Master of Science in Mechanical Engineering

Identifying defects on unclean wafer surface through anomaly detection

이상 탐지를 이용한 불분명한 표면을 가지는
웨이퍼 표면에서의 크랙 검출

February 2023

Graduate School of Mechanical Engineering
Seoul National University

Sunjae Park

Identifying defects on unclean wafer surface through anomaly detection

Advisor Do-Nyun Kim

Submitting a master's thesis of
Mechanical Engineering

October 2022

Graduate School of Mechanical Engineering
Seoul National University

Sunjae Park

Confirming the master's thesis written by
Sunjae Park

December 2022

Chair	Byeng-Dong Youn	(Signature)
Vice Chair	Do-Nyun Kim	(Signature)
Examiner	Jin-kyu Yang	(Signature)

Abstract

Defect detection is a crucial process to improve the productivity and quality of products in the industry. However, defects in the nanoscale-manufacture become difficult to detect, since the shapes of the defects are complex and noises and unclean backgrounds cover the defects frequently. It is laborious and inefficient to utilize human resources for defect detection because the rate of defects in the industry is extremely low and it requires professional knowledge to detect the defects in some cases. Applying an anomaly detection model as a defect detector in the industry is the best solution which will save time and human resources. However, there are many difficulties to apply the data-driven based anomaly detection model to real industry inspection. In our research, we found that our target product wafers contain "resin bleed", which hinders detecting cracks on the wafer surfaces. The resin bleed impedes the anomaly detection on wafers because it is similar to the cracks in the wafer and at the same time it belongs to the normal components. In this paper, we propose a method to improve the crack detection performance of the anomaly detection model by enhancing the edge information of cracks. Our model achieved 96.7% at the image level AUROC and 98.6% at pixel level AUROC by improving 4.5% and 2.0% respectively without additional annotation.

Keyword : Industrial inspection, Machine learning, Anomaly detection, Edge detection, Wafer inspection, Crack
Student Number : 2021-26800

Table of contents

List of tables

List of figures

Figure 14. Ninth results of anomaly detection on blurry cracks.

Figure 15. Tenth results of anomaly detection on blurry cracks.

Figure 16. Eleventh results of anomaly detection on cracks under the resin bleed.

Figure 17. Twelfth results of anomaly detection on normal images with the resin bleed

Figure 18. The extracted edge images by selecting one or two edge features.

Figure 19. The extracted edge images by selecting three or four edge features.

Figure 20. The extracted edge images by selecting five or six or seven edge features.

Chapter 1. Introduction

Defect detection becomes important as the demand for manufacturing at the nanoscale increases. Unlike defects that occur in macroscale manufacturing, defects in the nanoscale manufacturing are hard to discover. Because they have complex shapes and require professional knowledge about the defects for identification. Detecting all the defects that occurred in production is the best situation, however, humans can't supervise all the products that have been produced. Therefore, automatic defect detection is the better alternative for productivity improvement. The defect detection model can inspect every product have been made and define the abnormality in the products. In addition, we do not need abnormal data for the defect detection model. This ability makes the anomaly detection model to be easily applied to the industry as a defect detection tool. Well-trained anomaly detection can be a perfect inspector in the industry and saves time and resources that would have been wasted if the anomaly detection model was not applied.

There are many studies for anomaly detection that shows high performance in the industrial dataset. The rapid development of machine learning and deep learning makes data-driven based anomaly detection models efficient. They learn the normal data distribution in the many different methods and define their abnormality criteria. The anomaly detection models already guarantee high defect detection performance in the industrial benchmark dataset. However, there are many challenges to applying

computer vision models in the real industry field. First, the ratio of defective data to normal data is extremely low [1] since the goal of optimizing manufacturing processes in the industry is to reduce defective products. Thus, a lack of defective data is inevitable, the data-driven anomaly detection methods focus on utilizing normal image data that are very easy to acquire in the industry. Second, the defects can have various shapes, even if they belong to the same crack type [1]. The anomaly detection model needs too many defective images if the model deals with test input images by learning every type of defect that could have been in the manufacture. The strategy to focus on the normal images solves not only the lack of defective data but also various types of defects that possibly occur. Third, complex backgrounds hinder the defects from being detected as problems [2]. In the case of crack detection on concrete roads, the complex texture of the concrete roads hinders crack detection. In our research, a wafer dataset contains 'resin bleed', which belongs to the normal component, hindering crack detection. Our primary object in research is to detect cracks in wafers from the unclean background (e.g., the complexity of background and resin bleed) by using an anomaly detection model. This section introduces related works for solving problems of wafer industrial inspection in subsections: (1.1) Anomaly detection (1.2) Wafer defect detection (1.3) Crack detection

1.1. Anomaly Detection

Anomaly detection is an active research field for outlier detection in machine learning. Visual anomaly detection can be used as defect detection when it is used in an industrial inspection. Most anomaly detection models infer the outlier by learning the normal data distribution. Before the deep learning-based methods are researched, SIFT [3] and HOG [4] utilize the shallow features in the image's gray value. Deep learning-based methods utilize the good representation capability of convolution neural networks, achieving high performance in anomaly detection. Anomaly detection can be grouped into two categories: image-level and pixel-level anomaly detection. Image-level anomaly detection models only decide whether a given image is abnormal or not. In [5], a traditional machine learning technique, SVM, is used for One-Class SVM (OC-SVM) that searches the support vector at the normal data distribution composed on the hyperplane. Similar to [5], [6] trains the deep convolution network to map normal images into the points that construct a hypersphere in the latent space. Then, the [6] infers the outlier by measuring the distance between the hypersphere and the point to which the given test image is mapped. Pixel-level anomaly detection models decide whether or not a region of the image is abnormal, as well as an abnormality on individual images. The anomaly detection models for a pixel-level decision can be categorized into image reconstruction and feature modeling [7]. Image reconstruction-based methods utilize variational autoencoder

(VAE) [8] and generative adversarial network (GAN) [9] as backbone models for image reconstruction. Both [10] and [11] use autoencoder for image reconstruction, but [10] uses image structure similarity measure (SSIM) [12] for measuring the degree of the unconstructed region of the image, and [11] uses l_2−distance [13] for measurement. The image reconstruction method using generative adversarial networks trains the networks to generate the normal input images [14, 15, 16]. The recent feature−based methods utilize the representation capability of deep convolution networks pre−trained on ImageNet [17]. The SPADE [18] uses a memory bank technique for saving deep features of normal images and refers to the features when the model infers the given test images by using the nearest neighbor method. Then, the PaDiM [19] solves the computation complexity problem in the SPADE, which costs time for the inference. The PaDiM solves the problem by assuming that the whole set of normal features constructs the normal distribution. The PatchCore, which is our research's backbone model, avoids the computation complexity in the SPADE [18] by core set−subsampling and achieves the highest performance in the benchmark dataset [1]. The anomaly detection model can be evaluated at both the image−level and the pixel−level AUROC (Area Under Receiver Operator Curve).

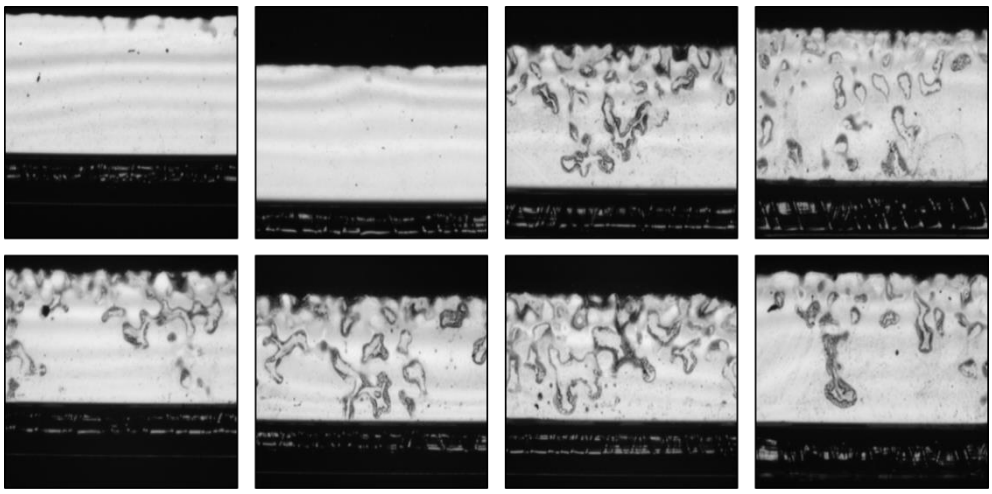
1.2. Wafer Defect Inspection

Before computer vision methods are applied, conventional techniques, mainly operated by manpower, are used for wafer surface inspection. For example, thin-crack on wafers can be detected by measuring sound from mechanically exciting vibratory modes in silicon wafers [19]. However, deep learning-based methods are more efficient both in terms of time and resources than the existing defect detection techniques [20]. Many defect detection techniques are being researched to detect defects throughout the semiconductor manufacturing process. The research for wafers can be grouped into three categories: (1) defect detection on wafers (2) Inference of root problems in semiconductor manufacturing (3) Wafer die defect inspection. As in [21], defect detection on the wafer entails detecting and segmenting defects on the wafer while taking silicon wafer properties into account. For example, in [21], a modified U-Net [22] is used for defect detection on a polycrystalline silicon wafer. Inhomogeneous texture [23] in polycrystalline silicon wafers and wafer contamination [24] caused by the dicing process hinder the detection of defects on the wafers. Other studies for inferring problems in semiconductor manufacturing use a wafer defect map, which is a set of wafer dies marked with a normal or abnormal label [25, 26]. The wafer map that is used to find out the problems in manufacturing steps is made up of labels of dies. Wafer die defect inspection decides whether given dies are normal or not for quality control. Neural networks are trained with pre-processed

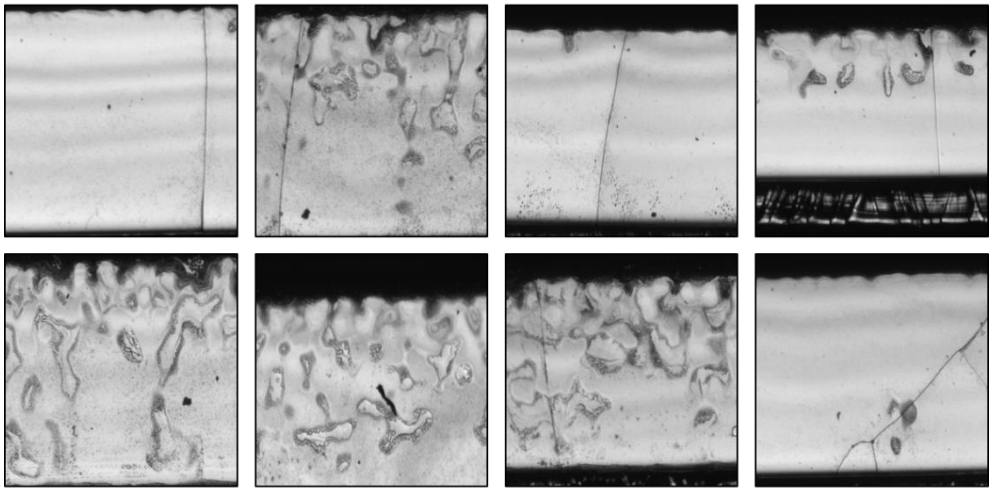
die images for geometric and texture inspection in [27] and [28]. Three categories of defect inspection on wafers are applied to the whole manufacturing process for improving yield.

1.3. Crack Detection

Detecting cracks is a crucial process for manufacturing systems, especially for products with thin structures that are vulnerable to cracks. Since cracks are able to propagate under fatigue stress and repeated loading, early crack detection prevents subsequent loss. Image-based methods are actively researched for reasons that they are efficient and non-destructive. The image-based methods utilize many types of images, including camera, ultrasonic, and IR images to detect cracks. Traditional image-based approaches consider edge information [29] or image correlation [30]. However, the traditional methods have some difficulties that are due to the random shape and size of cracks and noises from the surroundings. Unlike crack detection through machine learning, the deep learning-based method shows high detection performance. [31] proposed image-based method using YOLO v2 [32], which is a famous object detection model, to detect cracks. Also classification model (e.g., GoogLeNet [33]) is able to be used to detect cracks on concrete surfaces [34]. Deep learning-based models can be evaluated with accuracy, recall, and F1-score.



(a)



(b)

Figure 1 Images from the wafer surface boundary dataset (a) examples from the wafer training dataset (b) examples from the wafer test dataset The test dataset is classified into one normal class and two abnormal classes [Normal, Cracks, Resin Bleed Cracks]. Wavy patterns are at both (a) and (b). The resin bleed is located at the random position of the wafer surface regardless of the class.

Chapter 2. Edge-Enhanced Anomaly Detection

In this section, we propose a new method to detect cracks on wafer surfaces by enhancing edge information in the cracks. It becomes difficult to detect cracks because many components appearing on the wafer surface weaken the crack's edge information. Since the capability to detect outliers in the anomaly detection model is based on robustly learning the normal data distribution from the dataset which is only composed of normal images, these components, confusing the whole images not to clear, limit the performance of the anomaly detection model. The impeding components originate from the features of the product itself, the wafer. The difficulties originated from the properties of the wafer. Because of the wafer's structure, most cracks are thin, and resin bleed from the wafer covers cracks. Both of these factors cause the crack region to lose its edge information. Thus, to supplement the crack's edge information, we combine an edge detection model with the anomaly detection model. The method to augment the edge information is introduced in the rest of this section.

In the study, we select DexiNed [35], which is an edge detection model, for extracting edge information because the DexiNed shows high performance on the edge detection task, and we can choose features optionally from seven features of the DexiNed based on the depth of the deep learning model. The edge detection is performed in the pre-processing step, and the augmentation of edge information is performed after the feature extraction step. Structural changes to

the original anomaly detection model are shown in Figure 2. The anomaly detection augmented with edge information consists of the following four steps: (2.1) edge information extraction, (2.2) normal data set augmented with edge information, (2.3) compression of normal data set, and (2.4) Inference for anomalies.

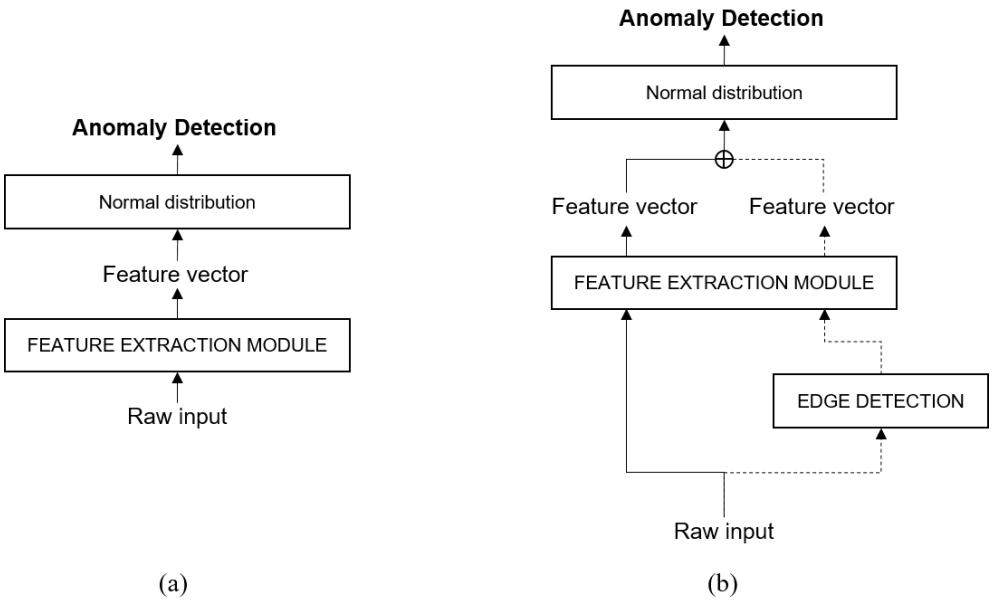


Figure 2 (a) Simplified schematic of the existing anomaly detection model, PatchCore (b) Simplified schematic of our modified anomaly detection model, PatchCore. A step for edge information by utilizing the pre-trained edge detection model, DexiNed is added into the preprocessing stage. The summation of the original information and the edge information is added to the training stage.

2.1. Edge Information Extraction

To utilize the edge information in anomaly detection, we use the DexiNed, the edge detection model, for extracting edge features from the wafer images. We can use pre-trained DexiNed on a BIPED v2 dataset because the DexiNed has a strong generalization to images that are not used in model training. When the wafer images are

entered into the edge detection model, the model outputs seven features: six features come from each convolution neural network block, and the last output can be obtained from an additional convolution block that receives the sixth output as an input. The simplified structure of DexiNed is shown in Figure 3. After the model outputs seven features, the pre-trained up-sampling convolution blocks unify each feature with a different resolution to have a unified resolution. The recommended method to extract the most accurate edge image is to average all of the model' s features, but the most advantageous method for anomaly detection on the wafer dataset is to select certain features based on the impact of each feature. The advantageous method is to extract edge information to average the fourth, fifth, and sixth features. As a result, we can get gray-scale images as auxiliary images, and examples of the result are shown in Figure 4. As the last step for pre-processing, a gray-scale image should be concatenated with itself twice so that we can have a three-channel edge image. In Figure 4, we can see that the model whitens the wafer's background, which has a wavy pattern in some examples. Also, blurry cracks on the wafer in Figure 4 (b) have become clear, especially an example in the fourth row, which is difficult to visually identify, showing an impressive improvement in the result edge image.

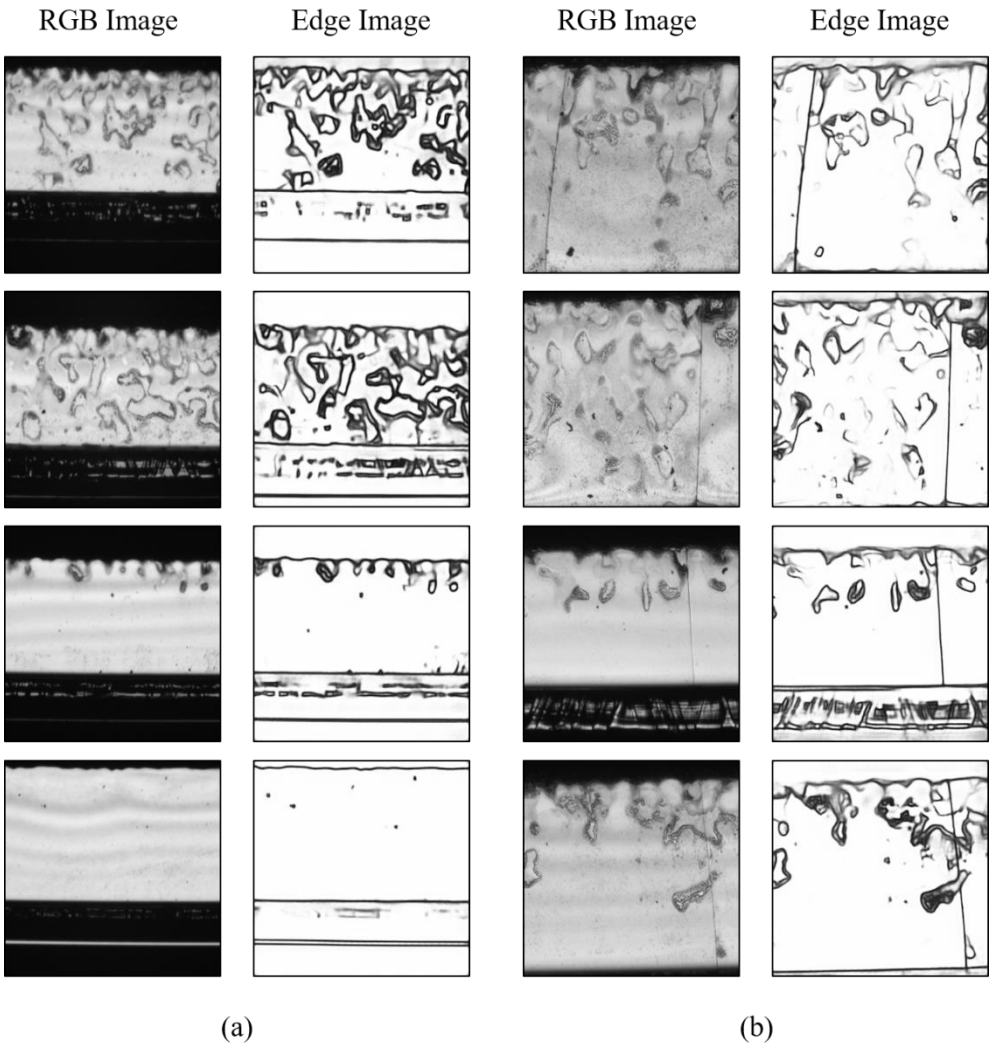


Figure 3 Edge detected wafer surface images (a) the normal RGB images and their corresponding edge images (b) the test RGB wafer images and their corresponding edge images. Both cases contain clean and unclean surfaces. Cracks, which are blurry or covered by the resin bleed in the edge images are highlighted.

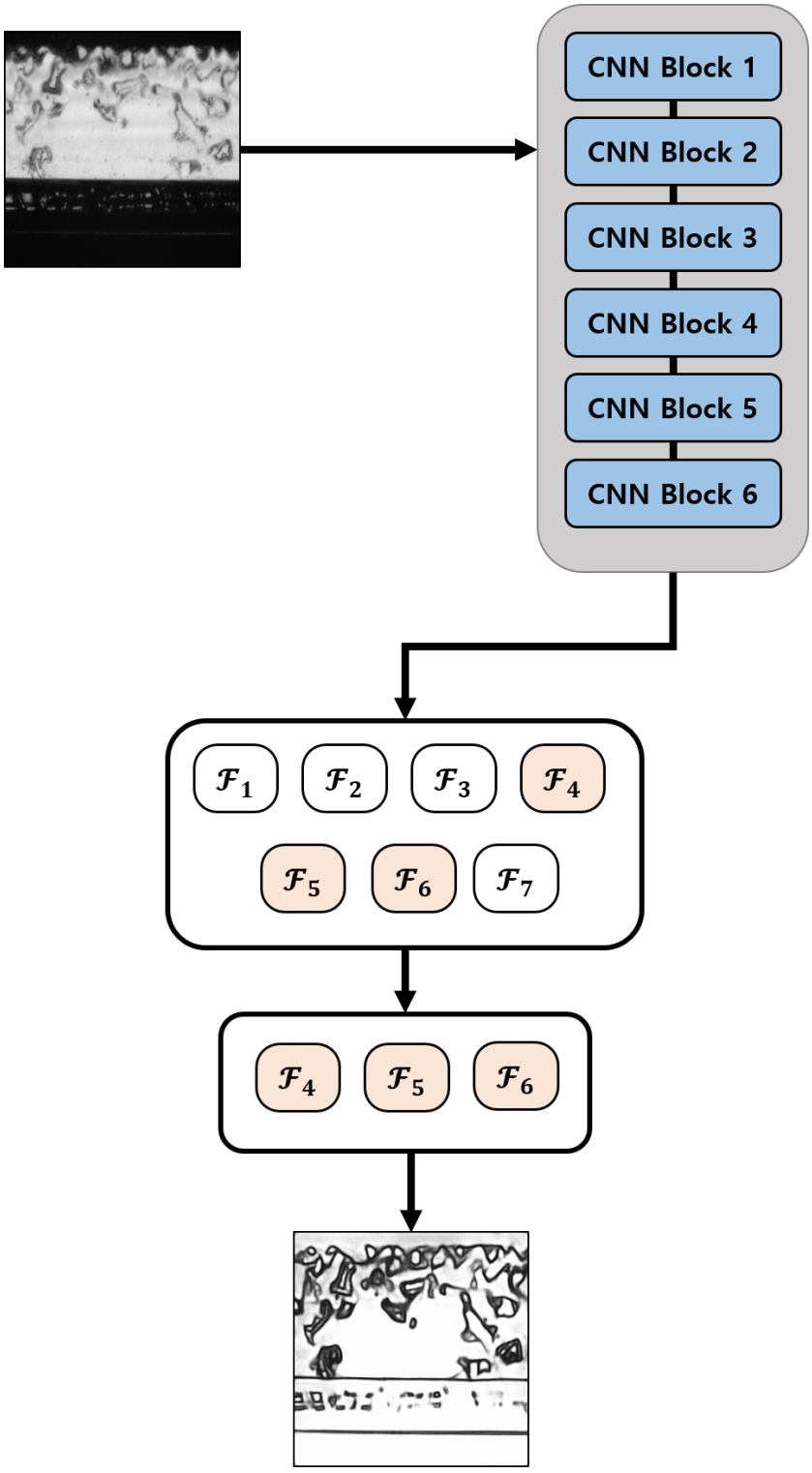


Figure 4 Overview of edge extraction step. Seven feature maps about edges are extracted from the original RGB images. Three feature maps from the fourth to sixth features are used as auxiliary data for edge information enhancement.

2.2. Edge-Enhanced Features into a Memory Bank

In this study, we use a modified PatchCore [36], which is an anomaly detection model, to detect cracks on the wafer surface as anomalies. Pre-trained convolution neural network, WideResNet50 [37], is used as a backbone model for feature extraction. In the feature extraction step, the CNN model is not trained with BackPropagation but outputs multi-resolution feature maps from each convolution neural network block when+ an image is entered into the CNN model. To utilize both the original images and the edge images, two WideResNet50 models are used in parallel. We can obtain the extracted feature maps $\{\mathcal{F}_{i,1}, \mathcal{F}_{i,2}, \mathcal{F}_{i,3}, \mathcal{F}_{i,4}\}$ from each normal image x_i in the normal train dataset $\mathcal{X}_N$, and $\{\mathcal{F}^e_{i,1}, \mathcal{F}^e_{i,2}, \mathcal{F}^e_{i,3}, \mathcal{F}^e_{i,4}\}$ from each edge image x^e_i in the edge train dataset $\mathcal{X}^e_N$. The first subscript i of the feature map means the index of the images, the second subscript, a number from 1 to 4, means the hierarchy level of the convolution blocks in the CNN model and the superscript e means that the feature map comes from the edge image. Among each image's four feature maps, the first and last feature maps are not adequate to use because the features in the first feature map are too general, and the features in the last feature map are too biased to the dataset ImageNet [17], which is the dataset used to train the CNN model. Therefore, $\{\mathcal{F}_{i,2}, \mathcal{F}_{i,3}, \mathcal{F}^e_{i,2}, \mathcal{F}^e_{i,3}\}$ are used for anomaly detection.

Considering that wafer images are not uniform as shown in Figure 1, the features that represent normal data should be able to deal with local spatial variation, since products can't be aligned

accurately every time the image is taken and in our case image dataset are acquired from the high spinning wafer's boundary. As in [36], we locally aggregate the pixel's neighbor information into every pixel in spatial resolution.

$$\mathcal{F}_{i,j}(h,w),\ \ \mathcal{F}^{e}_{i,j}(h,w) = \sum_{w^*=w-1}^{w^*=w+1}\sum_{h^*=h-1}^{h^*=h+1} \mathcal{F}_{i,j}(h^*,w^*),\ \ \mathcal{F}^{e}_{i,j}(h^*,w^*)$$

In the above equation, h and w mean the height and weight of the pixel in the feature map. We perform this process to all feature maps $\{\mathcal{F}_{i,2},\mathcal{F}_{i,3},\mathcal{F}^{e}_{i,2},\mathcal{F}^{e}_{i,3}\}$. The spatial size and the number of channels of each feature map remain constant after the process. To utilize the feature maps $\{\mathcal{F}_{i,3},\mathcal{F}^{e}_{i,3}\}$ in higher resolution, we resize the feature maps from the third hierarchy level of the CNN to the size of the feature maps $\{\mathcal{F}_{i,2},\mathcal{F}^{e}_{i,2}\}$ which come from the second hierarchy level. After resizing, we combine the normal feature maps and edge feature maps that have the same hierarchy level. In the combination process, the feature maps are element-wise summed. Each feature map consists of 28 pixels in width and height, and each pixel consists of 1792 channels. In the view of vector, the element-wise summation of both pixels can be interpreted into vector summation, which is a method conceptually used in [residual network].

$$\mathcal{F}_{i,2} = \mathcal{F}_{i,2} + \mathcal{F}^{e}_{i,2}$$

$$\mathcal{F}_{i,3} = \mathcal{F}_{i,3} + \mathcal{F}^{e}_{i,3}$$

The final feature map for each training image can be obtained by concatenating $\mathcal{F}_{i,2}$ and $\mathcal{F}_{i,3}$ in channel dimension.

$$\mathcal{F}_{i} = concat(\mathcal{F}_{i,2},\mathcal{F}_{i,3})$$

After concatenation, the feature map $\mathcal{F}_i$ is saved into the memory bank $\mathcal{M}$, which is used as feature storage. Processes for the feature extraction are performed until all normal train images' final feature maps are saved into the memory bank.

2.3. Effective Memory Bank Subset Search

The size of the memory bank is too large to use the memory bank as it is to infer anomalies. This computation complexity originated from the resin bleed on the wafer surface. The random shape of the resin bleed on the wafer surface makes it the anomaly detection model hard to deal with the resin bleed and requires more wafer images for anomaly detection. Since we use the k-nearest neighbor technique for searching the proper features from the memory bank, it is impossible to use as many wafer images as possible. To alleviate the computation complexity issue, we apply a core set sub-sampling method to the memory bank. The core set sub-sampling is an optimal method to find a subset that can represent the memory bank well [36]. It is more useful to use the core set sub-sampling method than random sampling in this case because features in the memory bank are not distributed uniformly.

To approximate the memory bank by using coreset sub-sampling, we apply a minimax facility location as in [38, 40] for searching the subset which approximates the memory bank well. Implementing the minimax facility location is a NP-hard problem, so

we solve the problem with iterative greedy approximation [38] and Johnson−Lindenstrauss theorem [39] as used in [36, 40], for reducing dimensionalities. By core set sub−sampling, we can reduce the size of the memory bank to 0.4% of the original size. This process accelerates anomaly detection inference because it avoids referring to all features of the memory bank.

2.4. Algorithm for Anomaly Detection and Localization

Inference of anomaly detection to the test wafer image gives the degree of abnormality to the given image, and if the given image is abnormal, it also gives the degree of abnormality to the regions in the image. The abnormality of the given image is decided by estimating the abnormal score s, which represents the degree of the given image's abnormality, and the abnormality of the region is decided by the abnormal score at each spatial position. To estimate the abnormal score s, we measure the maximum distance s^* between the extracted feature f^{test}, which makes the distance maximum, and the saved feature f, the nearest feature of f^{test}, searched from the memory bank $\mathcal{M}$.

$$f^{test,*}, f^* = \underset{f^{test} \in \mathcal{F}^{test}}{\operatorname{argmax}} \underset{f \in \mathcal{M}}{\operatorname{argmin}} \| f^{test} - f \|_2$$

$$s^* = \| f^{test,*} - f^* \|_2$$

However, there is a possibility that the memory bank contains abnormal features or unique features. This means that the abnormal score s^* may be low because the selected feature f^* cannot represent a normal feature. To prevent the wrong decision, re-weighting is applied to the maximum distance s^*. The re-weight value is determined by considering the nine nearest features of f^{test}, denoted the nearest set as $\mathcal{N}_9(f^*)$.

$$s = \left(1 - \frac{exp\|f^{test,*} - f^*\|_2}{\sum_{f \in \mathcal{N}_9(f^*)} \exp\|f^{test,*} - f^*\|_2}\right) \cdot s^*$$

In the above equation, the re-weight value is close to 1 when the $f^{test,*}$ is far from most of the features in the $\mathcal{N}_9(f^*)$ and it is close to 0 when the $f^{test,*}$ is close to the features in the $\mathcal{N}_9(f^*)$. The re-weight value allows anomaly detection to be performed well, even if the normal dataset contains rare or unique components. Anomaly segmentation is performed by calculating abnormal scores at the local spatial pixels as in Padim [19]. The score map has a 24×24 spatial resolution, and it is resized to the original resolution of 224×224 by bi-linear interpolation. Then the resized score map is smoothed by a Gaussian of kernel width $\sigma = 4$.

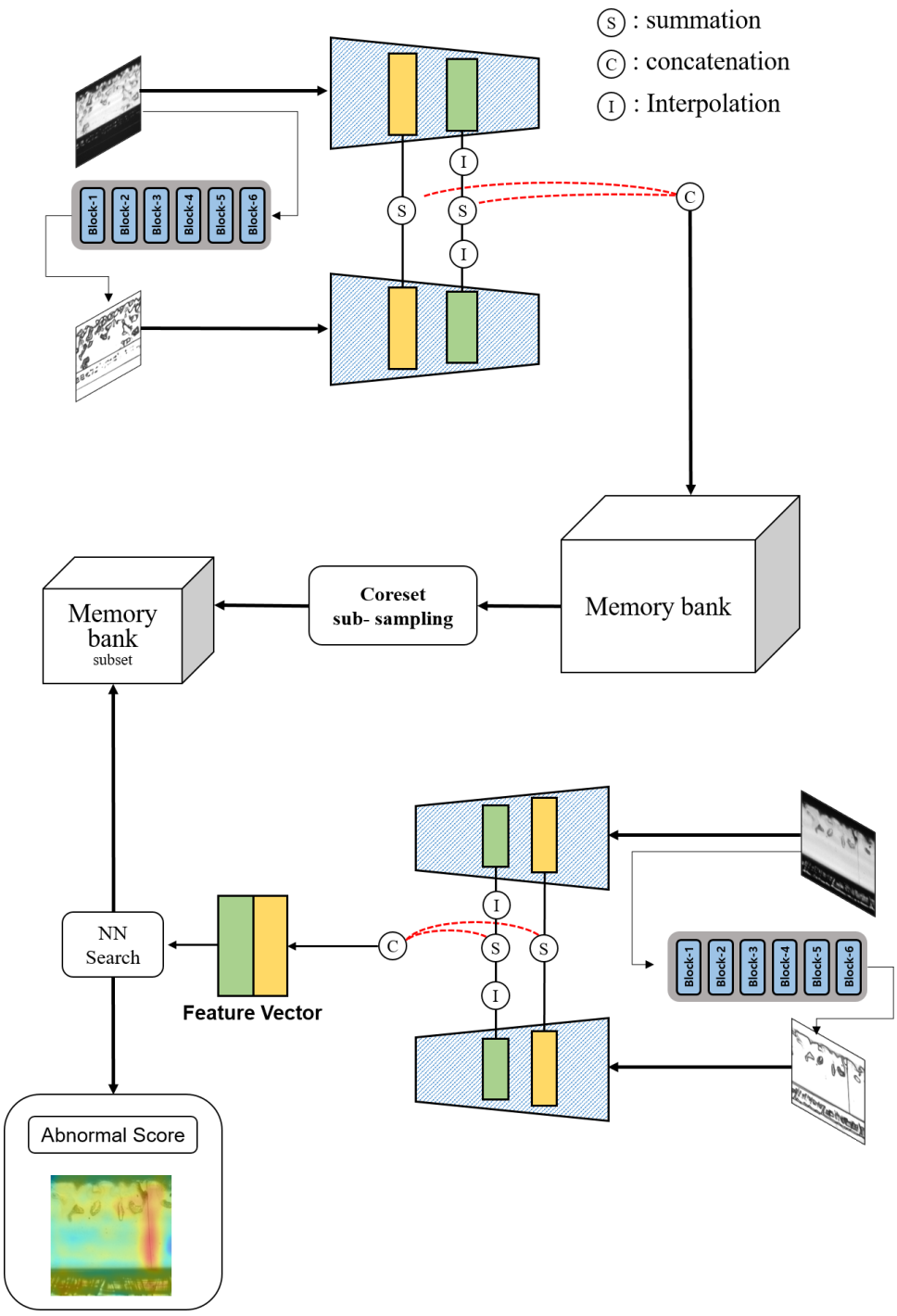


Figure 5 Overview of our proposed anomaly detection method. The overall structure is the same as the anomaly detection model, PatchCore. The PatchCore is modified to be able to accept additional edge images.

Note for Figure 5.

To use edge information of the original RGB image in the inference, the edge detection model, DexiNed, is used in the preprocessing step of the whole process. In the preprocessing step, we choose the fourth, fifth, and sixth features about edge among seven features about edge extracted from the original RGB images. An edge image for each RGB image is acquired by averaging three edge features.

In the training step, two CNN models, pre-trained on the ImageNet, are used in parallel. Four different sizes of feature maps are extracted after the RGB image and edge image are inputted into each CNN model. Two feature maps which are the second and third maps among whole feature maps are used to represent an input image. Thus, we can acquire four feature maps for an RGB image, two for an RGB image, and two for an edge image. Feature maps are combined by element-wise summation and concatenation. The collection of the feature maps from the entire training dataset is called a 'memory bank' that can represent the training image dataset in the matrix form. We can effectively compact the memory bank by searching the optimal subset through coreset subsampling.

In the inference step, one feature map that combined all extracted feature maps from the test image is used to infer the abnormality of the test image. Then we search the most similar matrices from the memory bank by K-NN search. The abnormality of the test image can be scored by measuring the Euclidean distance between the most similar matrices and the feature map representing the test image. In this research, we use the F2 score to localize the cracks adequately, since crack regions occupy a small portion of each test image.

Chapter 3. Model Validation on Wafer Dataset

3.1. Experiments Detail

3.1.1. Datasets and Training Details

Our method for crack detection aims to improve anomaly detection performance on a real wafer surface dataset. The dataset is acquired after a protective cover of the packaged wafer is removed. The wafer is vulnerable to cracks right after the cover is removed, and there is no industrial inspection in this phase. As shown in Figure 1, there are many failures to detect cracks because the cracks on the wafer are covered with resin bleed. The dataset used in our research is composed of 736 normal training images and 353 abnormal images. The training dataset only contains normal class and the test dataset includes 'Crack', 'Normal', and 'Resin bleed cracks'. We made ground truth masks for evaluating the anomaly localization performance. The images are resized to 256×256 pixels and center cropped to 224×224 during the preprocessing phase as in [7, 8, 13]. We set the corset sampling ratio to 0.4% in the training phase, considering the size of the training dataset. We used a single NVIDIA GeForce RTX 3090 for evaluating performance.

3.1.2. Evaluation Metrics for Anomaly Detection

To evaluate the anomaly detection performance, we use the area under the receiver-operator curve (AUROC) at both the image-level and pixel-level. The receiver-operator curve (ROC) is a curve representing the performance of the binary classifier for various thresholds. The curve is plotted with the true positive rate (TPR) and the false positive rate (FPR). Thus, a perfect binary classifier achieves 1.0 in the AUROC score.

3.2. Anomaly Detection on Wafer Surface

The image-level and pixel-level anomaly detection performance of our method are shown in Table 1. Most of the existing methods are unable to perform anomaly detection on the wafer datasets because the aforementioned problems make it difficult to distinguish extracted features. The PatchCore [8], used as our method's backbone model, shows the highest performance on both our wafer dataset and the benchmark dataset MVTec [18]. However, the PatchCore model can't localize the regions of cracks as shown in Figure 5. Cracks that were not clearly detected are usually covered by resin bleed or blurry due to out-focusing. Our method, which employs the edge detection model, DexiNed, improves anomaly detection performance on images with resin bleed and blurriness. For the wafer dataset, our method achieves 96.7% image-level AUROC

and 98.6% pixel-level AUROC, which resulted in 4.5% and 2.0% performance improvements, respectively. The performance improvements are achieved without any additional annotations.

Model	Image AUROC	Pixel AUROC
SPADE	51.1%	52.3%
PaDiM	59.3%	90.0%
PatchCore (Res18)	85.1%	95.5%
PatchCore (WRN50)	92.2%	96.6%
Ours (Res18)	92.8%	97.7%
Ours (WRN50)	**96.7%**	**98.6%**

Table 1 The result of the anomaly detection on the wafer dataset. The approaches of SPADE and PaDiM are similar to the approaches of PatchCore. PatchCore is better able to deal with the wafer dataset than the other existing method, but it still shows poor performance. Our method shows improvements in both Image-level and Pixel-level scores.

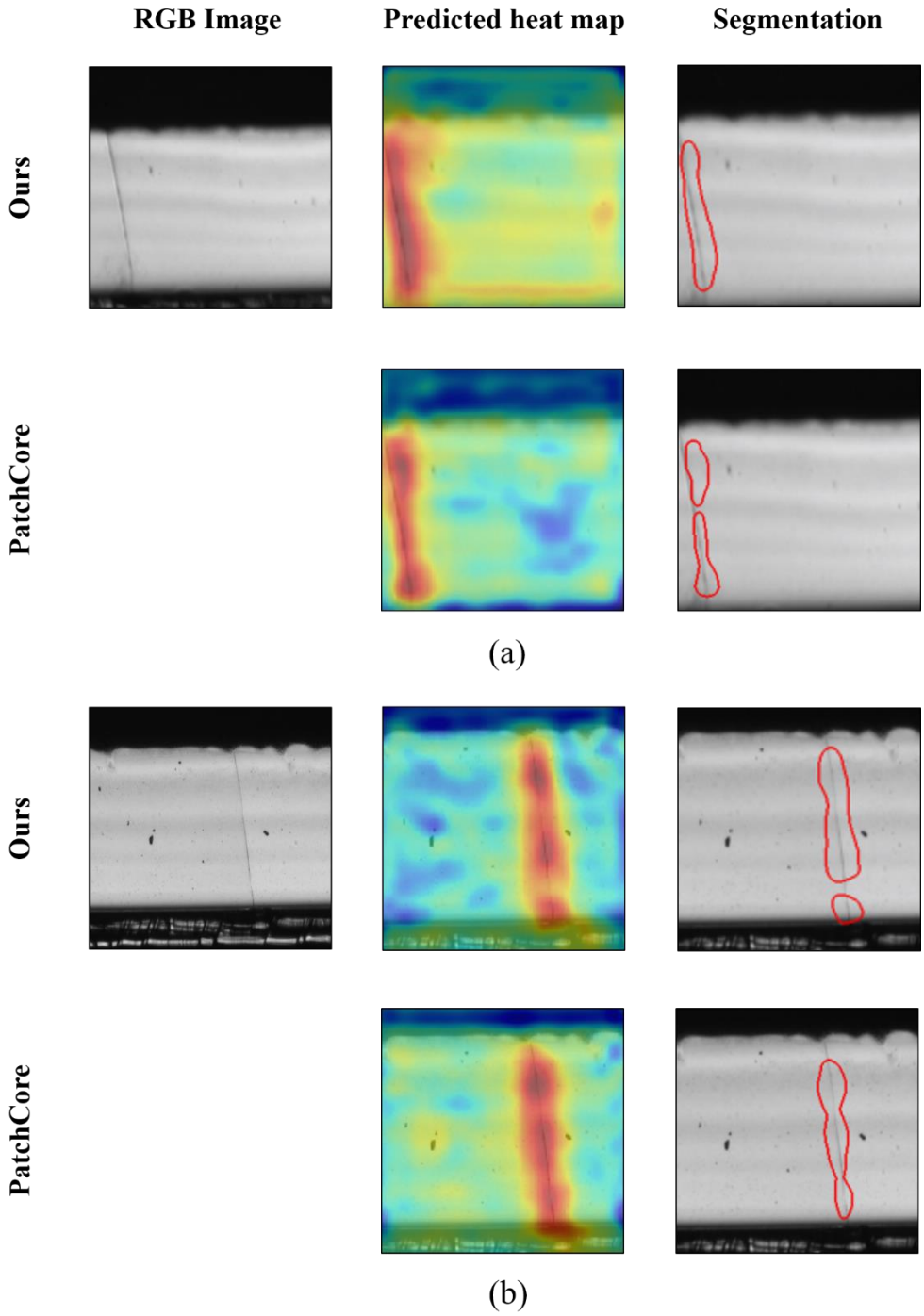


Figure 6 First results of visualized results on the anomaly detection model. Both wafer images have a crack on the clear background. The existing model and our modified model both show good detection performance.

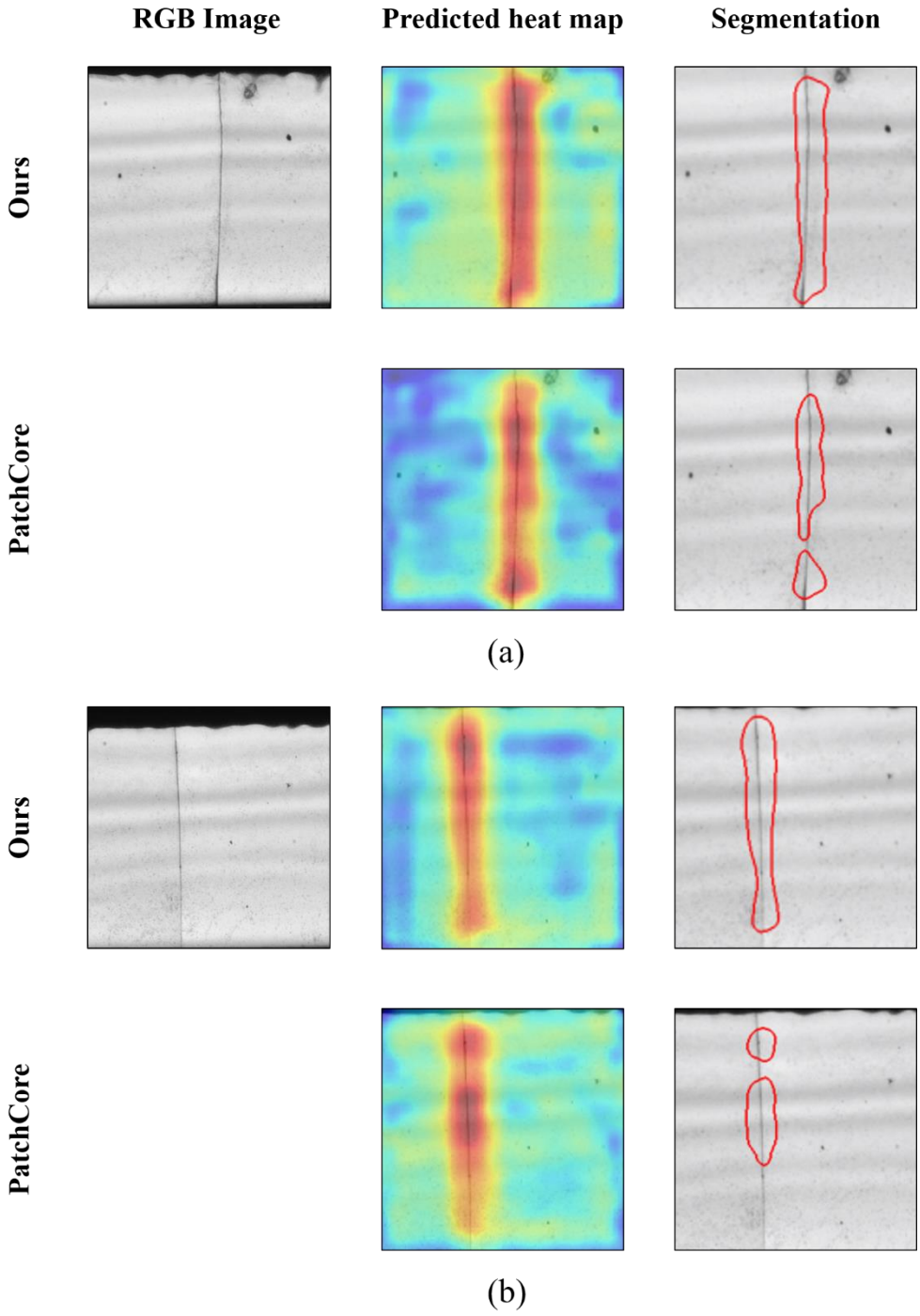


Figure 7 Second results of visualized results on the anomaly detection model. Though both wafer images have a crack on a clear background, the existing model cannot detect the lower part of the crack. The difference originates from the blurriness and wavy pattern of the images. (a) crack is well segmented at both ours and PatchCore. (b) Our model detects the crack well, but PatchCore misses the portion of the crack.

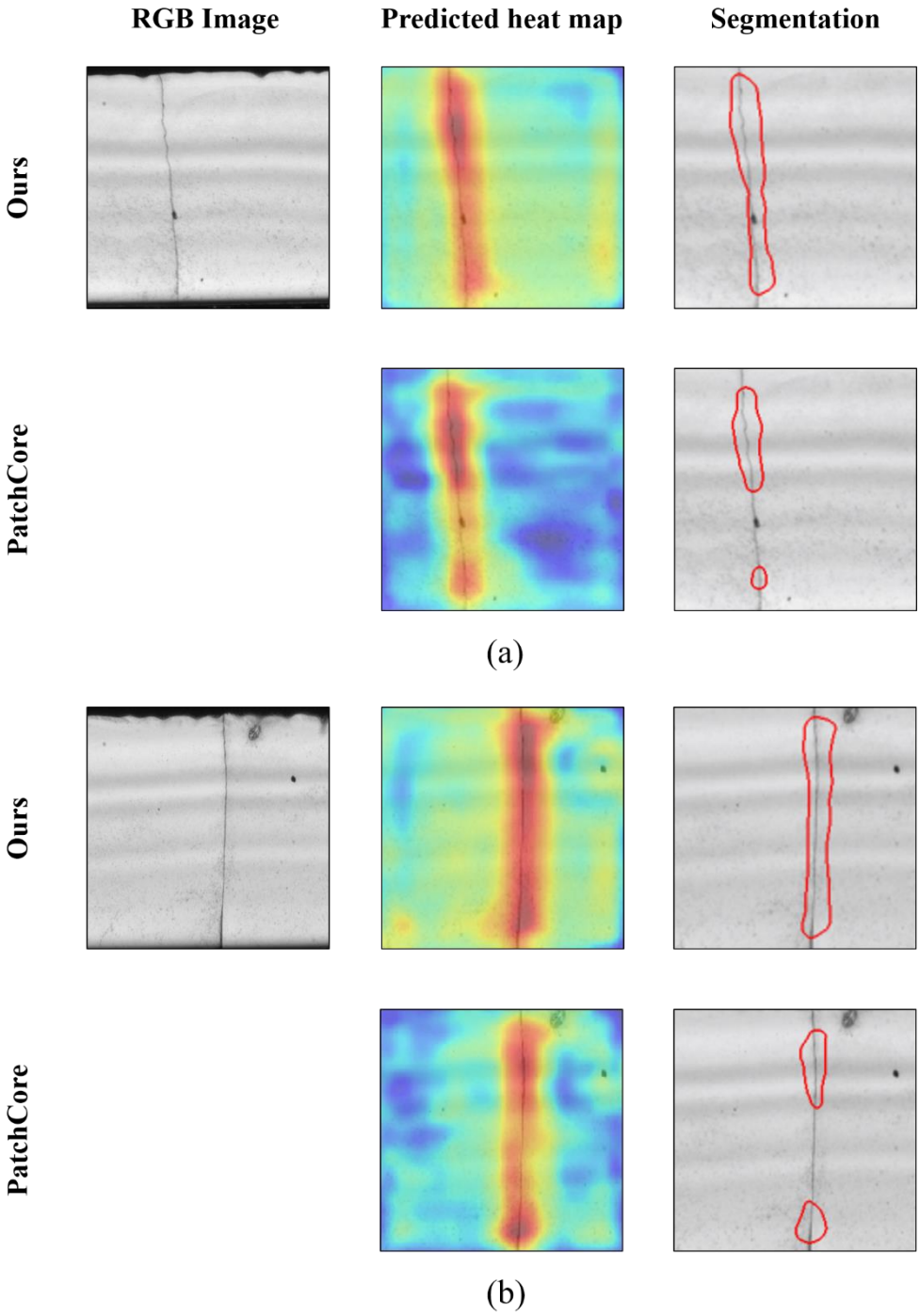


Figure 8 Third results of visualized results on the anomaly detection model. The existing model does not give high abnormal scores to the middle part of a crack on both images. These results show that the PatchCore is not trained to be sensitive to the edges.

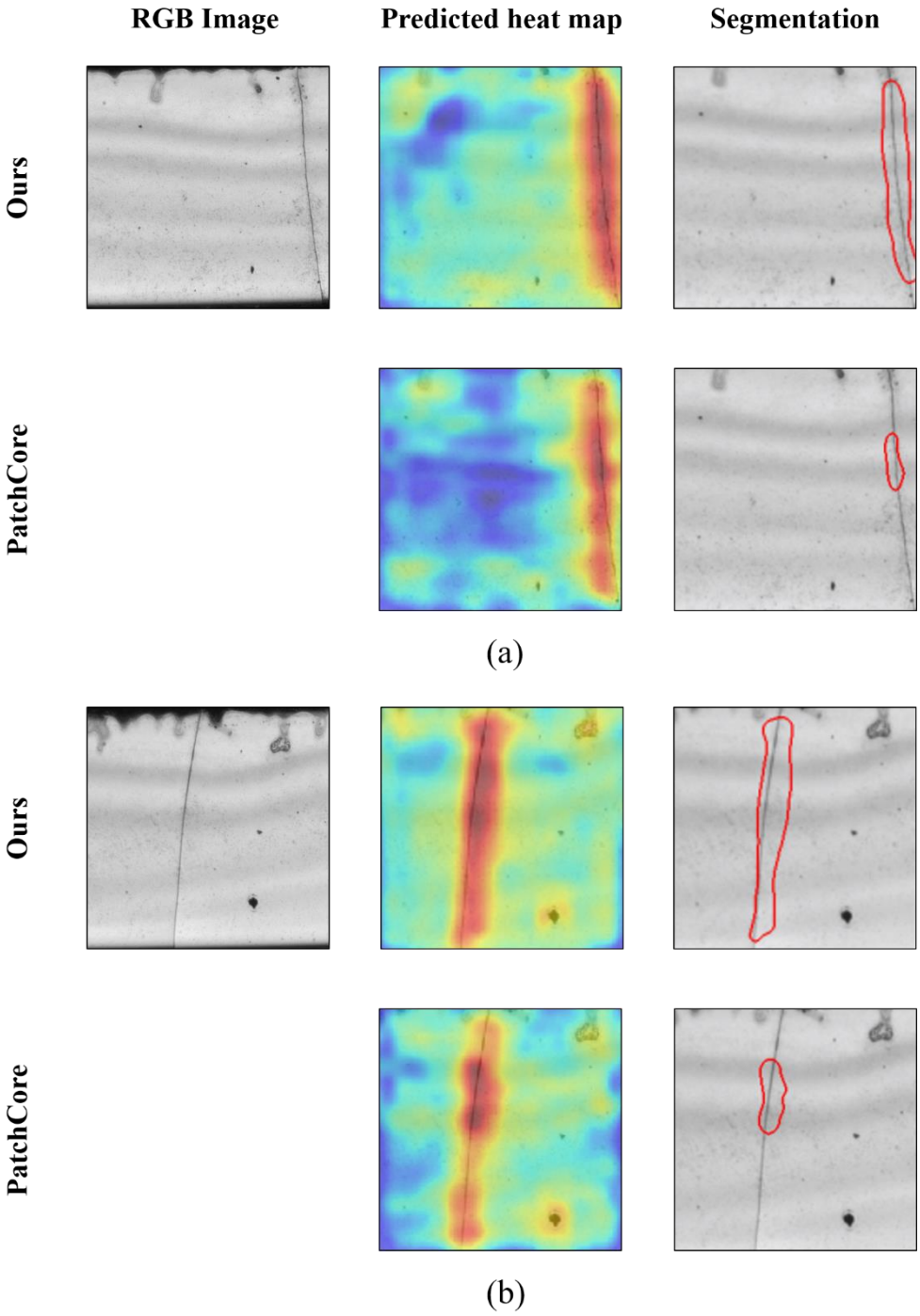


Figure 9 Fourth results of visualized results on the anomaly detection model. These examples are the case of not having good performance despite cracks on a clean background. Adding edge information shows dramatic performance improvements in the cases.

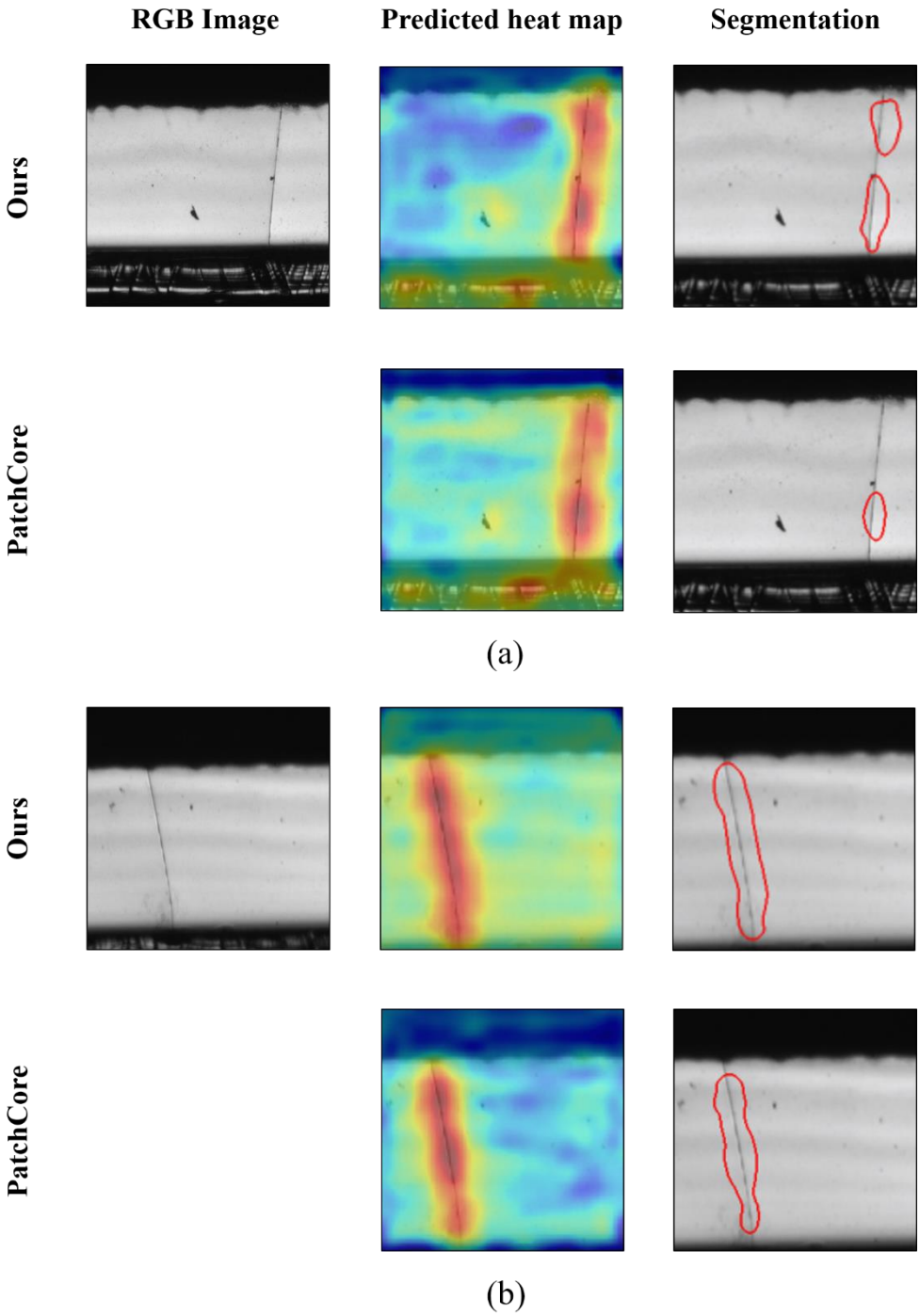


Figure 10 Fifth results of visualized results on the anomaly detection model. In Figure 10 (a), the model gives relatively lower values to the crack, since the crack is similar to the resin bleed in the training dataset.

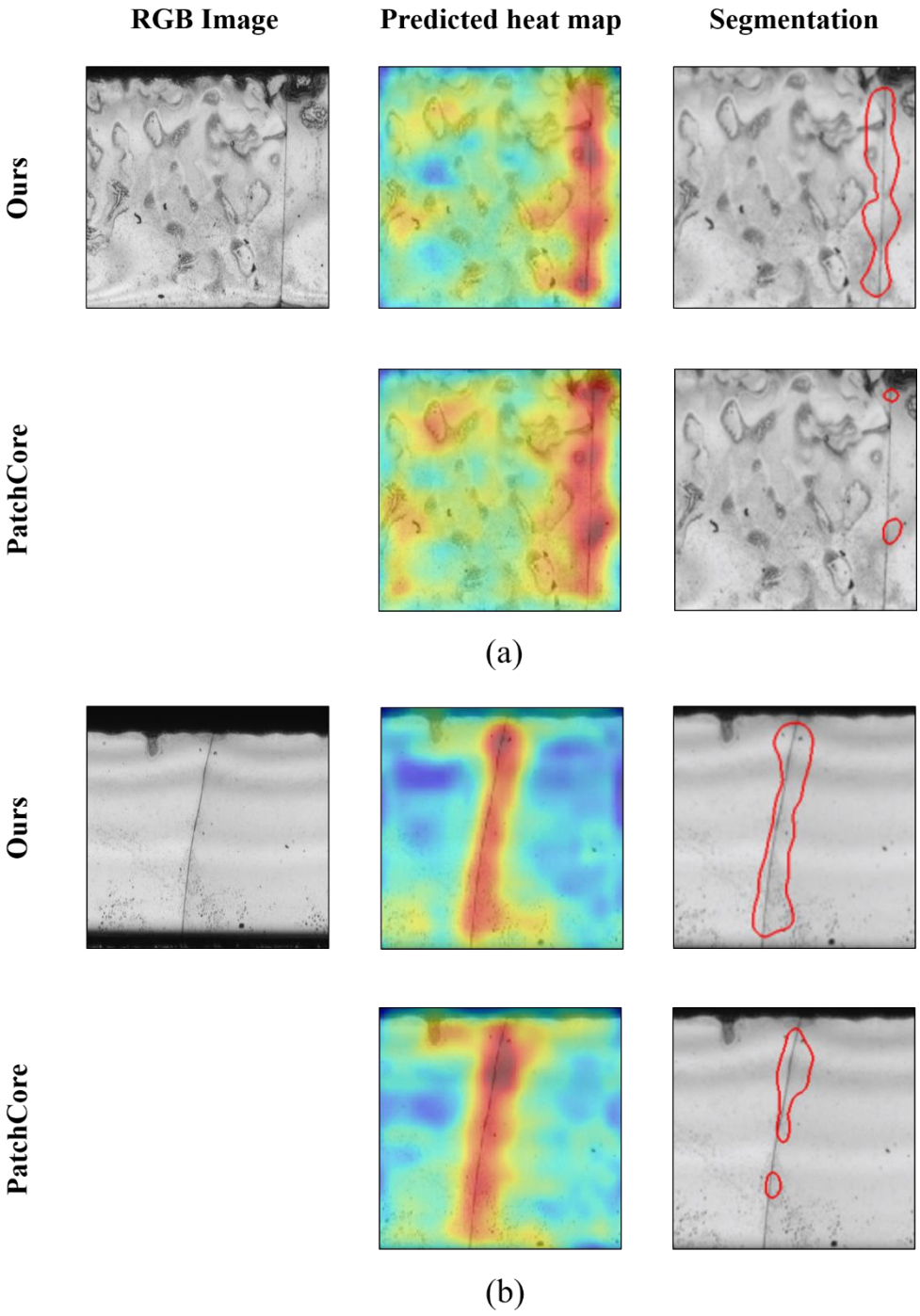


Figure 11 Sixth results of visualized results on the anomaly detection model. When the resin bleed or black dots are around the cracks, the anomaly detection model hardly detects the cracks.

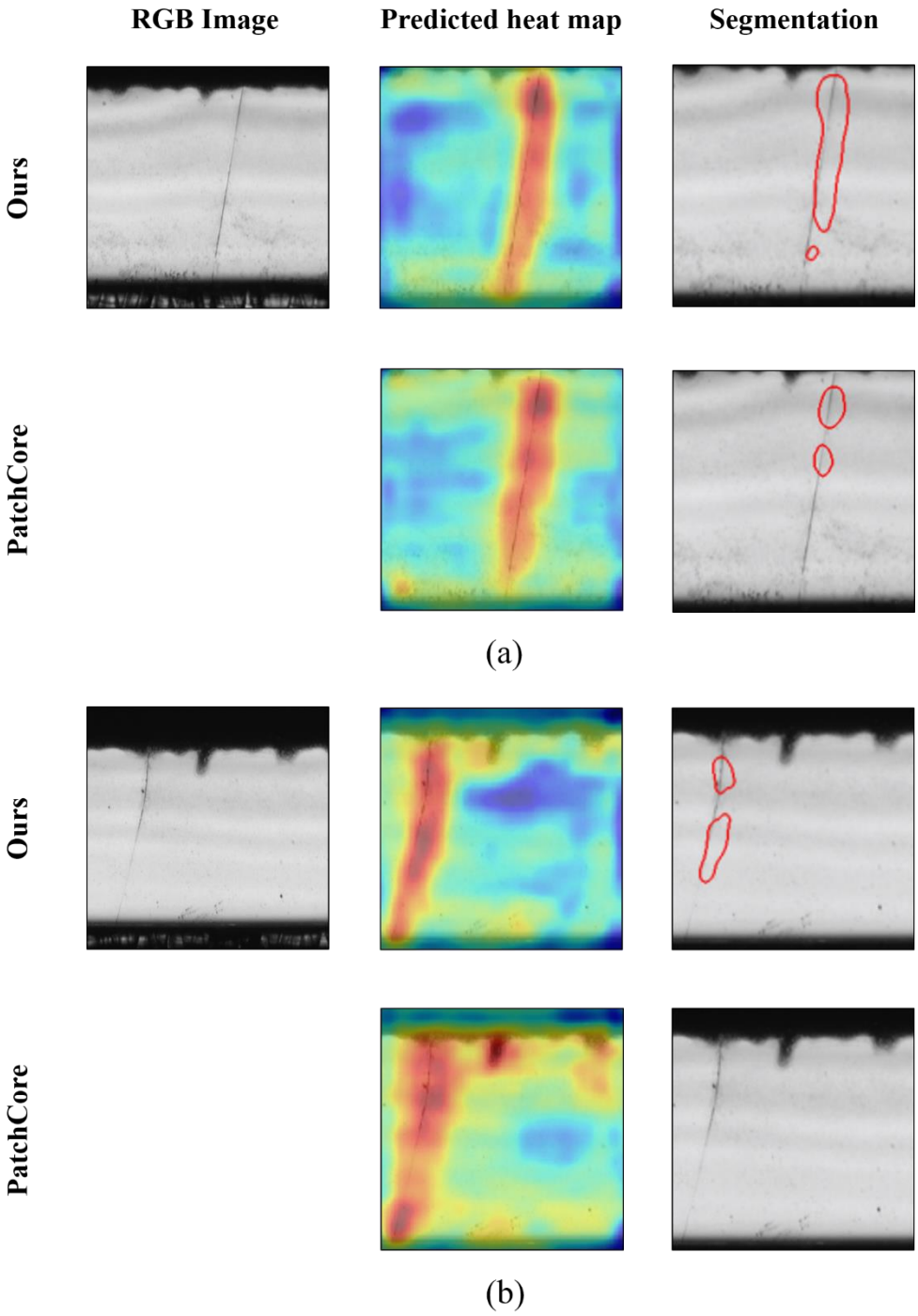


Figure 12 Seventh results of visualized results on the anomaly detection model. Not only the resin bleed or noises but also the blurriness of the cracks make the anomaly detection model hardly detect cracks on the wafers.

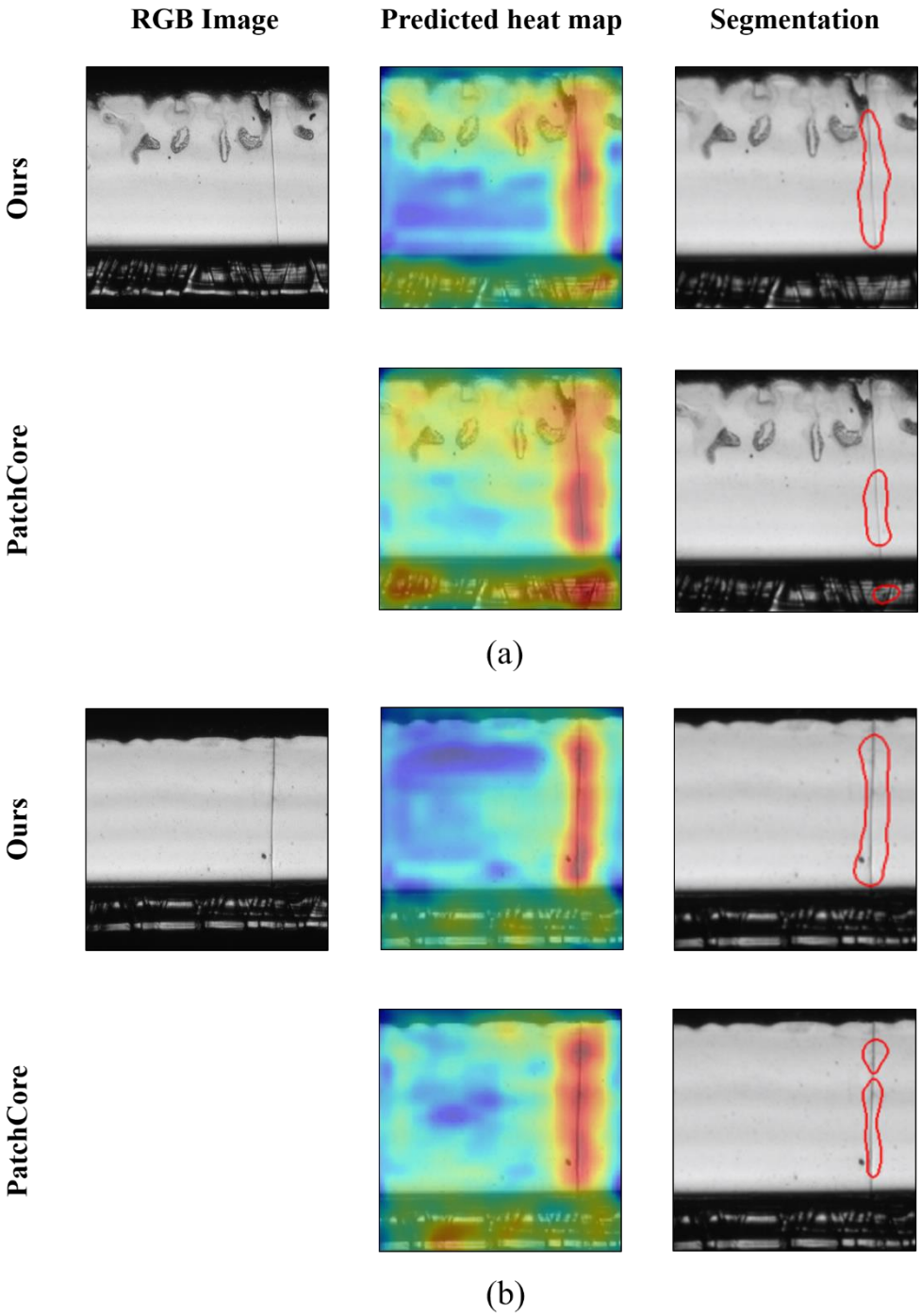


Figure 13 Eighth results of visualized results on the anomaly detection model. The upper image has the resin bleed layer at the boundary compared to the lower image. The existing model cannot detect the crack at the resin bleed layer.

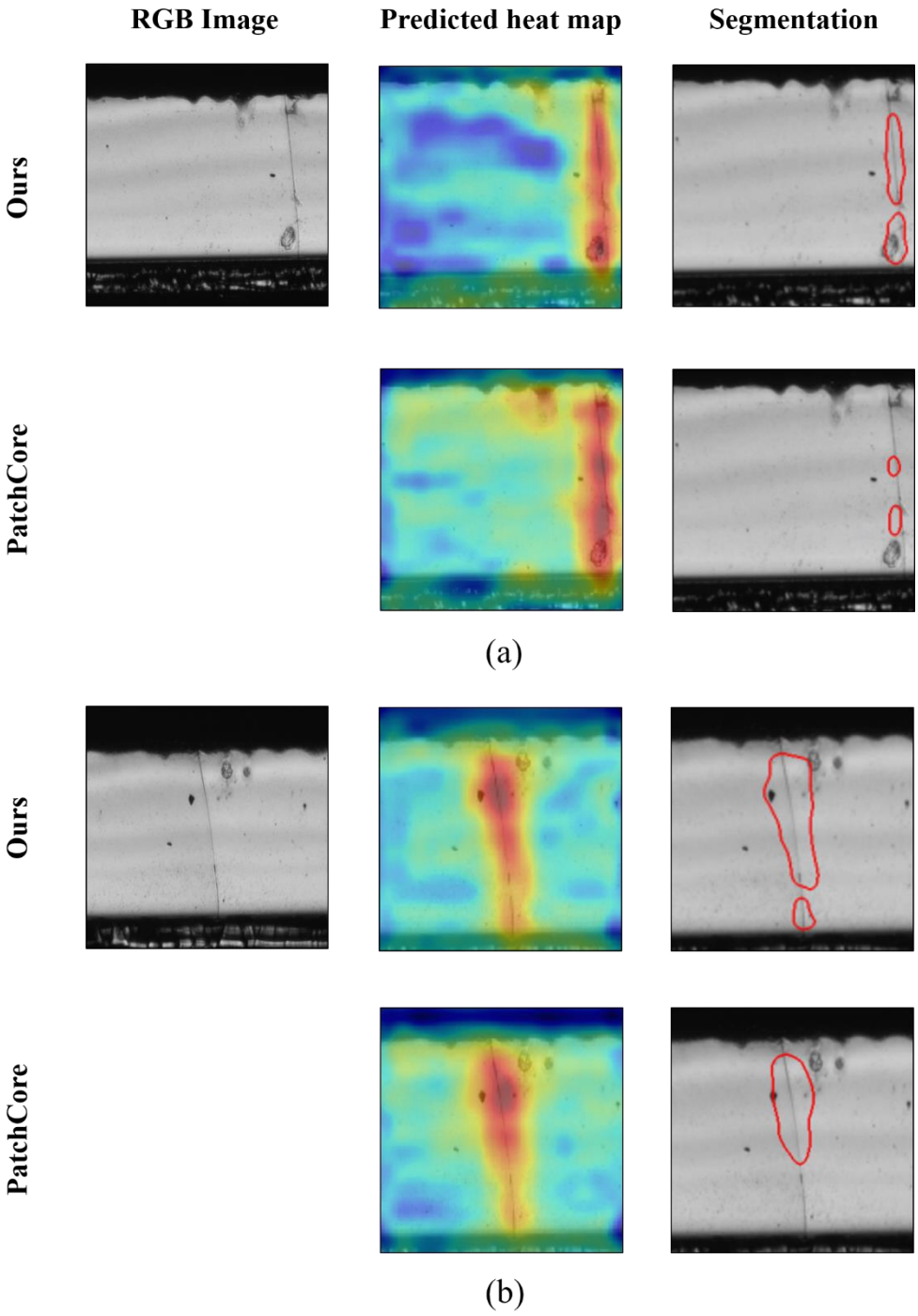


Figure 14 Ninth results of visualized results on the anomaly detection model. Our method shows general performance improvements on both images. These performance improvements are because the addition of edge information makes the score boundary clear.

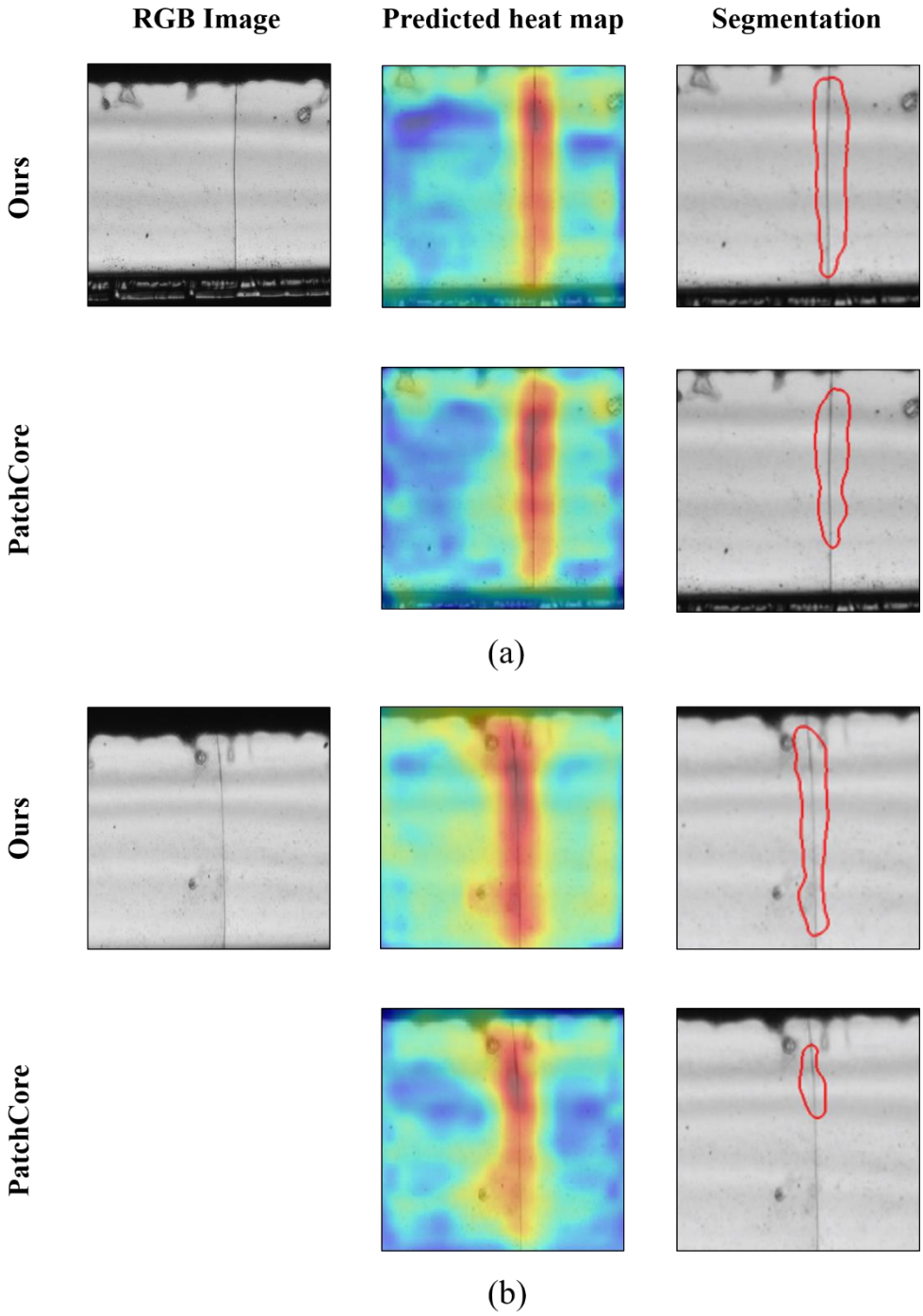


Figure 15 Tenth results of visualized results on the anomaly detection model. Even though both models give relatively high values to the region of the cracks as shown in heat maps, our model gives the more adequate level of abnormal scores.

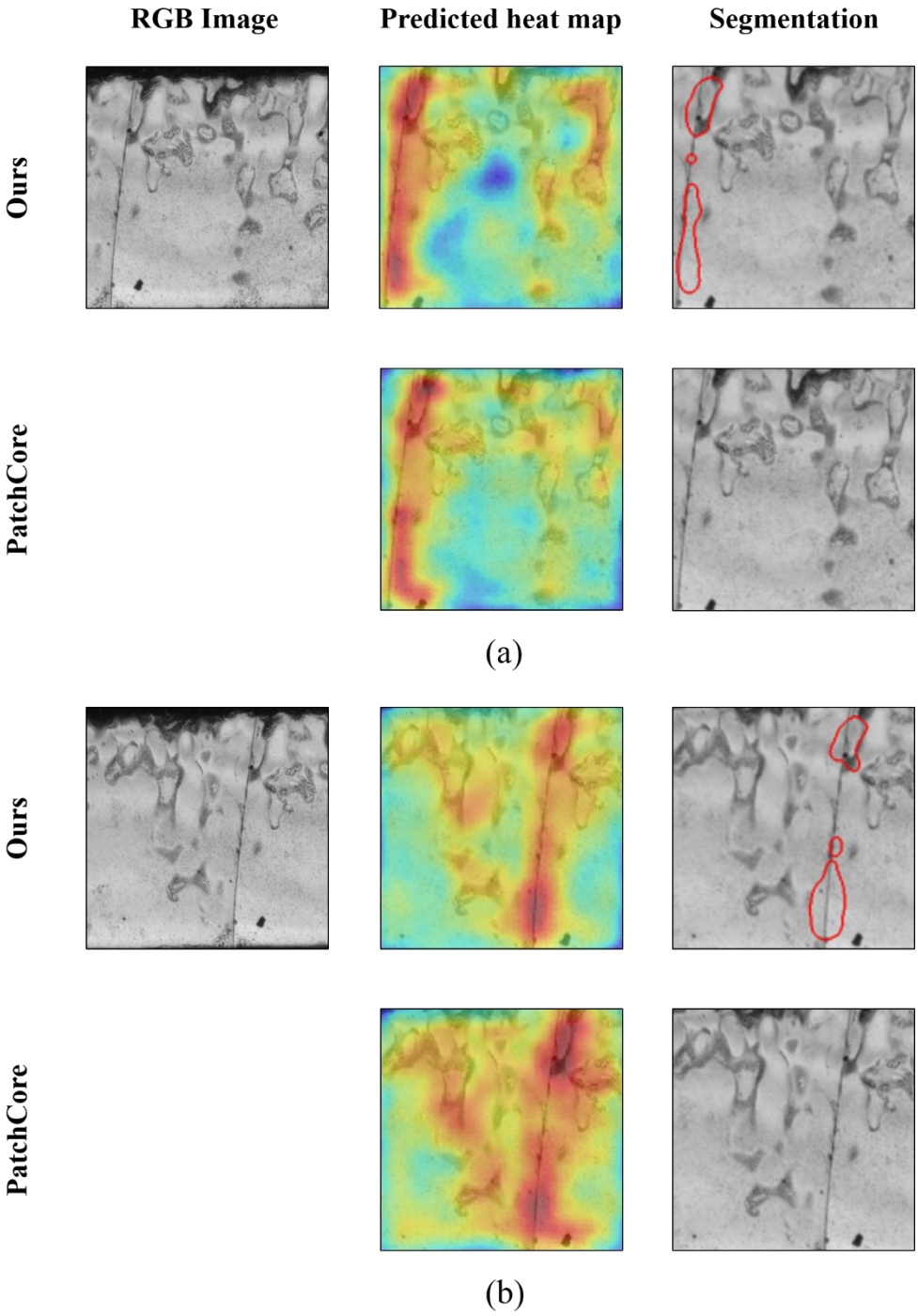


Figure 16 Eleventh results of visualized results on the anomaly detection model. Our modified anomaly detection model detects cracks from the images that the existing model fails to detect cracks. The existing model gives not enough high abnormal scores to the region of the cracks.

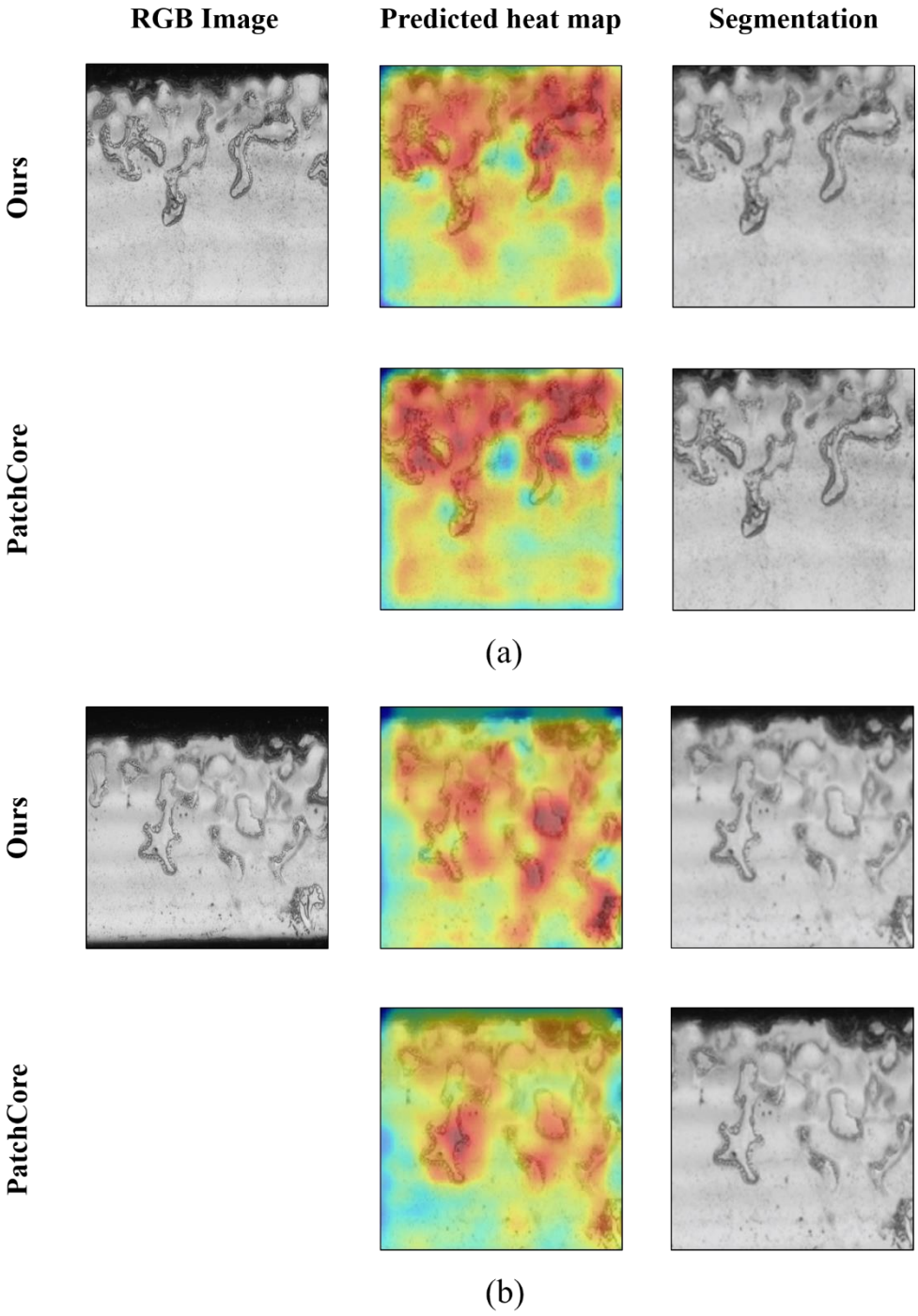


Figure 17 Twelfth results of visualized results on the anomaly detection model. Both models give relatively high abnormal scores to the region of resin bleed than the clear background. But they have the ability to classify the region of resin bleed as the normal part.

3.3. Result Analysis

Our method improves the anomaly detection model, PatchCore to be able to detect cracks on the wafers' unclean background. We proposed many conditions that affect the crack detection performance of PatchCore, the anomaly detection model, from Figure 6 to Figure 17. The modified version of the anomaly detection model becomes more sensitive to the edges so that the model can detect cracks better than its original version. The modified version has a 2.0% higher pixel-wise crack localization score. Our model shows that it can detect the entire crack area, which the existing model cannot detect, in Figure 16 (a) and (b).

The existing model, PatchCore already can detect cracks on a clear wafer surface as shown in Figure 6. The anomaly detection model can easily detect cracks that are not blurry and has clear backgrounds. However, the PatchCore is very vulnerable to the noises like black dots as figured in Figure 7 (b). In Figure 7 (b), the PatchCore gives not enough abnormal scores to the lower portion of the crack. Cracks, with wavy patterns on the clear surface or black dots, are close to the normal components from the perspective of the PatchCore. In Figure 8 (a), the middle part of the crack is classified as normal. The result means that the crack region with a big black dot and the wavy patterns is in the normal data distribution to the PatchCore. Also, Figure 8 (b) belongs to the case that the noises like black dots and wavy patterns confuse the model to regard the crack as normal. The PatchCore cannot detect cracks in Figure 9 for the

same reason. In the numerical view, these cases can be interpreted as that crack has a weak edge information intensity. Even though these cases are not dealing with blurry cracks, the anomaly detection model needs to regard the cracks as the stronger intensity. Our method solves this issue by summing the edge information extracted from the deep-learning based edge extraction model. The information about the black dots and wavy patterns appears once in the original backbone model, however, the information about the cracks appears both in the original backbone and modified backbone, thus the crack's edge information can be enhanced. Our modified PatchCore can detect the entire region of cracks as shown in Figures 7, 8, and 9. Noises are not the only reason why the existing anomaly detection model cannot detect cracks from clear backgrounds. Blurry cracks are also the components, which the PatchCore cannot detect. Figure 12 (b) is a clear example of the vulnerability of blurriness. The crack in Figure 12 (b) is not disturbed by noises and already has a weak intensity. Because many anomaly detection models are based on the patch-based model that analyzes the test input in the patch unit, a long crack isn't compared by itself. All the cracks, which are long in most cases, are compared after being split into small patch units. The efficient strategy to the view of the memory computation is a weakness to the properties of the cracks. It will be difficult to determine if it is a defect by looking at only a small piece of the blurred crack. Our method highlights the cracks so that they can have strong value. The modified model can detect blurry cracks in Figure 14 too. Both noises and blurriness cases can be explained in the

vector view. The PatchCore and our modified version model learn the normal data distribution in the vector form. They infer the abnormality by measuring the distance between the vector that is explaining the test input region and its most similar vector from the normal image dataset. In the case of noise, our method to enhance the edge information of cracks is used to distinguish the cracks from the noises. In other words, the vector of cracks with noises points in the wrong direction compared with the direction the vector of the cracks without noises points in. In the blurriness case, it is similar to the case of the noise but the difference is that the vector of the crack points in the right direction. Our method in the PatchCore has the same role in both cases, however, they acted differently in the view of the vector.

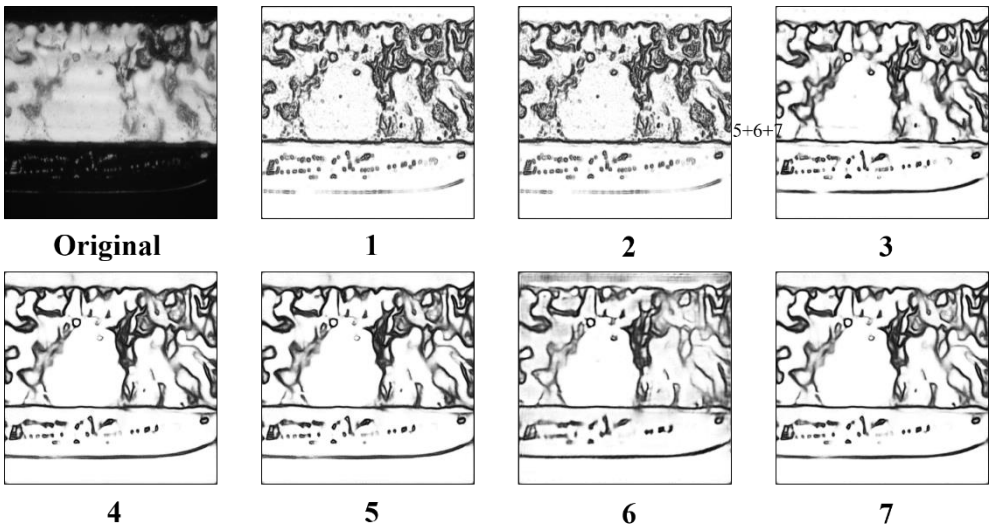
The dramatic performance improvements are shown in Figure 11 (a), Figure 16 (a), and Figure 16 (b). Those figures are filled with the resin bleed, and the PatchCore cannot detect entire cracks in Figure 16 (a) and (b). The PatchCore and our modified model both use a 24×24 feature map to infer the 224×224-size image. It means that one pixel in the feature map represents 9~10 pixels in the input image. Since the cracks in the wafer dataset usually have a thin-width and long-height, one pixel in the score map can contain information on not only cracks but also the resin bleed in the test images. In some cases, the resin bleed regions occupy more region than the crack region in 1×1 pixel of the score map, which is 81~100 pixels in the input image. Figure 11 (a), Figure 16 (a), and (b), are the examples that correspond to the cases. In this case, it is hard to

distinguish the patch of the wafer's resin bleed region in the normal dataset from the patch of the wafer's crack region with the resin bleed. In the figures, our model shows dramatic improvements in anomaly segmentation under the F2-score threshold. Our model has improvements on the cracks when we refer the Figure 17 (a) and (b). Results on images show that our model does not classify the resin bleed as abnormal but does localize more crack regions compared to the existing model. And the results prove that our proposed method augments the edge information that is important to detect cracks.

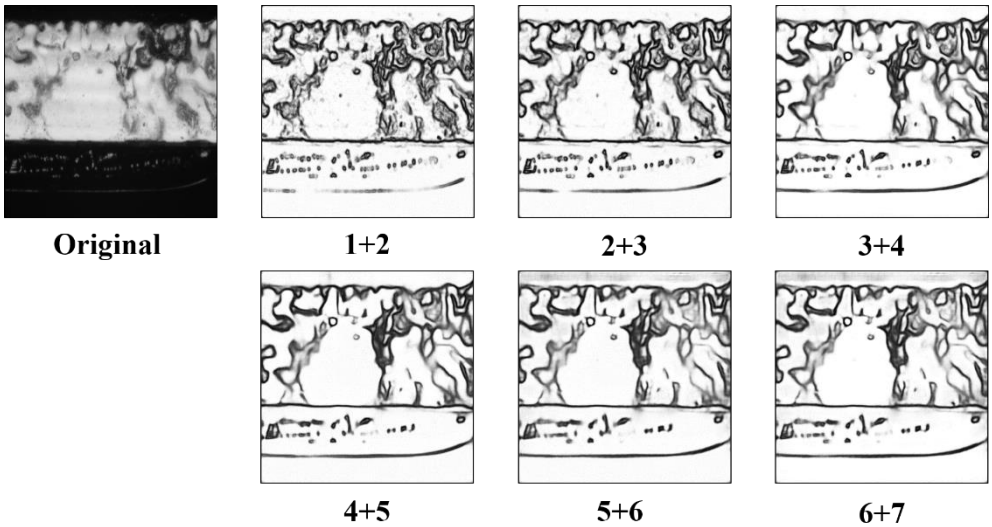
3.4. Comparison Study for Selecting Edge Features

We experimented to evaluate the impact of edge information on anomaly detection performance. The DexiNed [35], an edge detection model, outputs different edge features depending on depth, and an edge image is created by averaging seven edge features. Our method to utilize the edge information to detect anomalies aims to make the anomaly detection model focus on the crack's edge. At the same time, our method aims to make the anomaly detection model distract from the resin bleed, which confuses the model. To maximize the impact of auxiliary edge data, we verify the impact of each combination of edge features on anomaly detection performance.

Among all edge features, we select those with continuous depth levels ranging from one to seven. We choose edge features, which have continuous depth levels from one to seven among all edge features. The more we select edge features from the shallow depth level, the coarser the extracted edge images become, as in Figure 18. Using the auxiliary edge image improves performance in all cases, but each case has a different degree of improvement. The improved results are shown in Table 2. The performance results are an average of ten trial tests for each case. Because the problem in wafer anomaly detection is in the score map, which has a small score difference between the crack and the normal area, small performance improvement on pixel AUROC shows significant improvements for detecting cracks fully covered with the resin bleed.

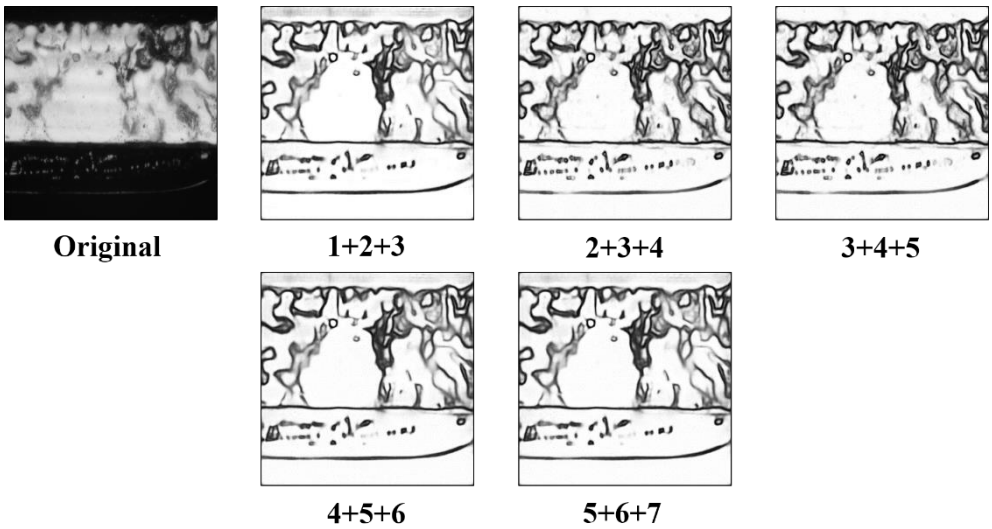


(a)

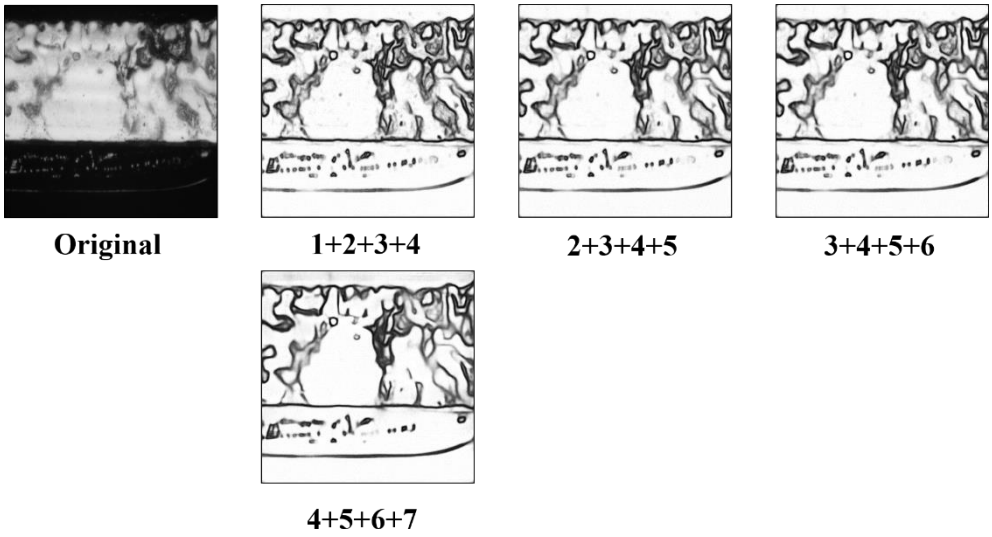


(b)

Figure 18 (a) Seven examples from the corresponding level of the edge features. The edge images are coarse at the shallow level and soft at the deep level. Both properties have pros and cons in the image abnormal decision and pixel abnormal localization. (b) Seven examples from the combination of two near features from the edge features. Overall, all images are similar, but they are different in detail.

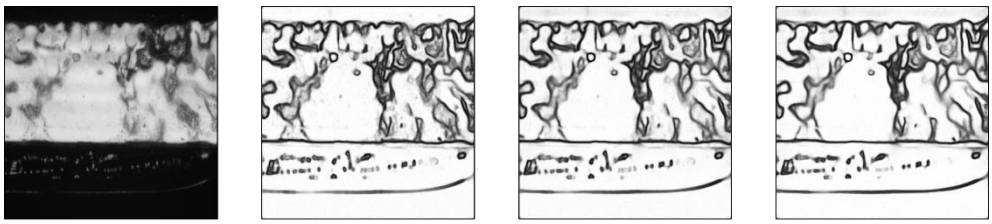


(c)



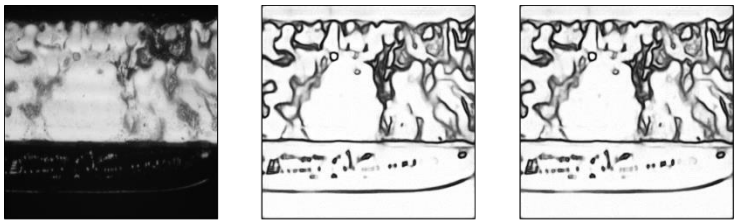
(d)

Figure 19 The examples of edge images that three and four features are mixed. In the case that the fourth, fifth, and sixth features are mixed, the model shows the maximum crack detection performance. There is the biggest difference in the ability to describe the interior part of the resin bleed.



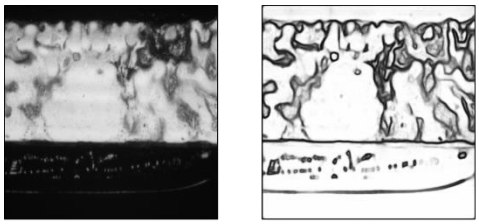
Original **1+2+3+4+5** **2+3+4+5+6** **3+4+5+6+7**

(e)



Original **1+2+3+4+5+6** **2+3+4+5+6+7**

(f)



Original **1+2+3+4+5+6+7**

(g)

Figure 20 The examples of the combinations of four, five, and six edge features. They show a small difference in images and their corresponding detection performances. Their detection performance at the pixel level is almost the same, showing a difference of about 0.2%.

Selected features	Average Image-AUROC	Average Pixel-AUROC
1	95.9%	97.7%
2	95.4%	97.7%
3	93.5%	97.9%
4	93.9%	98.0%
5	95.6%	98.4%
6	95.1%	98.4%
7	93.5%	98.0%
1+2	95.8%	97.8%
2+3	94.6%	97.8%
3+4	93.7%	98.0%
4+5	94.9%	98.3%
5+6	95.3%	98.3%
6+7	94.8%	98.2%
1+2+3	94.7%	97.8%
2+3+4	94.8%	98.0%
3+4+5	94.9%	98.2%
4+5+6	**95.8%**	**98.4%**
5+6+7	95.1%	98.3%
1+2+3+4	95.4%	97.9%
2+3+4+5	95.4%	98.0%
3+4+5+6	95.0%	98.3%
4+5+6+7	94.7%	98.4%
1+2+3+4+5	95.9%	98.0%
2+3+4+5+6	95.6%	98.1%
3+4+5+6+7	94.8%	98.2%
1+2+3+4+5+6	96.1%	98.1%
2+3+4+5+6+7	95.8%	98.1%
1+2+3+4+5+6+7	95.9%	98.0%

Table 2 Results of anomaly detection on every selected edge features case.

According to the results, feature selection affects the additional improvement of anomaly detection performance. In the case of which fourth, fifth, and sixth features in auxiliary edge image acquisition, the model has the highest anomaly detection performance. The anomaly detection model performs worse when using the middle part of the features. Using the second half of the edge features shows higher performance in pixel-level AUROC, but using the last features is harmful to the performance in both image and pixel scores. The rest of the results let us know in which cases we can achieve the highest anomaly detection performance.

Chapter 4. Conclusions

In this paper, we propose an anomaly detection method to improve anomaly detection performance on the wafer dataset, including a 'resin bleed' that hinders detecting cracks. Our method improves performances without any additional datasets and annotations. Our method extracts 7 features of the edge using the edge detection model and selects some of the outputs based on their impact on anomaly detection performance. Then we propose a method to combine the features from the original RGB images and the features from the edge image. As a result, our method has a great ability to detect cracks with unclean backgrounds while maintaining good performance for clear cracks with clean backgrounds. On the wafer surface dataset, our approach achieves 96.7% in image-level AUROC and 98.6% in pixel-level AUROC. No additional data and annotations are used for performance improvements.

Our method has several contributions: (1) Our method can be applied to the real wafer inspector (2) Our approach proposes a better method to approximate cracks so that the cracks can be more clearly distinguished. (3) Our method can be applied to other real datasets that have cracks and components that hinder detecting cracks.

Bibliography

[1] Bergmann, P., et al. (2019). MVTec AD--A comprehensive real-world dataset for unsupervised anomaly detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

[2] Ali, R., et al. (2022). "Structural crack detection using deep convolutional neural networks." Automation in Construction 133: 103989.

[3] Lowe, D. G. (2004). "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60(2): 91-110.

[4] Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Ieee.

[5] Schölkopf, B., et al. (2001). "Estimating the support of a high-dimensional distribution." Neural computation 13(7): 1443-1471.

[6] Ruff, L., et al. (2018). Deep one-class classification. International conference on machine learning, PMLR.

[7] Yang, J., et al. (2021). "Visual anomaly detection for images: A survey." arXiv preprint arXiv:2109.13157.

[8] Kingma, D. P. and M. Welling (2013). "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114.

[9] Goodfellow, I., et al. (2020). "Generative adversarial networks." Communications of the ACM 63(11): 139-144.

[10] Bergmann, P., et al. (2018). "Improving unsupervised defect segmentation by applying structural similarity to autoencoders." arXiv preprint arXiv:1807.02011.

[11] Sakurada, M. and T. Yairi (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis.

[12] Wang, Z., et al. (2004). "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13(4): 600-612.

[13] Hadsell, R., et al. (2006). Dimensionality reduction by learning an invariant mapping. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), IEEE.

[14] Schlegl, T., et al. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. International conference on information processing in medical imaging, Springer.

[15] Akcay, S., et al. (2018). Ganomaly: Semi-supervised anomaly detection via adversarial training. Asian conference on computer vision, Springer.

[16] Almohsen, R., et al. (2022). Generative Probabilistic Novelty Detection with Isometric Adversarial Autoencoders. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[17] Deng, J., et al. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition, Ieee.

[18] Bergman, L., et al. (2020). "Deep nearest neighbor anomaly detection." arXiv preprint arXiv:2002.10445.

[19] Hilmersson, C., et al. (2008). "Crack detection in single-crystalline silicon wafers using impact testing." Applied Acoustics 69(8): 755-760.

[20] Agroui, K., et al. (2017). "Analysis techniques for photovoltaic modules based on amorphous solar cells." Arabian Journal for Science and Engineering 42(1): 375-381.

[21] Han, H., et al. (2020). "Polycrystalline silicon wafer defect segmentation based on deep convolutional neural networks." Pattern Recognition Letters 130: 234-241.

[22] Ronneberger, O., et al. (2015). U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention, Springer.

[23] Li, W.-C. and D.-M. Tsai (2012). "Wavelet-based defect detection in solar wafer images with inhomogeneous texture." Pattern Recognition 45(2): 742-756.

[24] Zoroofi, R. A., et al. (2001). "Automated inspection of IC wafer contamination." Pattern Recognition 34(6): 1307-1317.

[25] Chang, C.-Y., et al. (2011). "Wafer defect inspection by neural analysis of region features." Journal of Intelligent Manufacturing 22(6): 953-964.

[26] Nakazawa, T. and D. V. Kulkarni (2019). "Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder-decoder neural network architectures in semiconductor manufacturing." IEEE Transactions on Semiconductor Manufacturing 32(2): 250-256.

[27] Chang, C.-Y., et al. (2007). Learning vector quantization neural networks for LED wafer defect inspection. Second International Conference on Innovative Computing, Information and Control (ICICIC 2007), IEEE.

[28] Yu, N., et al. (2019). "Wafer defect pattern recognition and analysis based on convolutional neural network." IEEE Transactions on Semiconductor Manufacturing 32(4): 566-573.

[29] Abdel-Qader, I., et al. (2003). "Analysis of edge-detection techniques for crack identification in bridges." Journal of Computing in Civil Engineering 17(4): 255-263.

[30] Gehri, N., et al. (2020). "Automated crack detection and measurement based on digital image correlation." Construction and Building Materials 256: 119383.

[31] Mandal, V., et al. (2018). Automated road crack detection using deep convolutional neural networks. 2018 IEEE International Conference on Big

Data (Big Data), IEEE.

[32] Redmon, J. and A. Farhadi (2017). YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition.

[33] Szegedy, C., et al. (2015). Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition.

[34] Li, S. and X. Zhao (2018). Convolutional neural networks-based crack detection for real concrete surface. Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018, SPIE.

[35] Poma, X. S., et al. (2021). "Dense extreme inception network for edge detection." arXiv preprint arXiv:2112.02250.

[36] Roth, K., et al. (2022). Towards total recall in industrial anomaly detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[37] Zagoruyko, S. and N. Komodakis (2016). "Wide residual networks." arXiv preprint arXiv:1605.07146.

[38] Sener, O. and S. Savarese (2017). "Active learning for convolutional neural networks: A core-set approach." arXiv preprint arXiv:1708.00489.

[39] Dasgupta, S. and A. Gupta (2003). "An elementary proof of a theorem of Johnson and Lindenstrauss." Random Structures & Algorithms 22(1): 60-65.

[40] Sinha, S., et al. (2020). Small-gan: Speeding up gan training using core-sets. International Conference on Machine Learning, PMLR.

[41] He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.

Abstract

결함 탐지는 산업에서 제품의 생산성이나 질을 향상시키는데 매우 중요한 과정입니다. 그러나 나노 스케일 공정에서 결함의 형상이나 노이즈, 불분명한 배경 같은 요소들은 결함 탐지를 어렵게 만듭니다. 산업에서 결함의 비율은 매우 작고 결함 탐지를 위해서 전문적인 지식을 필요로 하는 경우도 많기 때문에 사람이 직접 결함 탐지를 수행하는 것은 매우 소모적이고 비효율적입니다. 그러므로 산업에서 컴퓨터 비전 기반 결함 탐지 모델을 활용하는 것은 시간이나 물적, 인적 자원을 절약하고 부족한 결함 데이터 문제도 해결할 수 있는 훌륭한 방법입니다. 그러나 데이터 기반 이상 탐지 모델을 실제 산업 검사에 활용하는 것은 많은 어려움을 가지고 있습니다. 해당 연구에서 우리는 결함 탐지의 목표로 하는 웨이퍼 제품에서 'resin bleed' 라는 크랙 검출을 방해하는 요소를 확인했습니다. 'resin bleed'는 정상 요소에 속하지만 머신 비전의 관점에서는 크랙과 비슷합니다. 이러한 특징들은 데이터 셋 전체에 분포되어 있는 Resin bleed가 결함 탐지 모델이 크랙들을 정상 요소들과 분명하게 구별할 수 있는 능력을 저해합니다. 이 논문에서 우리는 크랙의 엣지 성분을 강화하여 이상 탐지 모델이 크랙을 더 잘 검출할 수 있도록 하는 방법을 제시합니다. 저희가 제안하는 방법들은 결함 탐지 성능을 이미지 레벨에서 96.7%, 픽셀 레벨에서 98.6% 성능을 달성했습니다. 저희가 달성한 성과들은 기존 이상 탐지 모델을 사용했을 때와 비교하여 추가 데이터 주석 없이 이미지 레벨에서 4.5%, 픽셀 레벨에서 2.0% 성능 향상한 결과입니다.