

Received September 3, 2020, accepted September 13, 2020, date of publication September 18, 2020, date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024554

# Weakly-Supervised Defect Segmentation on Periodic Textures Using CycleGAN

MINSU KIM<sup>1</sup>, HOON JO<sup>2</sup>, MOONSOO RA<sup>3</sup>, (Member, IEEE),  
AND WHOI-YUL KIM<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea

<sup>2</sup>Samsung Electronics Company Ltd., Hwaseong 18448, South Korea

<sup>3</sup>LightVision Inc., Seoul 04793, South Korea

Corresponding authors: Moonsoo Ra (ravimoon@gmail.com) and Whoi-Yul Kim (wykim@hanyang.ac.kr)

This work was supported in part by the Samsung Electronics' University Research and Development Program, in part by the Korea Evaluation Institute of Industrial Technology (KEIT) Grant funded by the Korean Government (MOTIE), the Road Surface Condition Detection Using Environmental and In-Vehicle Sensors, under Grant 20000293, and in part by the Human Resources Program in Energy Technology Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry, and Energy (MOTIE), South Korea, under Grant 20204030200100.

**ABSTRACT** The importance of an automated defect inspection system has been increasing in the manufacturing industries. Various products to be examined have periodic textures. Among image-based inspection systems, it is common that supervised defect segmentation requires a great number of defect images with their own region-level labels; however, it is difficult to prepare sufficient training data. Because most products are of normal quality, it is difficult to obtain images of product defects. Pixel-wise annotation for semantic segmentation tasks is an exhausting and time-consuming process. To solve these problems, we propose a weakly-supervised defect segmentation framework for defect images with periodic textures and a data augmentation process using generative adversarial networks. With only image-level labeling, the proposed segmentation framework translates a defect image into its defect-free version, called a golden template, using CycleGAN and then segments the defects by comparing the two images. The proposed augmentation process creates whole new synthetic defect images from real defect images to obtain sufficient data. Furthermore, synthetic non-defect images are generated even from real defect images through the augmentation process. The experimental results demonstrate that the proposed framework with data augmentation outperforms an existing weakly-supervised method and shows remarkable results comparable to those of supervised segmentation methods.

**INDEX TERMS** Automated defect inspection, visual inspection system, weakly-supervised learning, periodic textures, data augmentation, generative adversarial networks.

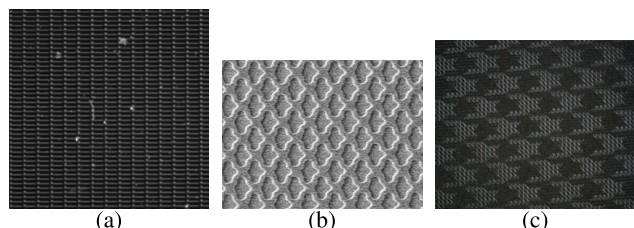
## I. INTRODUCTION

Most manufacturing industries have aimed to provide their clientele with defect-free products to enhance their corporate competitiveness. In order to achieve this goal, product inspections are usually conducted at the final stage of the manufacturing process, and a large number of manufactured products have been examined by human inspectors. It is rare for the accuracy of the inspector to be uniform during long working hours because inspectors become tired as time elapses. Furthermore, because it is difficult for novices to adequately check the product quality initially, it is likely that

The associate editor coordinating the review of this manuscript and approving it for publication was Li He <sup>1b</sup>.

the inspection results will be unsatisfactory, and time will be required for training purposes. For these reasons, the demand for automated defect inspection has been increasing in the manufacturing industries. Especially, defect inspection is quite important in the semiconductor manufacturing process.

To satisfy the need, several automated inspection systems have been proposed. Image-based, thermography-based, and ultrasonic-based inspection systems, which are utilized for specific purposes, have been widely applied. Image-based systems perform inspection applying image processing and computer vision techniques on images. These systems focus on the appearance of the area where a defect exists on visible objects. Thermography-based systems examine defects by analyzing the thermal distribution of objects [1]–[4]. It is



**FIGURE 1.** Examples of periodic texture image for inspection: (a) TFT-LCD [8], (b) wafer [25], and (c) fabric [26].

applied to objects whose defects are derived from certain thermal characteristics. Ultrasound-based systems that transmit ultrasonic waves into objects to detect flaws are mainly used when an inspection for the internal structures of the objects is needed [5]–[7].

An image-based inspection system is one of the most utilized approaches and has been widely applied to the inspection of products such as textiles, wafers, and thin film transistor liquid crystal displays (TFT-LCDs). As shown in Figure 1, such products have periodic patterns, respectively. In order to detect defects in periodic patterns, various methods utilizing image processing techniques have been introduced: template-based, filter-based, and statistical methods [8]. Template-based methods are simpler than other approaches, and are used to compare an input image with its defect-free shape, called a golden template [9]–[12]. This approach is only useful when a golden template for the input image can be obtained. Filter-based methods perform convolution with filter banks and detect a defective region by analyzing the response of the results [13]–[15]. In this approach, knowing the structure of both defective and defect-free regions helps to design appropriate filters. Statistical methods inspect defects based on the statistical difference between the defective and defect-free regions [16]–[18]. In order to discriminate between the two regions, an adequate number of sample images are needed. In recent years, convolutional neural networks (CNNs) have shown remarkable outcomes in various computer vision applications. In particular, several networks including FCN [19], SegNet [20], and AdapNet [21] demonstrated a notable performance for semantic segmentation, and thus, deep learning-based approaches have been widely applied to defect inspections [22]–[24].

Although the field of image-based defect inspection has advanced, some challenging issues still remain: data insufficiency, data imbalance, and annotation cost, to name a few. The first issue is data insufficiency. Several datasets for general object detection and semantic segmentation have been publicly released such as PASCAL VOC [27], COCO [28], and Cityscapes [29]. Unlike general objects, it is difficult to obtain images that contain defects because most products are not faulty. It is common that only three or four occurrences per million units or events are allowed in the modern manufacturing industries employing Six Sigma methodology. Especially, it is more difficult to capture defects in semiconductors than other products since defects in semi-

conductors are several nanometers in size. Furthermore, generating defective products for the purpose of inspection is even more difficult. In addition, in most cases, the inspection results are treated as strictly in-house and confidential. The second issue is data imbalance, which indicates that data of certain classes are lack or missed. In general cases of defect segmentation, imbalanced data are concerned with the kinds of defects; however, this is occasionally broadened to the existence of defects. Because inspection tasks mainly focus on defective products, it is possible that data of normal products will not exist. The last but not the least one is annotation cost. In the field of machine learning, model training often requires labels of data called the ground truth (GT), which are the correct answers to the data (e.g., class, bounding box, and segmentation mask) on the object of interest. In detail, the annotation required for a semantic segmentation task is tedious and time-consuming owing to the pixel-wise labeling.

In this paper, we propose an image-based defect segmentation framework for periodic texture images using GAN-based golden template generation and data augmentation process. The proposed framework generates the golden template of the input image and then segments defects in a pixel-wise manner using simple post-processing. The concept of this framework is inspired by the dissimilarities between the defective and normal regions. Moreover, synthetic defect and non-defect images are generated from a small number of real defect images through the proposed data augmentation process, the volume of which is sufficient to train a model. The process does not apply simple geometric transformations to existing images, such as scaling, translation, and rotation, but generates whole new images. The main contributions of this paper are:

- We propose a framework using CycleGAN [30] for defect segmentation on periodic textures. We achieve a competitive performance of pixel-wise segmentation compared to supervised learning-based methods while using only class labels. The proposed framework lowers the burden of the annotation cost.
- We propose two data augmentation processes for generating synthetic defect and non-defect images, respectively. The proposed data augmentation process alleviates the data insufficiency and data imbalance problems described earlier.

The rest of this paper is organized as follows. Section II introduces existing image-based defect segmentation methods. Section III describes the proposed defect segmentation framework and data augmentation process. Section IV presents our experimental settings and results. Finally, Section V provides some concluding remarks.

## II. RELATED WORKS

### A. TRADITIONAL APPROACHES

As mentioned in the introduction section, various methods based on traditional image processing and computer

vision techniques have been reported. Template-based methods utilize defect-free template images for comparison. Khalaj *et al.* [9] constructed the building block, the structure of repeated patterns, from direct patterned wafers. The repetition periods of the patterns were estimated along the horizontal and vertical directions in the input image. Xie and Guan [10] generated defect-free images using a simulated building block whose size is equal to the horizontal and vertical periods of repeating patterns for patterned wafer inspection. A golden template was built in a manner analogous to that of constructing a building block. Shankar and Zhong [11] introduced a template-based vision system to inspect semiconductor wafer surfaces. In this system, the mean square error (MSE) analysis between the reference circuit image and the test image was performed using a two-dimensional discrete cosine transform (DCT). A rule-based approach for semiconductor defect segmentation was reported in which the segmentation was performed based on the diagnostic rule, similarity rule, and logical rule using an error image [12]. The resulting error image is generated by matching the inspected die image and the golden master (GM) image.

Filter-based methods are based on various types of filters that output different responses in each of the defective and defect-free regions. Tsa and Wu [13] chose the best parameters of a Gabor filter based on the output response of convolution for each textured surface to deal with unseen defects in the given surface. They demonstrated the effectiveness of their method for both structural and statistical textures. Gabor filter-based supervised and unsupervised approaches for defect detection in textured materials were presented [15]. In the supervised scheme, the best representative Gabor filter was determined based on the filter-selection methodology. For unsupervised defect detection, a multi-channel filtering scheme was used and an imaginary Gabor function (IGF) was employed to lower the computational time. Chan and Pang [14] analyzed the frequency spectrum to detect defects in the patterned fabric on the basis that the frequency spectrum will vary when the fabric image has defects. They utilized a fast Fourier transform (FFT) instead of a discrete Fourier transform (DFT) for computational efficiency.

In the statistical methods, defects can be discriminated against flawless regions based on the characteristics of the texture, such as the intensity distribution. Liu *et al.* [16] introduced an inline defect-defection (IDD) system for TFT-LCD inspection. After some pre-processing to obtain patches for classification, they classified whether a patch is defective or not using locally linear embedding (LLE) and support vector data description (SVDD). In order to solve the problems in SVDD, automatic target defect identification based on fuzzy support vector data description (F-SVDD) ensemble was reported [17]. A partitioning-entropy-based kernel fuzzy *c*-means (KFCM) algorithm was utilized for constructing F-SVDD ensemble. Yu and Lu [18] developed a wafer map defect detection and recognition method using local and

nonlocal preserving projection (LNPP) and joint local and nonlocal linear discriminant analysis (JLNDA). They used several features for representing wafer maps: geometrical features, gray features, textual features, and projection features.

In particular, there were some studies where local binary patterns (LBP) variants were used to detect defective regions. Tajeripour and Fekri-Ershad [31] developed an approach for porosity detection in stone textures using one-dimensional local binary patterns (1DLBP). They divided a stone image into sub-windows and compared a feature vector of each window with that of the porosity-less image. A surface defect detection approach using noise-resistant color local binary patterns (NrCLBP) was presented [32]. They combined feature vectors that have different sizes of neighborhood radius in NrCLBP for multi-resolution analysis. Cao *et al.* [33] introduced a nickel foam surface defect detection method using multi-scale block local binary patterns (MB-LBP). They utilized a non-subsampled contourlet transform (NSCT) to extract multi-scale texture characteristics.

## B. METHODS IN THE ERA OF DEEP NEURAL NETWORKS

After the success of AlexNet [34], several defect inspection methods using deep neural networks (DNNs) have been recently reported. According to the level of supervision for the training data, DNN-based methods can be categorized into three groups: supervised, weakly-supervised, and unsupervised learning.

Ouyang *et al.* [22] constructed a network called PPAL-CNN, which consists of seven layers for fabric defect detection. In order to localize fine defects and deal with data imbalance, they generated a defect probability map from an input image and utilized it as a dynamic activation layer (PPAL) instead of an activation function. Marino *et al.* [35] applied class activation mapping (CAM) [36] to potato defect classification and localization. CAM gives a network trained for classification tasks the ability to localize target objects in images by adding a convolutional layer and a global average pooling layer before the last fully-connected layer [36]. They employed several well-known networks such as AlexNet [34], VGGNet [37], and GoogLeNet [38] as backbone networks and modified these networks to extract the CAM results. A network called LEDNet, which is based on CAM for classifying and localizing defects in LED chip images, was presented [39]. In addition, data augmentation was performed randomly for the collected images using geometric transformation techniques including rotation, flipping, translating, noising, and blurring to improve the accuracy. Schlegl *et al.* [40] introduced AnoGAN to detect lesions in medical images with deep convolutional generative adversarial networks (DCGAN) [41]. They trained a model with only normal data, and thus, the trained network can represent the distribution of normal data. They then compared the input images and the images generated from the latent variable computed by the inverse operation of the generator. When an input image is normal, it is analogous to its generated image. By contrast, a visual

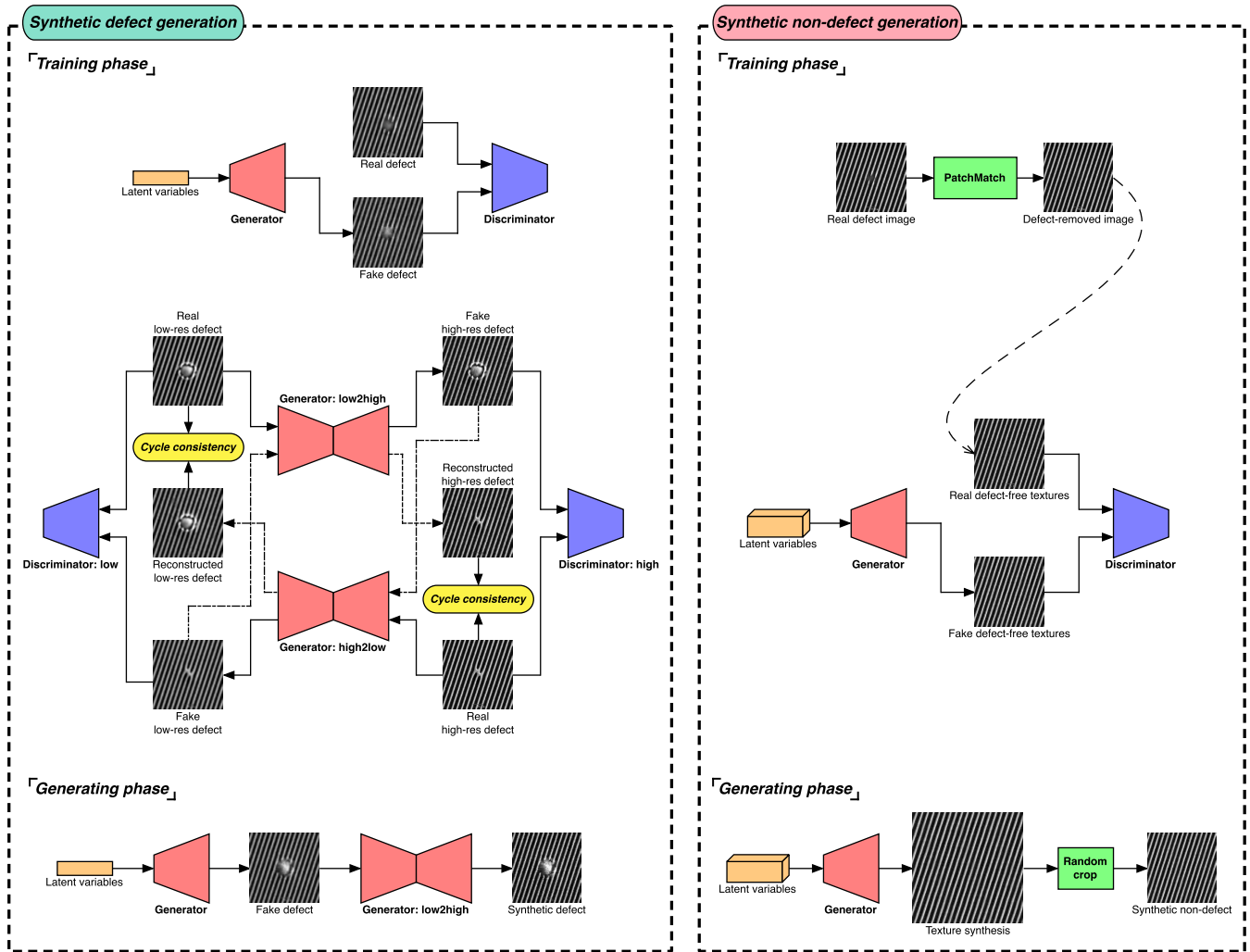


FIGURE 2. Flowchart of the proposed data augmentation process for periodic texture images.

difference exists between the input image and the generated image when the input image is anomalous.

Niu *et al.* [42] presented DefectGAN using CycleGAN [30] for weakly-supervised defect detection. This study is similar to our proposed framework in terms of the non-defect image generation using CycleGAN; however, the total loss function used in the study was inadequate to make a golden template well for periodic patterns. In order to deal with this problem, we utilized another term, *identity mapping loss* [30], to the total loss function and demonstrated the effectiveness of the additional term. Moreover, we performed data augmentation to solve the problems of data insufficiency, and generalization performance of the golden template generation is enhanced through our data augmentation scheme.

### C. DATA AUGMENTATION FOR IMPROVING PERFORMANCE

The volume and diversity of data are crucial to data-driven approaches such as DNN-based methods. Various data augmentation approaches for improving performance in different tasks have been presented.

Budvytis *et al.* [43] investigated the effect of video data augmentation for semantic segmentation in driving environments. They increased the segmentation performance of different networks by performing label propagation from coarsely labeled frames to adjacent unlabeled ones. The effectiveness of data augmentation in image classification tasks was demonstrated [44]. Three augmentation approaches including traditional transformations, GANs, and the augmentation network were utilized in this study. Bowles *et al.* [45] improved segmentation accuracy in medical imaging with augmenting training data. They investigated the performance of segmentation networks trained with different amounts of synthetic data. GAN-based medical image augmentation was performed for liver lesion classification [46]. In this study, two GAN variants were employed to generate synthetic liver lesion images, and the classification performance was improved by using the generated synthetic images. Zhao *et al.* [47] synthesized labeled medical images for the segmentation task in magnetic resonance imaging (MRI) brain scans. They trained spatial and appearance transform models for generating synthetic images and labels.

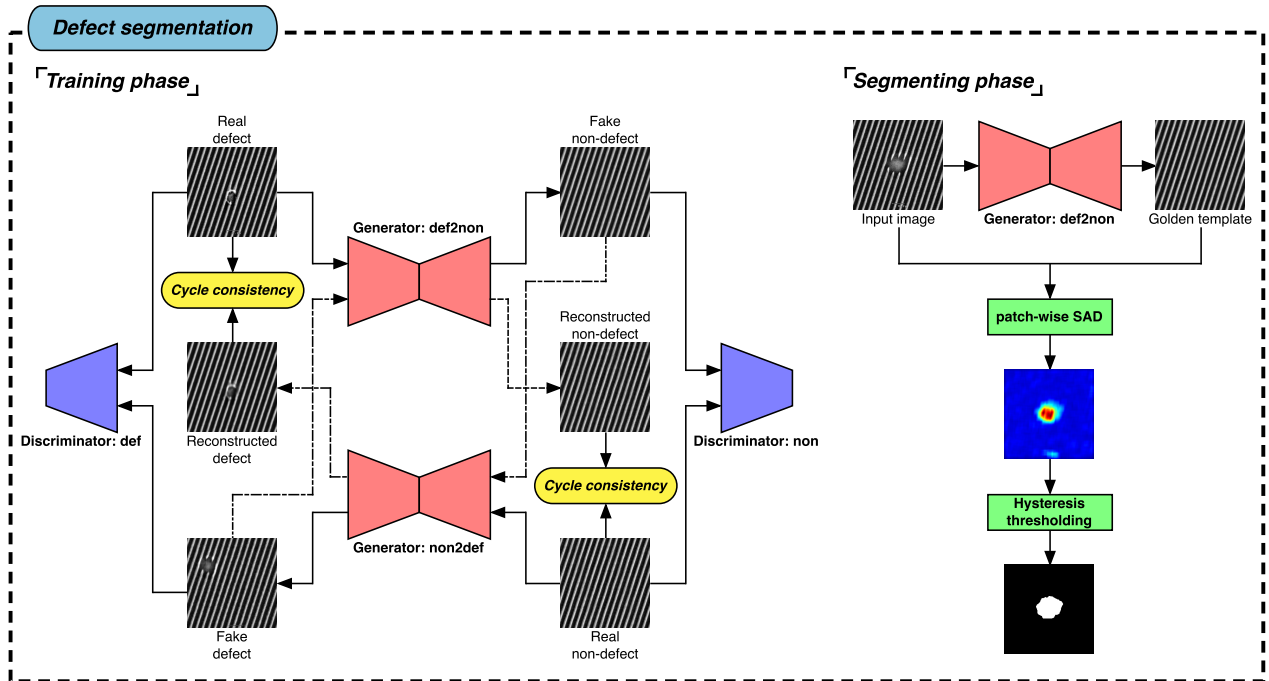


FIGURE 3. Flowchart of the proposed framework for defect segmentation in periodic texture images.

### III. PROPOSED FRAMEWORK

#### A. OVERVIEW

Before describing the details of the defect segmentation in periodic texture images, we first describe the overall scheme. Flowcharts of the proposed data augmentation process and the defect segmentation framework are depicted in Figures 2 and 3, respectively.

As shown in Figure 2, the proposed data augmentation process includes the two subordinate procedures for synthetic image generation: defect and non-defect. By using DCGAN [41] and CycleGAN [30], we create synthetic defect images whose volume is sufficient for training a network. PatchMatch [48] and periodic spatial generative adversarial network (PSGAN) [49] are utilized to create synthetic non-defect images. The proposed data augmentation allows the golden template generator to produce more plausible results. This suggests that our data augmentation scheme improves the generalization performance of the golden template generation.

The proposed defect segmentation framework consists of the golden template generation and post-processing, as illustrated in Figure 3. The golden template generation is performed using CycleGAN, which makes a defect-free version of the input image. After the golden template, straightforward image processing techniques are applied for detecting defects. To make a golden template for periodic patterns, we employ another loss term, *identity mapping loss* [30]. Although the loss is often auxiliary in other CycleGAN applications, in this work at least, it is crucial to the golden template generation from the perspective that the periodicity of the pattern must remain.

#### B. SYNTHETIC DEFECT IMAGE GENERATION

Synthetic defect images are generated out of real defect images through DCGAN [41] and CycleGAN [30]. DCGAN is utilized to make synthetic defect images; however, whose resolution is too small to be used as training samples for golden template generation. In addition, it is inadequate to apply a naive scaling method to the images. To solve this problem, image-to-image translation using CycleGAN is carried out for super-resolution.

In the training phase, each of the two networks is trained for their own purposes. In the generating phase, synthetic defect images are created through the trained networks.

DCGAN consists of two adversarial modules, a generator  $G$  and a discriminator  $D$ , which are trained by min-max game with the loss function  $V(G, D)$ :

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

where  $x$  is the input data for a discriminator  $D$ , and  $z$  is a latent variable for a generator  $G$ . In this work, a generator that creates fake defect images indistinguishable from real ones is learned when given the actual defect images as training samples. From now on this network will be called as  $\mathcal{N}_{FD}$ .

CycleGAN for super-resolution learns the two mapping functions  $G_H : X_L \rightarrow X_H$ ,  $G_L : X_H \rightarrow X_L$  where  $X_L$  is the low-resolution domain of the defect and  $X_H$  is the high-resolution domain of the defect. The two adversarial discriminators,  $D_{X_H}$  and  $D_{X_L}$ , aim to distinguish the data of the domain and the data mapped from the other domain, respectively. Loss functions for the total loss are expressed as:

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)} [\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)} [\log (1 - D_Y(G(x)))], \quad (2)$$

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1], \quad (3)$$

$$\mathcal{L}_{idt}(G, F) = \mathbb{E}_{y \sim p_{data}(y)} [\|G(y) - y\|_1] + \mathbb{E}_{x \sim p_{data}(x)} [\|F(x) - x\|_1]. \quad (4)$$

Our total loss function used to train a model for super-resolution is:

$$\begin{aligned} & \mathcal{L}(G_H, G_L, D_{X_L}, D_{X_H}) \\ &= \mathcal{L}_{GAN}(G_H, D_{X_H}, X_L, X_H) \\ &+ \mathcal{L}_{GAN}(G_L, D_{X_L}, X_H, X_L) \\ &+ \lambda_{cyc} \mathcal{L}_{cyc}(G_H, G_L) \\ &+ \lambda_{idt} \mathcal{L}_{idt}(G_H, G_L), \end{aligned} \quad (5)$$

where  $\mathcal{L}_{GAN}$ ,  $\mathcal{L}_{cyc}$ , and  $\mathcal{L}_{idt}$  are the *adversarial loss*, *cycle consistency loss*, and *identity mapping loss*, respectively.  $\lambda_{cyc}$  and  $\lambda_{idt}$  are used to control the impact of  $\mathcal{L}_{cyc}$  and  $\mathcal{L}_{idt}$ , respectively. The network for super-resolution will be termed  $\mathcal{N}_{SR}$  hereafter.

After the two networks are trained, the images produced from  $\mathcal{N}_{FD}$  trained to create fake defect images are fed into  $\mathcal{N}_{SR}$  trained for super-resolution. We use the resulting images of  $\mathcal{N}_{SR}$  as our synthetic defect images for training the golden template generator.

### C. SYNTHETIC NON-DEFECT IMAGE GENERATION

To train CycleGAN for golden template generation, the two domains of the image are required. For this reason, PatchMatch [48] and PSGAN [49] are employed to perform synthetic non-defect image generation. In order to deal with a situation in which real non-defect images can not be acquired, factitious images such as defect-removed are generated using PatchMatch. Periodic textures are then synthesized from the defect-removed images using PSGAN.

In PatchMatch, a nearest-neighbor field (NNF) is initialized as patches, which are at the uniformly random offset  $f(x, y)$  across the whole image. Based on the patch distance  $D(\mathbf{v})$  between the patch at  $(x, y)$  in an image and the patch at  $(x, y) + \mathbf{v}$  in the other image, the offset  $f(x, y)$  is propagated. On odd iterations,  $f(x, y)$  is changed into a value that minimizes  $\{D(f(x, y)), D(f(x - 1, y)), D(f(x, y - 1))\}$ . Furthermore, this propagation is performed in the opposite direction using  $f(x + 1, y)$  and  $f(x, y + 1)$  on even iterations.

After the propagation, the offset  $\mathbf{v}_0 = f(x, y)$  is checked with different candidate offsets to avoid convergence to the local minima. A series of candidate offsets exponentially decrease as:

$$\mathbf{u}_i = \mathbf{v}_0 + w\alpha^i \mathbf{R}_i, \quad (6)$$

where  $w$  is the maximum search radius, which is initially set to the maximum image dimension,  $\alpha$  is the decaying parameter for reducing the search window sizes, and  $\mathbf{R}_i$  is

a random value in  $[-1, 1] \times [-1, 1]$ . This random search finishes when  $w\alpha^i$  is less than 1 pixel.

PSGAN is based on the DCGAN architecture; however, it is composed solely of convolutional layers. In addition, the generator  $G$  is extended in a two-dimensional spatial domain to map a latent variable  $Z \in \mathbb{R}^{L \times M \times d}$  to an image  $X \in \mathbb{R}^{H \times W \times C}$ . A latent variable  $Z$  consists of three sections: local independent part  $Z^l$ , spatially global part  $Z^g$ , and periodic part  $Z^p$ . The channel dimension of  $Z$ ,  $d$ , is the sum of the channel dimensions of the three parts  $\{d^l, d^g, d^p\}$ . In accordance with the extension of the generator  $G$ , the discriminator  $D$  outputs a  $L \times M$  field from an image  $X$ .

As the variation of the generator and discriminator in PSGAN, the standard GAN loss function is also altered as:

$$\begin{aligned} \mathcal{V}(D, G) &= \frac{1}{LM} \sum_{i=1}^L \sum_{j=1}^M \mathbb{E}_{X' \sim p_{data}(X')} [\log D_{ij}(X')] \\ &+ \frac{1}{LM} \sum_{i=1}^L \sum_{j=1}^M \mathbb{E}_{Z \sim p_Z(Z)} [\log (1 - D_{ij}(G(Z)))], \end{aligned} \quad (7)$$

where  $D_{ij}(X')$  is the discriminator at  $(i, j)$ ,  $1 \leq i \leq L$  and  $1 \leq j \leq M$ , for a local part  $X'$  in an input image  $X$ .

After a network for texture synthesis is trained, the enlarged synthesis results are randomly cropped. The resulting images of the random cropping are used as our synthetic non-defect images for the training of a golden template generator. Henceforth, the network for texture synthesis will be dubbed  $\mathcal{N}_{TS}$ .

### D. GOLDEN TEMPLATE GENERATION

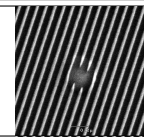
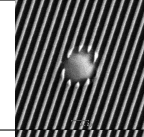
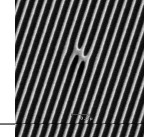
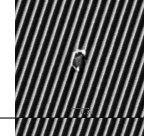
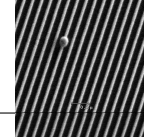
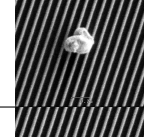
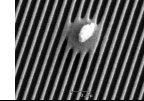
For golden template generation, CycleGAN learns the two mapping functions  $G_N : X \rightarrow Y$  and  $G_D : Y \rightarrow X$  between the defect domain  $X$  and the non-defect domain  $Y$ . In order to train a network for golden template generation, we used the same total loss function in Equation (5). The total loss function for a golden template generator is expressed as:

$$\begin{aligned} \mathcal{L}(G_N, G_D, D_X, D_Y) &= \mathcal{L}_{GAN}(G_N, D_Y, X, Y) \\ &+ \mathcal{L}_{GAN}(G_D, D_X, Y, X) \\ &+ \lambda_{cyc} \mathcal{L}_{cyc}(G_N, G_D) \\ &+ \lambda_{idt} \mathcal{L}_{idt}(G_N, G_D), \end{aligned} \quad (8)$$

where  $\mathcal{L}_{GAN}$ ,  $\mathcal{L}_{cyc}$ , and  $\mathcal{L}_{idt}$  are calculated using Equations (2) to (4). After this section we will call the network for golden template generation as  $\mathcal{N}_{GT}$ .

We employ the *identity mapping loss* [30] which allows a model to have the capability to preserve the color composition after translation as the additional term in the total loss function. Although the loss term is not commonly used in other applications, we found that it helps to preserve the periodicity of patterns. When the coefficient of the identity mapping loss is zero, flawless regions are slightly varied. This phenomenon occurs when  $\mathcal{N}_{SR}$  is trained without  $\mathcal{L}_{idt}$  in Equation (5). Only the defective region should be changed while the defect-free

**TABLE 1.** Examples of defect images and categorization of the type of defect in our semiconductor dataset.

Examples	Descriptions
	Some line patterns are broken.
	Some line patterns are broken, and there is a bulge in the broken patterns.
	Some line patterns are bridged.
	Some line patterns are broken and bridged with their nearby patterns.
	There is a bead-like foreign body on the line patterns.
	There is an amorphous foreign body on the line patterns.
	Some line patterns are bridged, and there is an amorphous foreign body on the bridged surface.

region is unaltered, which is our goal and the reason for using the *identity mapping loss*.

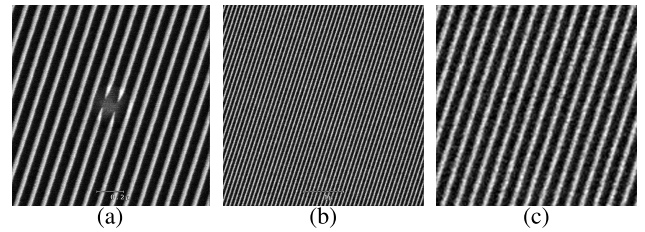
**E. DEFECT SEGMENTATION**

After the golden template of the input image is obtained, simple image processing techniques are applied for detecting defects. To measure the similarity between the input image and its golden template, the patch-wise sum of the absolute difference (pSAD) is calculated as:

$$pSAD(i, j) = \sum_{(k, l) \in \Omega(i, j)} |I_D(k, l) - I_G(k, l)|, \quad (9)$$

where  $\Omega(i, j)$  is a patch ranging from  $(i - W, j - H)$  to  $(i + W, j + H)$ .  $I_D$  and  $I_G$  denote the input defect image and the golden template of the input image created by the golden template generator, respectively.

In general, defective regions are quite different from their golden templates. On the contrary, defect-free regions are extremely similar to the templates. Therefore, the values in the pSAD results are usually larger in the defective region



**FIGURE 4.** Modification of non-defect images to construct *real* subset: (a) real defect image, (b) real non-defect image captured at different scales, and (c) the resulting non-defect patch.

**TABLE 2.** The three subsets of our semiconductor dataset for golden template generation.

Subset	Domain	Defect images		Non-defect images	
		real	synthetic	real	synthetic
<i>real</i>		200	0	200	0
<i>real+syn</i>		200	0	0	200
<i>syn</i>		0	10,000	0	10,000

than in the flawless region. Defects are then segmented by applying hysteresis thresholding to the pSAD results.

**IV. EXPERIMENTS**

**A. IMPLEMENTATION DETAILS**

We utilized several networks to verify our framework. In our experiments, all the network training and testing were performed on an AMD Ryzen 7 2700X CPU and an NVIDIA RTX 2080Ti GPU using CUDA 10.0.

**Synthetic Defect Image Generation** Firstly, we trained  $\mathcal{N}_{FD}$  on defect images of  $64 \times 64$  resolution and employed the architecture introduced by Radford *et al.* [41]. The network was trained for 2500 epochs and the dimension of the latent variable was set to 100. Secondly, we trained  $\mathcal{N}_{SR}$  on low-resolution defect images and high-resolution defect images of  $256 \times 256$  resolution with the architecture presented by Zhu *et al.* [30]. The  $64 \times 64$  low-resolution images were resized to  $256 \times 256$  for training. We trained the network for 200 epochs keeping the learning rate for the first 100 epochs and linearly decaying the initial rate to zero in the next 100 epochs. The coefficients  $\lambda_{cyc}$  and  $\lambda_{idt}$  in Equation (5) were 10 and 0.5, respectively.

**Synthetic Non-defect Image Generation** In order to obtain defect-removed images from real defect images, we performed PatchMatch. The defective region was located manually and re-drawn through the approximate nearest-neighbor algorithm. For texture synthesis,  $\mathcal{N}_{TS}$  was trained on images of  $160 \times 160$  resolution with the architecture introduced by Bergmann *et al.* [49]. The network was learned for 100 epochs to generate fake texture images from the latent variables. For the channel dimension of the latent variables, we set the dimension of the three parts as  $d^l = 10$ ,  $d^s = 0$ , and  $d^p = 2$ . With the trained network, we produced synthesized texture images of  $640 \times 640$  resolution and then randomly cropped the images to obtain small texture patches.

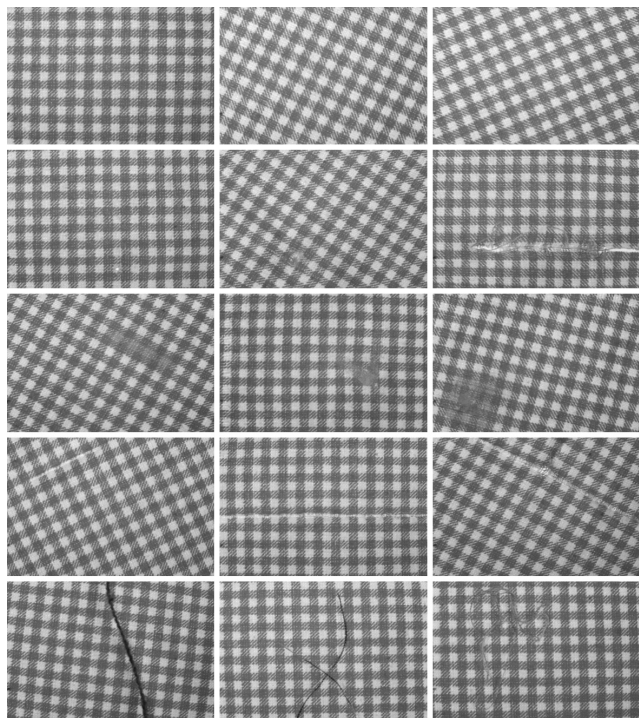


FIGURE 5. Examples of original images of the  $E0$ ,  $E1$ ,  $E2$ ,  $E3$ , and  $E4$  in the  $C3R1$ .

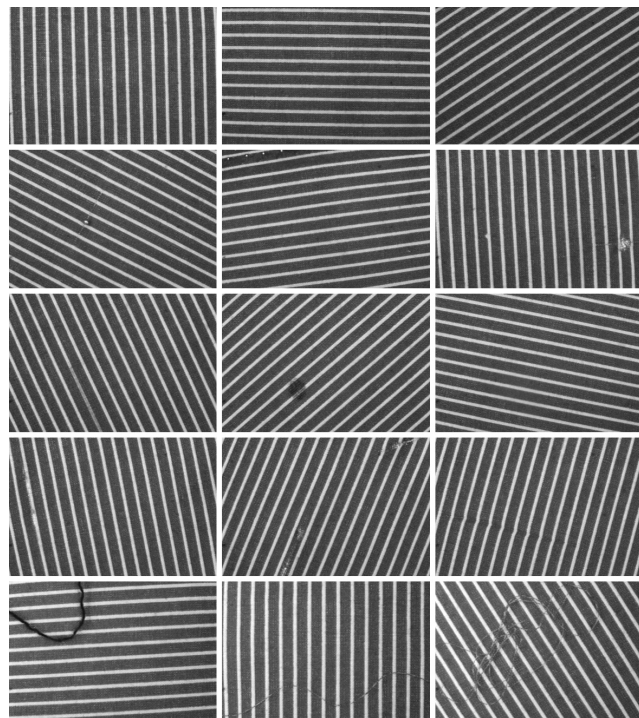


FIGURE 6. Examples of original images of the  $E0$ ,  $E1$ ,  $E2$ ,  $E3$ , and  $E4$  in the  $C3R3$ .

TABLE 3. The four subsets of TILDA dataset for golden template generation.

Subset	Domain	Defect images		Non-defect images	
		real	synthetic	real	synthetic
$C3R1\_real$		300	0	300	0
$C3R1\_syn$		0	10,000	0	10,000
$C3R3\_real$		250	0	250	0
$C3R3\_syn$		0	10,000	0	10,000

**Golden Template Generation** To perform golden template generation, we trained  $\mathcal{N}_{GT}$  on defect images of  $256 \times 256$  resolution and non-defect images of  $256 \times 256$  resolution with the aforementioned architecture in synthetic defect image generation. In addition to random flip in training phase, random rotation with a maximum of  $\pm 5^\circ$  was performed to achieve a generalization performance of the representation. The coefficients  $\lambda_{cyc}$  and  $\lambda_{idt}$  in Equation (8) were both 10.

**Defect Segmentation** We chose the parameters of the patch size and thresholding values empirically. The pSAD results were calculated with a patch whose size ranges from  $5 \times 5$  to  $21 \times 21$ . Hysteresis thresholding was performed using an upper bound  $T_u$  and a lower bound  $T_l$ . The upper bound  $T_u$  and the lower bound  $T_l$  were multiples of 0.02 with a constraint that  $T_u$  should be between 0.5 and 0.98, and that  $T_l$  should be lower than  $T_u$ .

**B. DATASET**

We experimented with three periodic textures. One is the images of defects in semiconductor wafers, and the others are those of defects in textiles. Each of them has distinctive defects and periodic textures.

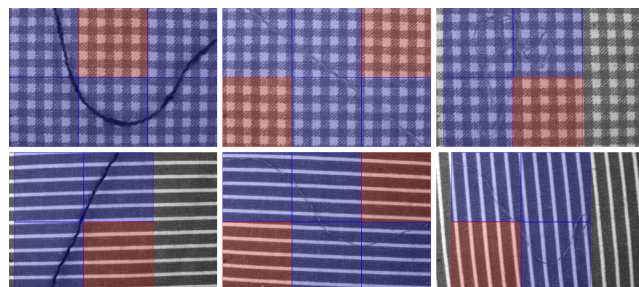


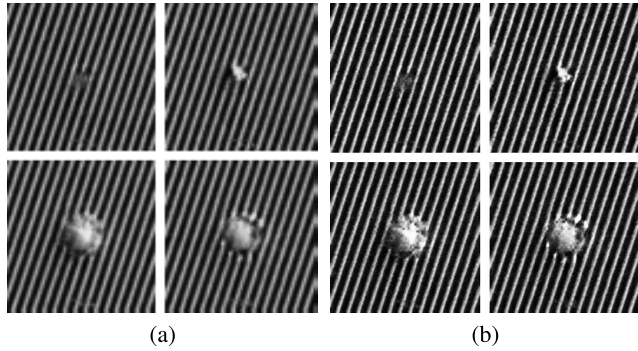
FIGURE 7. Acquisition of defect patches from original defect images in the  $C3R1$  and  $C3R3$ .

**Our Dataset** In order to demonstrate our proposed framework, we experimented with our private dataset. This dataset is about defects occurring in semiconductors, the images of which were captured by scanning electron microscope (SEM). There are seven types of defects in the dataset, and the sample images and their descriptions of our dataset are shown in Table 1. The dataset contains 264 grayscale defect images of  $480 \times 480$  resolution with periodic textures; however, there are no non-defect images.

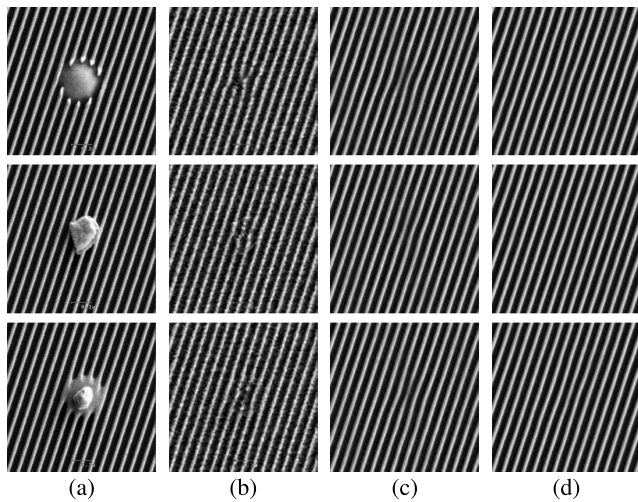
We resized the original images to images of  $256 \times 256$  resolution and used the resized images for the experiments. For the golden template generation and data augmentation process, 200 images were randomly selected as the training data. The performance on this dataset was evaluated with the others.

Because our dataset does not include non-defect images, we used some non-defect images acquired at different scales to construct the *real* subset. By utilizing the scale





**FIGURE 8.** Synthetic defect image generation using our dataset: (a) low-resolution images, and (b) high-resolution images.

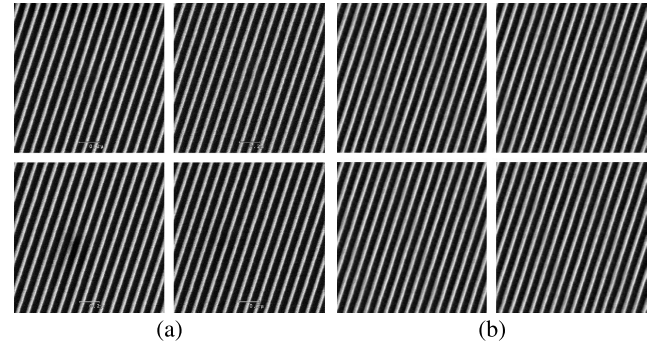


**FIGURE 9.** Golden template generation for the test images: (a) input images, and generation results of the networks trained with *identity mapping loss* on (b) *real* subset, (c) *real+syn* subset, and (d) *syn* subset.

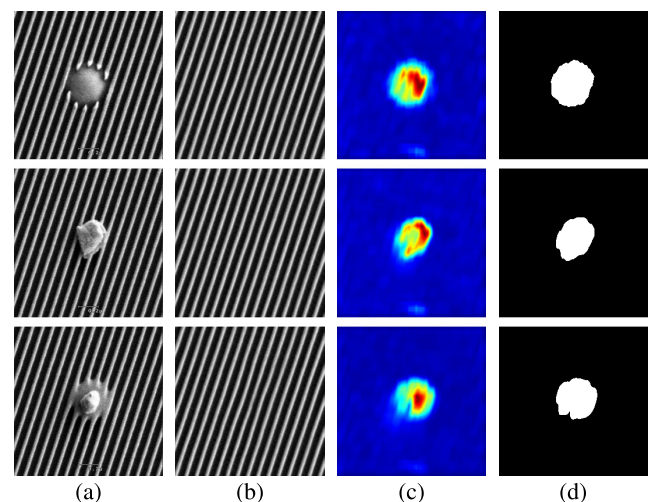
information in the defect and non-defect images, we resized the non-defect images to make the textures in the non-defect images similar to the patterns of the defect images. Then, we randomly cropped the resized non-defect images as the size of real defect images, as shown in Figure 4. Consequently, we obtained 200 non-defect images with the modification scheme. As shown in Table 2, the *real* subset includes 200 real defect and non-defect images. The *real+syn* subset contains 200 real defect and synthetic non-defect images. The *syn* subset consists of 10,000 synthetic defect and non-defect images.

**TILDA** In addition, we utilized a public dataset, TILDA textile texture-database [50], to apply our framework to other periodic textures. This is a dataset of defects in textiles, and there are eight types of fabric. Among the textiles, we used the subsets, {*C3R1*, *C3R3*}, which have periodic structures. In the subcategories of the two subsets, we used {*E1*, *E2*, *E3*, *E4*} as target defects and {*E0*} as defect-free textures. Each subcategory contains 50 grayscale images of  $768 \times 512$  resolution and the sample images of the subcategories in the two subsets are shown in Figures 5 and 6.

In order to be adapted for the proposed framework, the original non-defect images were divided into six patches



**FIGURE 10.** Synthetic non-defect image generation using our dataset: (a) defect-removed images, and (b) synthesized texture patches.

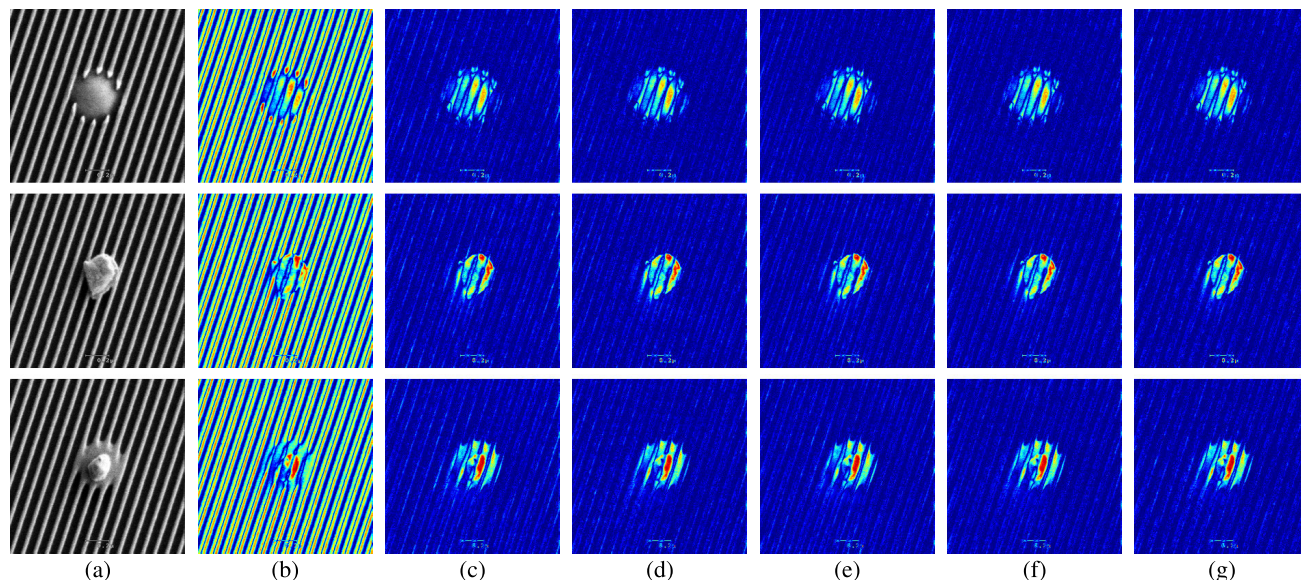


**FIGURE 11.** Post-processing for defect segmentation: (a) input images, (b) golden templates, (c) pSAD results, and (d) hysteresis thresholding results.

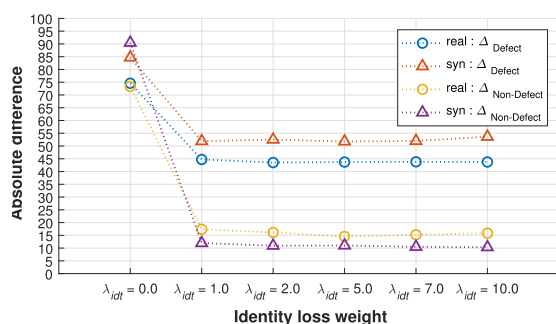
of size  $256 \times 256$  without overlap. For defect images, first, we manually labeled defects in a pixel-wise manner. Based on the annotation results, we set the smallest ROI which covers all the defective regions in the image. The ROI has a constraint that the center of the ROI is equal to that of defects and the width and height of ROI are multiples of 256. With this constraint, we divided the ROI into patches of size  $256 \times 256$ . Among the acquired patches, those which contain defects less than 100 pixels were discarded. As shown in Figure 7, the patches covered with the blue bounding boxes were obtained as defect images, whereas the regions in the red boxes were not used. Accordingly, we obtained 397 and 323 defect patches in the {*C3R1*, *C3R3*}, respectively. In order to train networks for the golden template generation and data augmentation process, 300 and 250 images were arbitrarily chosen. The other images were used as test data.

### C. QUALITATIVE RESULTS OF THE PROPOSED FRAMEWORK

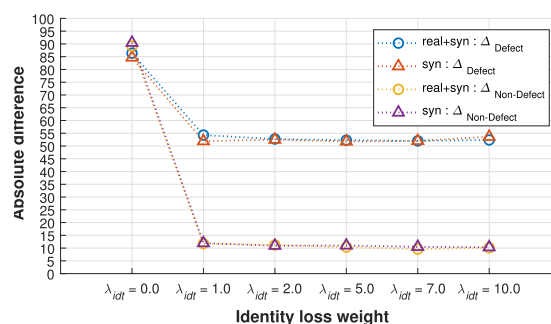
To verify the effectiveness of our data augmentation process and the *identity mapping loss*, we trained the network for golden template generation with the three subsets {*real*, *real+syn*, and *syn*} of our dataset.



**FIGURE 12.** Absolute difference images of the input image and its golden template per identity mapping loss: (a) input images, and absolute difference between the input image and its golden template of networks trained on *syn* subset with *identity mapping loss* using (b)  $\lambda_{idt} = 0$ , (c)  $\lambda_{idt} = 1$ , (d)  $\lambda_{idt} = 2$ , (e)  $\lambda_{idt} = 5$ , (f)  $\lambda_{idt} = 7$ , and (g)  $\lambda_{idt} = 10$ .



**FIGURE 13.** Average values of absolute difference image in both defective and defect-free regions for the *real* and *syn* subsets of our dataset.



**FIGURE 14.** Average values of absolute difference image in both defective and defect-free regions for the *real+syn* and *syn* subsets of our dataset.

Through our data augmentation process, we generated synthetic defect images and synthetic non-defect images using our dataset, as shown in Figures 8 and 10. The resulting non-defect images seemed to be almost the same as the golden templates of the real defect images; however, the generated defect images did not look like completely the real ones. Nonetheless, we achieved remarkable results with them. With many synthetic images, the network for golden template generation could learn a robust mapping from various defective regions to defect-free textures.

We acquired the most reasonable results for the test images when the network for golden template generation was trained on the *syn* subset, as shown in Figure 9. The defective regions in the input image changed as normal, and it was difficult to distinguish where the original defective region had been. When the network was trained on the *real+syn* subset, the defect-free regions in the golden template were very similar to those in the input image. By contrast, the regions in the golden template where defects had existed were slightly different from the defect-free regions.

As shown in Figure 11, our framework generated clean defect-free images from the input defective images. The post-processing for defect segmentation could be seemed to be superfluous; however, we dealt with the noises that are due to the great change of the intensity in a normal pixel. Despite the different intensity changes of the pixels in the defective region, we detected the region as one defect. With this step, we achieved better performance.

As shown in Figure 12, the *identity mapping loss* has a great effect on the golden template generation. The golden template generator trained without the *identity mapping loss* missed the periodicity of the textures, so that the difference values in the defect-free regions were as large as those in the defective regions. On the contrary, the generator trained with the loss represented the periodicity while removing defects in the images.

In order to analyze these results numerically, we obtained the average values of both the defective and defect-free regions in the absolute difference images with region-level labeling, as shown in Figures 13 and 14. Because the

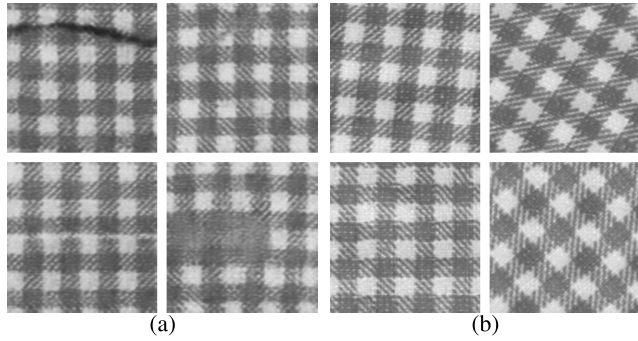


FIGURE 15. Data augmentation using the *C3R1* subset: (a) synthetic defect image, and (b) synthetic non-defect image.

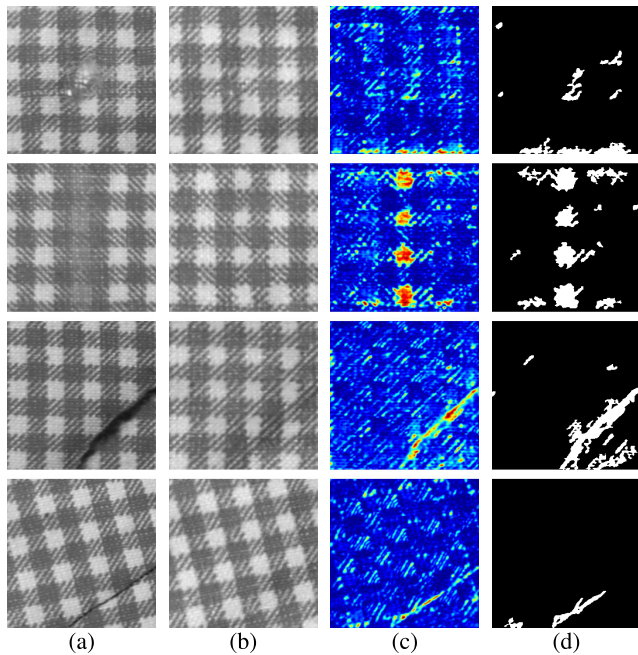


FIGURE 16. Post-processing on the *C3R1* subset: (a) input images, (b) golden templates, (c) pSAD results, and (d) hysteresis thresholding results.

non-defect images in the *real* subset are quite different from the defect-free regions in the real defect images, the average value of the defect-free region was the largest among the three subsets. The network trained on the *syn* subset reduced the average value of the absolute difference in the defect-free region. This suggests that our data augmentation process made more realistic images. It was difficult to discriminate between the defective regions and the defect-free regions in the absolute difference images when the *identity mapping loss* was not utilized; however, the use of the loss made the gap between the average values of the two regions clear.

The very similar method, DefectGAN [42], does not include the *identity mapping loss* in the total loss function. The model trained without the loss created the textures shifted from those in the input image. This could make defect-free regions be segmented as defects. In addition, with our data augmentation process, the regions where defects existed in the input image could turn into the textures more similar to

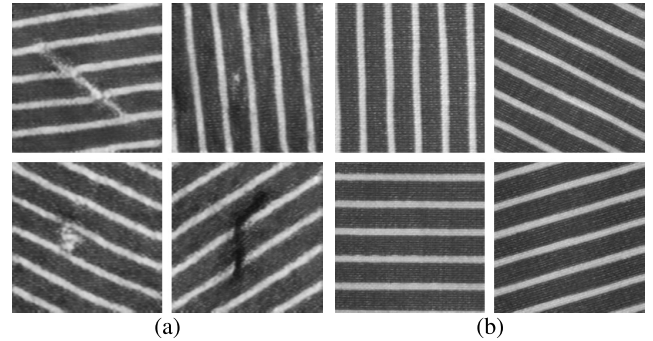


FIGURE 17. Data augmentation using the *C3R3* subset: (a) synthetic defect image, and (b) synthetic non-defect image.

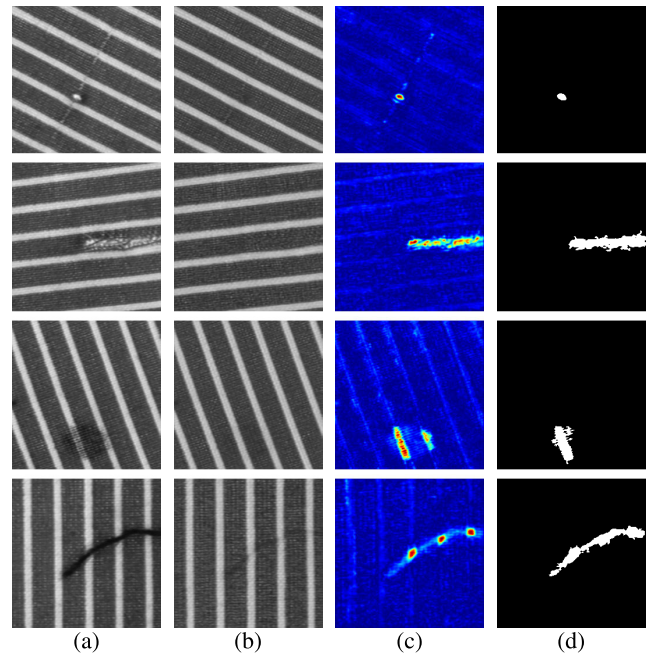
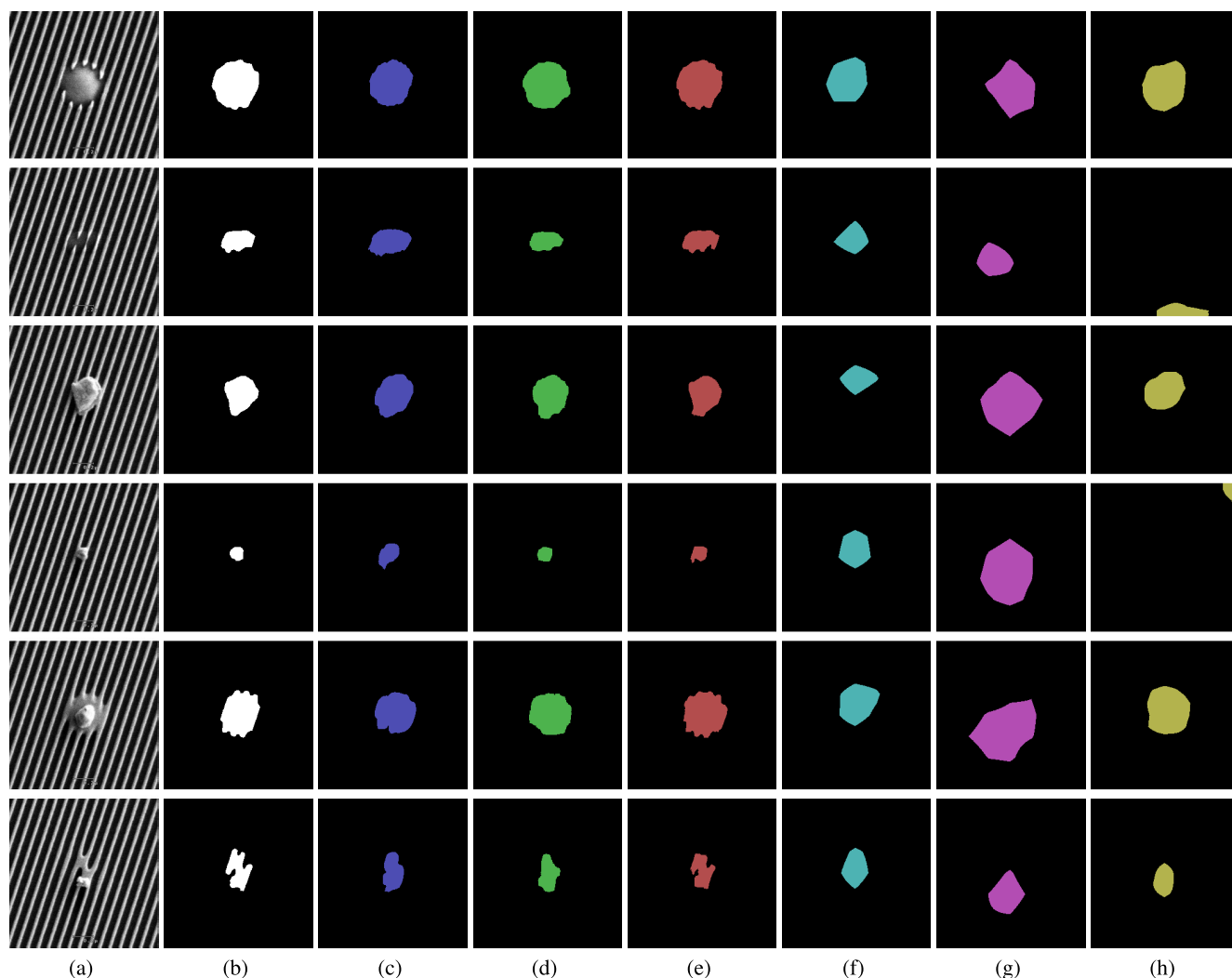


FIGURE 18. Post-processing on the *C3R3* subset: (a) input images, (b) golden templates, (c) pSAD results, and (d) hysteresis thresholding results.

those of defect-free regions. For these reasons, our framework could be differentiated from DefectGAN for segmenting defects on periodic textures.

Additionally, we applied our framework to the TILDA dataset and adjusted the training scheme and hyperparameters of some networks for data augmentation of the TILDA dataset. Because directions of the pattern in the subsets are different, the two networks,  $\mathcal{N}_{FD}$  and  $\mathcal{N}_{TS}$ , for synthetic defect and non-defect image generation were trained by direction of pattern. We generated synthetic defect and non-defect images, the number of which was as same as the number of those of our dataset.

As shown in Figures 15 and 17, we generated synthetic defect and non-defect images using the *C3R1* and *C3R3* subsets. Unlike the generated non-defect images, synthetic defect images were of poor quality, relatively. As shown



**FIGURE 19.** Results of defect segmentation on our dataset: (a) input images, (b) ground truth, (c) ours, (d) FCN [19], (e) AdapNet++ [51], (f) CAM [36] (DenseNet [52]), (g) CAM [36] (ResNet [53]), and (h) CAM [36] (SqueezeNet [54]).

in Figures 16 and 18, our framework made decent golden templates for input defect images; however, there was a little vestige of the defective region in the generated golden template.

**D. QUANTITATIVE EVALUATION OF THE PROPOSED FRAMEWORK**

In order to demonstrate the competitiveness of our framework on defect segmentation, we compared our framework with the three methods: CAM [36], FCN [19], and AdapNet++ [51].

The two supervised semantic segmentation networks were trained on the *real* subset of our dataset. With the real defect images and their region-level labels, FCN and AdapNet++ learned the semantic context for 200 and 100 epochs, respectively. Our framework and CAM were trained on the *real*, *real+syn*, and *syn* subsets of our datasets with only image-level labels. They were trained for 200 epochs, except that our framework learned two mapping functions between the defect and non-defect domains on the *syn* subset for

50 epochs. In the training and testing phase of the CAM, several backbone networks were employed: DenseNet [52], ResNet [53], and SqueezeNet [54]. The segmentation masks of the CAM were obtained by applying the same thresholding scheme of our framework to the resulting heatmaps of the CAM. We selected the thresholding parameters to achieve the best performance of the CAM.

As shown in Figure 19, our framework produced remarkable segmentation results for various defects. These outcomes were comparable to those of the other two supervised methods. While the CAMs of the three backbone networks localized defects, our framework segmented the defective region more accurately. Specifically, our framework showed better segmentation results for the relatively smaller defects, compared with the three CAMs. These results could be due to the downsampling in the backbone networks of the CAM. The spatial resolution of the feature map was reduced through pooling layers or strided convolutions, and the resulting heatmap was enlarged to the original input size.

TABLE 4. The segmentation performance on our dataset.

Method	Level of supervision	mIoU (%)
FCN [19]	Supervised	86.15
AdapNet++ [51]	Supervised	<b>92.91</b>
CAM [36] (DenseNet [52])	Weakly-supervised	60.84
CAM [36] (ResNet [53])	Weakly-supervised	34.34
CAM [36] (SqueezeNet [54])	Weakly-supervised	49.44
<b>Ours</b> (With <i>real</i> subset)	Weakly-supervised	63.77
<b>Ours</b> (With <i>real+syn</i> subset)	Weakly-supervised	80.98
<b>Ours</b> (With <i>syn</i> subset)	Weakly-supervised	<b>82.33</b>

TABLE 5. Comparison of the inference time.

Method	Image resolution	Time (ms)
FCN [19]	224 × 224	42.12
AdapNet++ [51]	768 × 384	34.31
<b>Ours</b>	256 × 256	<b>17.84</b>

We adopted the intersection over union (IoU) to evaluate the segmentation performance [27]. The IoU of the two regions,  $R_p$  and  $R_{gt}$ , is expressed as:

$$IoU(R_p, R_{gt}) = \frac{\text{area}(R_p \cap R_{gt})}{\text{area}(R_p \cup R_{gt})}, \quad (10)$$

and the mean IoU (mIoU) is calculated as:

$$mIoU = \frac{\sum_{k=1}^N \text{area}(R_p^k \cap R_{gt}^k)}{\sum_{k=1}^N \text{area}(R_p^k \cup R_{gt}^k)}, \quad (11)$$

where  $\text{area}(R_p \cup R_{gt})$  denotes the union of the predicted segmentation mask and the ground truth,  $\text{area}(R_p \cap R_{gt})$  indicates their intersection, and  $N$  is the number of test data. In this work, the mIoU was measured with only the defective region.

As shown in Table 4, the performance of our framework was enhanced with the proposed data augmentation process. Moreover, our framework outperformed the CAMs by a huge margin. Unfortunately, the performances of the two supervised segmentation methods were superior to those of our framework. However, given the qualitative segmentation results, it seemed that our framework achieved a decent performance of defect segmentation. Above all, our framework produced competitive segmentation results without pixel-wise labeling. In addition, we compared the inference time of the proposed framework with those of the two segmentation networks. As shown in Table 5, our framework showed the fastest inference time. In our framework, the inference time of the golden template generation and defect segmentation was about 16 and 2 ms, respectively.

E. DISCUSSION

There were a few limitations and failure cases of our framework, as shown in Figure 20.

Because pSAD was applied to deal with noises in the absolute difference images between the input image and

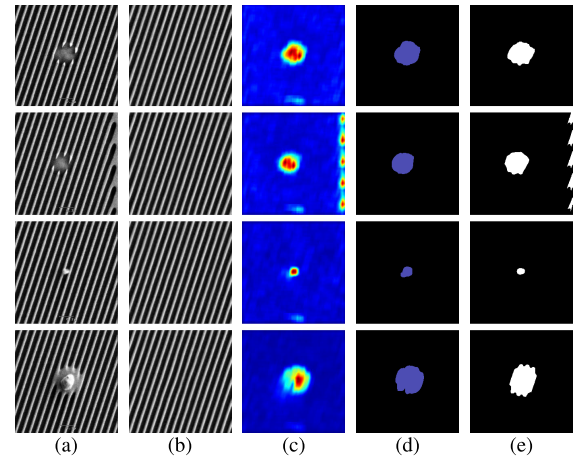


FIGURE 20. Failure cases on our semiconductor dataset: (a) input images, (b) golden templates, (c) pSAD results, (d) hysteresis thresholding results, and (e) ground truth.

its golden template, our framework produced less detailed results than the supervised methods. Hysteresis thresholding was employed to determine defective regions; however, there were the limitations that the number and the area of defects were heavily dependent on the two parameters of the thresholding. When a small foreign body was between line patterns, the segmentation result of our framework was wider than its actual region. When some line patterns were broken or bridged, a few defective regions were not detected because the intensities of the regions hardly changed.

TILDA dataset is more challenging than our semiconductor dataset because the textures of the two subsets are less strict and more complex than those of our dataset. Particularly, the directions of the patterns in the original images are different from those of one another. Furthermore, there are some indistinct defects that are indistinguishable from defect-free textures. The qualitative results of the golden template generation seemed to be decent with the naked eye; however, the segmentation results were not good. This was because the intensities in defective regions were similar to those in defect-free textures. The structures of the defects in the two subsets are more complex than that of the defects in our dataset. Detailed shapes of defects and surrounding textures could not be formed in the synthetic defect generation.

Particularly, the patterns in the *C3R1* subset have both global and local periodicity. This means that the images in the *C3R1* subset have check patterns globally and line patterns locally. Unfortunately, it was difficult to reproduce these distinctive textures in the synthetic defect images. For these reasons, the quantitative performance on the TILDA dataset was not good, as shown in Tables 6 and 7.

V. CONCLUSION

In this paper, we propose a weakly-supervised defect segmentation framework for periodic textures and a data augmentation process applicable to our framework. We

**TABLE 6. The segmentation performance on the C3R1 subset.**

Method	Level of supervision	mIoU (%)
FCN [19]	Supervised	69.39
AdapNet++ [51]	Supervised	<b>78.26</b>
<b>Ours</b> (With augmentation)	Weakly-supervised	<b>12.27</b>

**TABLE 7. The segmentation performance on the C3R3 subset.**

Method	Level of supervision	mIoU (%)
FCN [19]	Supervised	42.01
AdapNet++ [51]	Supervised	<b>69.33</b>
<b>Ours</b> (With augmentation)	Weakly-supervised	<b>14.84</b>

generated a golden template from an input defect image and segmented the defective region by applying straight-forward post-processing to the two images. Furthermore, we found that the identity mapping loss is crucial to the golden template generation of defect images with periodic textures. As a result, we localized the defects in a pixel-wise manner without region-level labeling. Through the proposed augmentation process, we created synthetic defect and non-defect images even from only real defect images. With the augmented data, the golden template generator made more plausible results, and the segmentation performance of our framework was enhanced. The proposed framework was qualitatively and quantitatively compared to other defect segmentation methods on periodic texture images with various defects. The experimental results suggest that the proposed framework outperformed the CAM-based method and showed results comparable to those of the supervised segmentation in strictly periodic textures.

In future work, we plan to simplify the whole proposed framework and develop the data augmentation process to make more realistic images. Since the difference of intensities showed a limitation for segmenting defective regions, we plan to develop more in-depth post-processing. Particularly, we will try to improve the quality of the golden template generation for loosely periodic textures and to segment the detailed structure of defects.

## REFERENCES

- [1] D. Lopez-Perez and J. Antonino-Daviu, "Application of infrared thermography to failure detection in industrial induction motors: Case studies," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 1901–1908, May 2017.
- [2] R. Usamentiaga, Y. Mokhtari, C. Ibarra-Castanedo, M. Klein, M. Genest, and X. Maldague, "Automated dynamic inspection using active infrared thermography," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5648–5657, Dec. 2018.
- [3] M. Delgado-Prieto, J. A. Carino-Corrales, J. J. Saucedo-Dorantes, R. de Jesus Romero-Troncoso, and R. A. Osornio-Rios, "Thermography-based methodology for multifault diagnosis on kinematic chain," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5553–5562, Dec. 2018.
- [4] M. Dhimish, P. Mather, and V. Holmes, "Novel photovoltaic hot-spotting fault detection algorithm," *IEEE Trans. Device Mater. Rel.*, vol. 19, no. 2, pp. 378–386, Jun. 2019.
- [5] Y. Song, J. A. Turner, Z. Peng, C. Chen, and X. Li, "Enhanced ultrasonic flaw detection using an ultrahigh gain and time-dependent threshold," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 65, no. 7, pp. 1214–1225, Jul. 2018.
- [6] V. V. B. Reddy, I. C. Ume, A. Mebane, K. Akinade, B. Rogers, C. Guirguis, P. B. Patel, K. Derksen, and R. Case, "Evaluation of FCBGA package subjected to four-point bend reliability test using fiber array laser ultrasonic inspection system," *IEEE Trans. Compon., Package., Manuf. Technol.*, vol. 9, no. 7, pp. 1219–1226, Jul. 2019.
- [7] Y. Guo, Z. Xiao, L. Geng, J. Wu, F. Zhang, Y. Liu, and W. Wang, "Fully convolutional neural network with GRU for 3D braided composite material flaw detection," *IEEE Access*, vol. 7, pp. 151180–151188, 2019.
- [8] Z.-D. Tsai and M.-H. Perng, "Defect detection in periodic patterns using a multi-band-pass filter," *Mach. Vis. Appl.*, vol. 24, no. 3, pp. 551–565, Apr. 2013.
- [9] B. H. Khalaj, H. K. Aghajan, and T. Kailath, "Automated direct patterned-wafer inspection," in *Proc. Mach. Vis. Appl. Ind. Inspection*, May 1993, pp. 266–273.
- [10] P. Xie and S.-U. Guan, "A golden-template self-generating method for patterned wafer inspection," *Mach. Vis. Appl.*, vol. 12, no. 3, pp. 149–156, Oct. 2000.
- [11] N. G. Shankar and Z. W. Zhong, "Defect detection on semiconductor wafer surfaces," *Microelectron. Eng.*, vol. 77, nos. 3–4, pp. 337–346, Apr. 2005.
- [12] N. G. Shankar and Z. W. Zhong, "A rule-based computing approach for the segmentation of semiconductor defects," *Microelectron. J.*, vol. 37, no. 6, pp. 500–509, Jun. 2006.
- [13] D.-M. Tsa and S.-K. Wu, "Automated surface inspection using Gabor filters," *Int. J. Adv. Manuf. Technol.*, vol. 16, no. 7, pp. 474–482, Jun. 1990.
- [14] C.-H. Chan and G. K. H. Pang, "Fabric defect detection by Fourier analysis," *IEEE Trans. Ind. Appl.*, vol. 36, no. 5, pp. 1267–1276, Sep. 2000.
- [15] A. Kumar and G. K. H. Pang, "Defect detection in textured materials using Gabor filters," *IEEE Trans. Ind. Appl.*, vol. 38, no. 2, pp. 425–440, Aug. 2002.
- [16] Y.-H. Liu, Y.-K. Huang, and M.-J. Lee, "Automatic inline defect detection for a thin film transistor–liquid crystal display array process using locally linear embedding and support vector data description," *Meas. Sci. Technol.*, vol. 19, no. 9, p. 95501, 2008.
- [17] Y.-H. Liu, S.-H. Lin, Y.-L. Hsueh, and M.-J. Lee, "Automatic target defect identification for TFT-LCD array process inspection using kernel FCM-based fuzzy SVDD ensemble," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 1978–1998, Mar. 2009.
- [18] J. Yu and X. Lu, "Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 1, pp. 33–43, Feb. 2016.
- [19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [20] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [21] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "AdapNet: Adaptive semantic segmentation in adverse environmental conditions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4644–4651.
- [22] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric defect detection using activation layer embedded convolutional neural network," *IEEE Access*, vol. 7, pp. 70130–70140, 2019.
- [23] R. Wang, Q. Guo, S. Lu, and C. Zhang, "Tire defect detection using fully convolutional network," *IEEE Access*, vol. 7, pp. 43502–43510, 2019.
- [24] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "PGA-net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020.
- [25] M. Zontak and I. Cohen, "Defect detection in patterned wafers using anisotropic kernels," *Mach. Vis. Appl.*, vol. 21, no. 2, pp. 129–141, Feb. 2010.
- [26] L. Geng, Q. Meng, Z. Xiao, and Y. Liu, "Measurement of period length and skew angle patterns of textile cutting pieces based on faster R-CNN," *Appl. Sci.*, vol. 9, no. 15, p. 3026, Jul. 2019.
- [27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
- [28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.
- [29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

- [30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.
- [31] F. Tajeripour and S. Fekri-Ershad, "Developing a novel approach for stone porosity computing using modified local binary patterns and single scale retinex," *Arabian J. for Sci. Eng.*, vol. 39, no. 2, pp. 875–889, Feb. 2014.
- [32] S. Fekri-Ershad and F. Tajeripour, "Multi-resolution and noise-resistant surface defect detection approach using new version of local binary patterns," *Appl. Artif. Intell.*, vol. 31, nos. 5–6, pp. 395–410, Jul. 2017.
- [33] B.-F. Cao, J.-Q. Li, and N.-S. Qiao, "Nickel foam surface defect detection based on spatial-frequency multi-scale MB-LBP," *Soft Comput.*, vol. 24, no. 8, pp. 5949–5957, Apr. 2020.
- [34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
- [35] S. Marino, A. Smolarz, and P. Beausery, "Potato defects classification and localization with convolutional neural networks," in *Proc. 14th Int. Conf. Qual. Control Artif. Vis.*, vol. 11172, Jul. 2019, Art. no. 111720.
- [36] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
- [37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [39] H. Lin, B. Li, X. Wang, Y. Shu, and S. Niu, "Automated defect inspection of LED chip using deep convolutional neural network," *J. Intell. Manuf.*, vol. 30, no. 6, pp. 2525–2534, Aug. 2019.
- [40] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag. (IPMI)*, 2017, pp. 146–157.
- [41] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: <http://arxiv.org/abs/1511.06434>
- [42] S. Niu, H. Lin, T. Niu, B. Li, and X. Wang, "DefectGAN: Weakly-supervised defect detection using generative adversarial network," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 127–132.
- [43] I. Budvytis, P. Sauer, T. Roddick, K. Breen, and R. Cipolla, "Large scale labelled video data augmentation for semantic segmentation in driving scenarios," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 230–237.
- [44] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*. [Online]. Available: <http://arxiv.org/abs/1712.04621>
- [45] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert, "GAN augmentation: Augmenting training data using generative adversarial networks," 2018, *arXiv:1810.10863*. [Online]. Available: <http://arxiv.org/abs/1810.10863>
- [46] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, Dec. 2018.
- [47] A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca, "Data augmentation using learned transformations for one-shot medical image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8543–8553.
- [48] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patch-Match: A randomized correspondence algorithm for structural image editing," in *Proc. ACM SIGGRAPH Papers SIGGRAPH*, 2009, vol. 28, no. 3, pp. 1–11.
- [49] U. Bergmann, N. Jetchev, and R. Vollgraf, "Learning texture manifolds with the periodic spatial GAN," in *Proc. 34th Int. Conf. Mach. Learn. (JMLR)*, 2017, pp. 469–477.
- [50] *TILDA Textile Texture-Database*. Accessed: Jul. 23, 2020. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/resources/datasets/tilda>
- [51] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1239–1285, May 2020.
- [52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
- [53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [54] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: <http://arxiv.org/abs/1602.07360>



**MINSU KIM** received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include pattern recognition, machine learning, data augmentation, defect segmentation, and vehicle detection.



**HOON JO** received the B.S. and Ph.D. degrees in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2012 and 2020, respectively. He is currently working as a Staff Engineer with System LSI Business in the DS (device solutions) Division, Samsung Electronics Company Ltd. His research interests include machine learning, real-time computer vision solutions, and camera image quality.



**MOONSOO RA** (Member, IEEE) received the B.S. and Ph.D. degrees in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2011 and 2019, respectively. He is currently working as a Founding Member and the CTO of LightVision Inc. His research interests include pattern recognition, machine learning, autonomous vehicle, and video surveillance.



**WHOI-YUL KIM** (Member, IEEE) received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989. From 1989 to 1994, he was with the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas. He joined Hanyang University, in 1994, where he is currently a Professor with the Department of Electronic Engineering. His research interests include health monitoring using mobile devices, visual surveillance, virtual devices, machine vision systems, advanced driver assistance systems, and 3D vision systems in sports, as well as home appliances.

...