of paired image data, all the generated synthesized rejected images (i.e.,  $I_r^s$ ) exhibited a relatively uniform reference material or real rejected epoxy drop substrate images (i.e.,  $I_r$ ). This table shows the averages and standard deviations of the metrics for the rejected images  $I_r$  and their generated counterpart, i.e., the generated synthesized images  $I_r^s$  translated from the good epoxy drop substrate images  $I_g$  using the CycleGAN network with different loss functions as the cycle consistency loss. The number of generated images for each loss function separately and in combination was 1408.

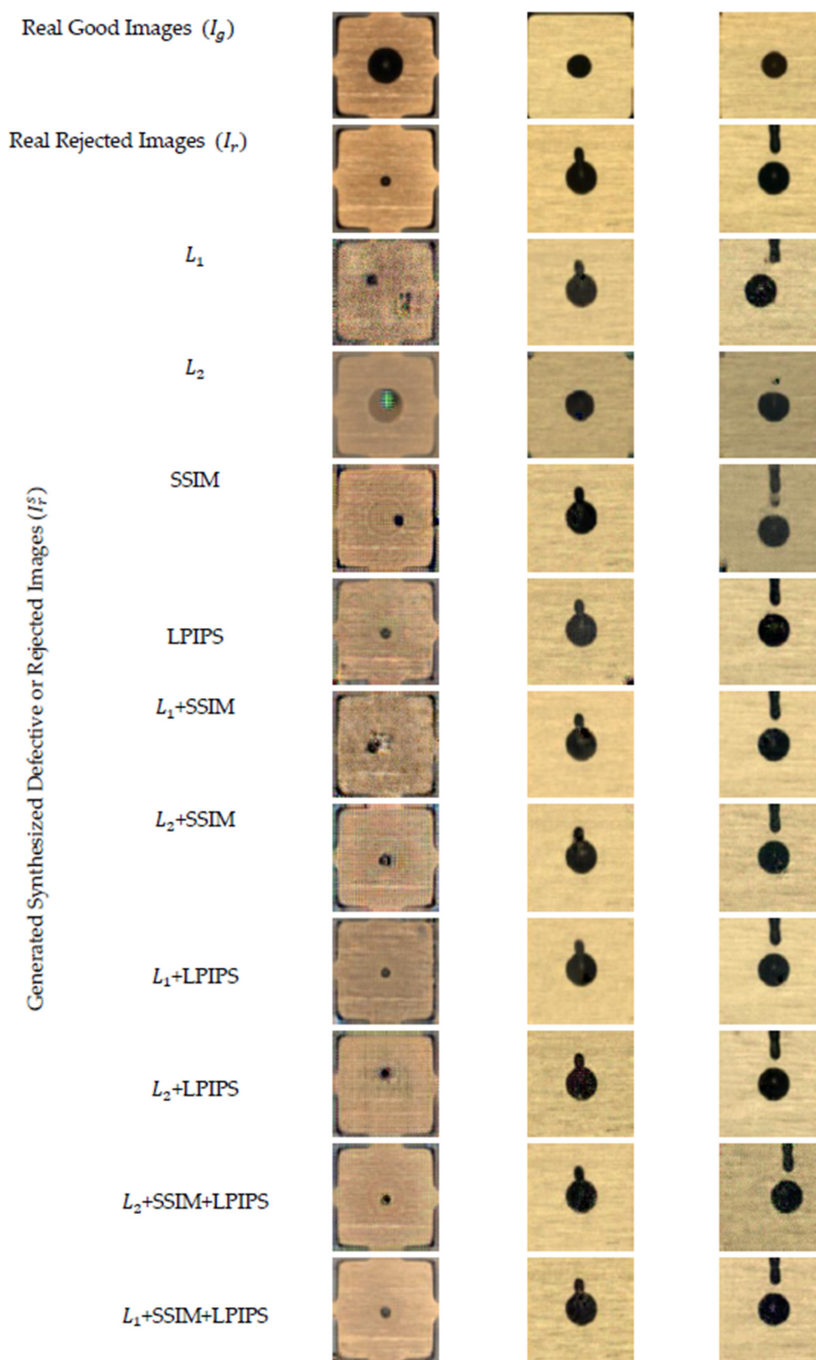


Figure 8. Sample images of generated rejected epoxy drop substrate using different loss functions as the cycle consistency loss function.

**Table 1.** Image quality evaluation metrics for different loss functions as the cycle consistency loss.

Loss Function as $\mathcal{L}_{Cycle}$	PSNR	UQI	VFI
	Mean $\pm$ STD	Mean $\pm$ STD	Mean $\pm$ STD
$L_1$ (Standard Cycle GAN)	18.77 $\pm$ 4.88	0.86 $\pm$ 0.16	0.19 $\pm$ 0.08
$L_2$	16.65 $\pm$ 1.66	0.89 $\pm$ 0.09	0.08 $\pm$ 0.05
SSIM	21.06 $\pm$ 2.23	0.93 $\pm$ 0.05	0.20 $\pm$ 0.06
LPIPS	22.51 $\pm$ 4.23	0.94 $\pm$ 0.09	0.26 $\pm$ 0.02
$L_1$ + SSIM	22.94 $\pm$ 4.50	0.93 $\pm$ 0.07	0.24 $\pm$ 0.02
$L_2$ + SSIM	23.39 $\pm$ 4.46	0.93 $\pm$ 0.08	0.25 $\pm$ 0.02
$L_1$ + LPIPS	23.26 $\pm$ 3.48	0.93 $\pm$ 0.07	0.32 $\pm$ 0.03
$L_2$ + LPIPS	22.24 $\pm$ 4.98	0.90 $\pm$ 0.15	0.27 $\pm$ 0.02
$L_2$ + SSIM + LPIPS	20.81 $\pm$ 6.75	0.90 $\pm$ 0.10	0.17 $\pm$ 0.02
$L_1$ + SSIM + LPIPS	29.86 $\pm$ 5.11	0.98 $\pm$ 0.01	0.44 $\pm$ 0.03

From Table 1, one can see that the  $L_2$  loss received the lowest score for all the metrics when used alone and performed better when used in combination with LPIPS and the SSIM. The standard cycle consistency loss function  $L_1$  loss performed worse than both the SSIM and LPIPS. These findings correlate with the visual examination of the images shown in Figure 8. We also found that combining the SSIM and LPIPS with  $L_1$  separately improved their scores, but combining all three together (i.e.,  $L_1$  + SSIM + LPIPS) gave the best results and visually looked more realistic and similar to the real rejected images.

Furthermore, from the results of Table 1, one can see that  $L_1$  performed better than  $L_2$  and the output generated from the  $L_2$  function had the blurring effect (see Figure 8), as mentioned in [34]. Additionally, combining  $L_1$  and  $L_2$  with the other loss functions also improved the performance of the model, as noted in [32]. It can be seen that applying the SSIM for  $L_1$  and  $L_2$  increased the performance of the model. LPIPS as the loss function was found to work better independently as well as in combination since it enhanced the image quality and helped to generate more realistic images, as noted in [45].

## 4.2. Impact of Generated Images on Defect Identification

### 4.2.1. Identification Metrics

The image classifier ResNet18 [42] was used here as a typical classifier to show the impact of the generated images when performing defect identification. As normally done for classification problems, the confusion matrix, precision, recall, and accuracy of the classifier were found with and without using the generated images. Table 2 shows a depiction of the confusion matrix with precision, recall, and accuracy denoted by

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

$$Recall = \frac{TP}{TP + FN} \quad (20)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (21)$$

where  $TP$  (true positive) indicates when a rejected image is placed in the defective or rejected class,  $TN$  (true negative) indicates when a good or non-defective image is placed in the non-defective or good class,  $FP$  (false positive) indicates when a non-defective or good image is placed in the defective or rejected class, and  $FN$  (false negative) indicates when a defective or rejected image is placed in the non-defective or good class.

**Table 2.** Confusion Matrix.

		Correct Labels	
		Rejected	Good
Predicted Labels	Rejected	TP	FP
	Good	FN	TN

#### 4.2.2. Datasets

To ready the dataset for the classification experiments, we randomly selected 1400 real good epoxy drop substrate images and labeled them as non-defective or good. For the defective or rejected class, the 16 available real defective or rejected epoxy drop substrate images were used to generate 2800 synthesized defective or rejected epoxy drop substrate images (1400 images were generated by the standard loss function CycleGAN and 1400 images were generated by our enhanced loss function CycleGAN). Then, these datasets were divided into 60% training, 20% validation, and 20% testing subsets with no overlap among them.

#### 4.2.3. Identification Outcomes

While keeping the same non-defective or good class images the same, for the defective or rejected class, the classifier ResNet18 was trained in three different ways, as follows:

1. By using real rejected epoxy drop substrate images;
2. By using rejected epoxy drop substrate images and generated rejected epoxy drop substrate images based on the standard loss function CycleGAN;
3. By using rejected epoxy drop substrate images and generated rejected epoxy drop substrate images based on our enhanced loss function CycleGAN.

Then, the above trained models were tested via the same testing data subset whose rejected class consisted of a combination of real and generated rejected epoxy drop substrate images. Table 3 shows a comparison of the identification outcomes. As can be seen from this table, the addition of the synthesized images significantly improved the identification outcome. Furthermore, our enhanced loss function CycleGAN provided a higher identification outcome compared to the standard loss function CycleGAN.

**Table 3.** Confusion matrix and identification outcomes for ResNet18 as a typical image classifier.

Data Augmentation Method	Confusion Matrix			Identification Metrics			
	Predicted Labels	Correct Labels		Precision	Recall	Accuracy	
		Rejected	Good				
No augmentation (only real data)	Predicted Labels	Rejected	23	02	0.92	0.04	36%
		Good	540	278			
CycleGAN standard with loss function	Predicted Labels	Rejected	381	05	0.99	0.68	78%
		Good	182	275			
CycleGAN with enhanced loss function	Predicted Labels	Rejected	458	13	0.97	0.81	86%
		Good	105	267			

## 5. Conclusions

In this paper, the loss function of the generative adversarial network of CycleGAN was enhanced or modified to generate high-quality defective epoxy drop images for die attachment in IC manufacturing. Such images are needed for the purpose of training or testing vision-based deep neural network inspection systems. A CycleGAN network with different cycle consistency loss functions was designed to generate different sets of synthesized images. Based on three evaluation metrics, it has been shown that by incorporating the loss functions of learned perceptual image patch similarity (LPIPS) and the structural similarity index metric (SSIM) into the standard CycleGAN loss function, more realistic or higher-quality synthesized epoxy drop images are generated as compared



to using the CycleGAN standard loss function. Furthermore, it has been shown that our enhanced loss function CycleGAN as a data augmentation approach leads to improved identification outcomes when using a typical image classifier. The enhancement approach developed in this paper is general purpose in the sense that it can be applied to other data augmentation scenarios involving other types of images.

**Author Contributions:** Conceptualization, L.A. and N.K.; methodology, L.A. and N.K.; software, L.A. and N.K.; validation, L.A. and N.K.; formal analysis, L.A. and N.K.; investigation, L.A. and N.K.; resources, L.A. and N.K.; data curation, L.A. and N.K.; writing—original draft preparation, L.A. and N.K.; writing—review and editing, L.A. and N.K.; visualization, L.A. and N.K.; supervision, N.K.; project administration, N.K.; funding acquisition, N.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work presented in this paper was supported by a grant from Texas Instruments (project number 1416) to the University of Texas at Dallas.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available on request due to privacy restrictions. Contact the authors.

**Acknowledgments:** The authors would like to express their sincerest gratitude to Hyowon Choi and Hetul Sanghvi from Texas Instruments (TI) for their feedbacks related to the problem addressed in this paper.

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

## References

1. Li, Y.; Wong, C.P. Recent advances of conductive adhesives as a lead-free alternative in electronic packaging: Materials, processing, reliability, and applications. *Mater. Sci. Eng. R Rep.* **2006**, *51*, 1–35. [[CrossRef](#)]
2. Capili, M.D. Understanding die attach epoxy open time. *Int. Res. J. Adv. Eng. Sci.* **2019**, *4*, 11–14.
3. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [[CrossRef](#)]
4. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
5. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
6. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
7. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)]
8. Antoniou, A.; Storkey, A.; Edwards, H. Augmenting Image Classifiers Using Data Augmentation Generative Adversarial Networks. In Proceedings of the 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I., Eds.; Lecture Notes in Computer Science (LNTCS, Volume 11141). Springer: Cham, Greece, 2018. [[CrossRef](#)]
9. Chen, Y.; Yang, X.-H.; Wei, Z.; Heidari, A.A.; Zheng, N.; Li, Z.; Chen, H.; Hu, H.; Zhou, Q.; Guan, Q. Generative adversarial networks in medical image augmentation: A review. *Comput. Biol. Med.* **2022**, *144*, 105382. [[CrossRef](#)]
10. Sampath, V.; Maurtua, I.; Aguilar Martín, J.J.; Iriondo, A.; Lluvia, I.; Aizpurua, G. Intra-class image augmentation for defect detection using generative adversarial neural networks. *Sensors* **2023**, *23*, 1861. [[CrossRef](#)]
11. Wen, L.; Wang, Y.; Li, X. A New Cycle-consistent adversarial networks with attention mechanism for surface defect classification with small samples. *IEEE Trans. Ind. Inf.* **2022**, *18*, 8988–8998. [[CrossRef](#)]
12. Wang, R.; Hoppe, S.; Monari, E.; Huber, M.F. Defect Transfer GAN: Diverse Defect Synthesis for Data Augmentation. In Proceedings of the 33rd British Machine Vision Conference (BMVC 2022), London, UK, 21–24 November 2022. [[CrossRef](#)]
13. Yang, B.; Liu, Z.; Duan, G.; Tan, J. Mask2Defect: A prior knowledge-based data augmentation method for metal surface defect inspection. *IEEE Trans. Ind. Inf.* **2022**, *18*, 6743–6755. [[CrossRef](#)]
14. Niu, S.; Li, B.; Wang, X.; Peng, Y. Region- and Strength-Controllable GAN for defect generation and segmentation in industrial images. *IEEE Trans. Ind. Inf.* **2022**, *18*, 4531–4541. [[CrossRef](#)]

15. Hu, J.; Yan, P.; Su, Y.; Wu, D.; Zhou, H. A method for classification of surface defect on metal workpieces based on twin attention mechanism generative adversarial network. *IEEE Sens. J.* **2021**, *21*, 13430–13441. [[CrossRef](#)]
16. Zhang, H.; Pan, D.; Liu, J.; Jiang, Z. A novel MAS-GAN-based data synthesis method for object surface defect detection. *Neurocomputing* **2022**, *499*, 106–114. [[CrossRef](#)]
17. Zhang, G.; Cui, K.; Hung, T.-Y.; Lu, S. Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 2523–2533. [[CrossRef](#)]
18. Jain, S.; Seth, G.; Paruthi, A.; Soni, U.; Kumar, G. Synthetic data augmentation for surface defect detection and classification using deep learning. *J. Intell. Manuf.* **2022**, *33*, 1007–1020. [[CrossRef](#)]
19. Du, Z.; Gao, L.; Li, X. A new contrastive GAN with data augmentation for surface defect recognition under limited data. *IEEE Trans. Instrum Meas.* **2023**, *72*, 3502713. [[CrossRef](#)]
20. He, Y.; Song, K.; Dong, H.; Yan, Y. Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. *Opt. Lasers Eng.* **2019**, *122*, 294–302. [[CrossRef](#)]
21. Niu, S.; Li, B.; Wang, X.; Lin, H. Defect image sample generation with GAN for improving defect recognition. *IEEE Trans. Autom. Sci. Eng.* **2022**, *17*, 1611–1622. [[CrossRef](#)]
22. Zhai, W.; Zhu, J.; Cao, Y.; Wang, Z. A Generative Adversarial Network Based Framework for Unsupervised Visual Surface Inspection. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 1283–1287. [[CrossRef](#)]
23. Lai, Y.T.K.; Hu, J.S.; Tsai, Y.H.; Chiu, W.Y. Industrial Anomaly Detection and One-class Classification using Generative Adversarial Networks. In Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 9–12 July 2018; pp. 1444–1449. [[CrossRef](#)]
24. Lu, H.-P.; Su, C.-T. CNNs combined with a conditional GAN for mura defect classification in TFT-LCDs. *IEEE Trans. Semicond. Manuf.* **2021**, *34*, 25–33. [[CrossRef](#)]
25. Liu, J.; Zhang, B.G.; Li, L. Defect detection of fabrics With Generative Adversarial Network Based flaws modeling. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 3334–3338. [[CrossRef](#)]
26. Qian, W.; Xu, Y.; Li, H. A self-sparse generative adversarial network for autonomous early-stage design of architectural sketches. *Comput. Aided Civ. Inf.* **2022**, *37*, 612–628. [[CrossRef](#)]
27. Xu, Y.; Tian, Y.; Li, H. Unsupervised deep learning method for bridge condition assessment based on intra-and inter-class probabilistic correlations of quasi-static responses. *Struct. Health Monit.* **2023**, *22*, 600–620. [[CrossRef](#)]
28. Chen, K.; Cai, N.; Wu, Z.; Xia, H.; Zhou, S.; Wang, H. Multi-scale GAN with transformer for surface defect inspection of IC metal packages. *Expert Syst. Appl.* **2023**, *212*, 118788. [[CrossRef](#)]
29. Li, J.; Cai, N.; Mo, Z.; Zhou, G.; Wang, H. IC solder joint inspection via generator-adversarial-network based template. *Mach. Vis. Appl.* **2021**, *32*, 96. [[CrossRef](#)]
30. Chen, S.-H.; Kang, C.-H.; Perng, D.-B. Detecting and measuring defects in wafer die using GAN and YOLOv3. *Appl. Sci.* **2020**, *10*, 8725. [[CrossRef](#)]
31. Jeon, Y.; Kim, H.; Lee, H.; Jo, S.; Kim, J. GAN-based Defect Image Generation for Imbalanced Defect Classification of OLED panels. In Proceedings of the Eurographics Symposium on Rendering 2022, Prague, Czech Republic, 4–6 July 2022; Ghosh, A., Wei, L.-Y., Eds.; The Eurographics Association: Eindhoven, The Netherlands, 2022. [[CrossRef](#)]
32. Abu-Srhan, A.; Abushariah, M.A.M.; Al-Kadi, O.S. The effect of loss function on conditional generative adversarial networks. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6977–6988. [[CrossRef](#)]
33. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context Encoders: Feature Learning by Inpainting. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544. [[CrossRef](#)]
34. Isola, P.; Zhu, J.; Zhou, T.; Efros, A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976. [[CrossRef](#)]
35. Abobakr, A.; Hossny, M.; Nahavandi, S. SSIMLayer: Towards Robust Deep Representation Learning via Nonlinear Structural Similarity. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 1234–1238. [[CrossRef](#)]
36. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss functions for image restoration with neural networks. *IEEE Trans. Comput. Imaging* **2017**, *3*, 47–57. [[CrossRef](#)]
37. Shao, G.; Huang, M.; Gao, F.; Liu, T.; Li, L. DuCaGAN: Unified dual capsule generative adversarial network for unsupervised image-to-image translation. *IEEE Access* **2020**, *8*, 154691–154707. [[CrossRef](#)]
38. Niu, S.; Li, B.; Wang, X.; He, S.; Peng, Y. Defect attention template generation CycleGAN for weakly supervised surface defect segmentation. *Pattern Recognit.* **2022**, *123*, 108396. [[CrossRef](#)]
39. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
40. Yamins, D.L.; DiCarlo, J.J. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci.* **2016**, *19*, 356–365. [[CrossRef](#)]

41. Li, C.; Wand, M. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. In *Proceedings of the Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 9907. [[CrossRef](#)]
42. He, K.; Zhang, X.; Ren, S.; Sun, S. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
43. Wang, Z.; Bovik, A.C. A universal image quality index. *IEEE Signal Process. Lett.* **2002**, *9*, 81–84. [[CrossRef](#)]
44. Sheikh, H.R.; Bovik, A.C. Image information and visual quality. *IEEE Trans. Image Process.* **2006**, *15*, 430–444. [[CrossRef](#)]
45. Jo, Y.; Yang, S.; Kim, S.J. Investigating Loss Functions for Extreme Super-Resolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1705–1712. [[CrossRef](#)]

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