Wafer Map Defect Pattern Recognition Using Rotation-Invariant Features

Rui Wang and Nan Chen, *Member, IEEE*

***Abstract*—In semiconductor manufacturing, the patterns on the wafer map provide important information for engineers to identify the root causes of production problems. The detection and recognition of wafer map patterns is thus an important issue in semiconductor industry. Automatic techniques are required to cut down on cost and to improve accuracy. In this study, we propose an approach to recognize patterns in the wafer maps which uses the extracted features based on the proposed weight masks. The proposed masks contain three types, namely, polar masks, line masks and arc masks. Polar masks aim to extract features of concentric patterns, while line and arc masks are designed to mainly deal with eccentric patterns like scratches. These masks can be applied to extract rotation-invariant features for the classification of the defect patterns. To demonstrate the effectiveness of our model, we apply the method to a real-world wafer map dataset. Comparisons with alternative methods show superiority of our method in the task of wafer map defect pattern recognition.**

***Index Terms*—Semiconductor wafer map, defect recognition, feature extraction, rotation invariance.**

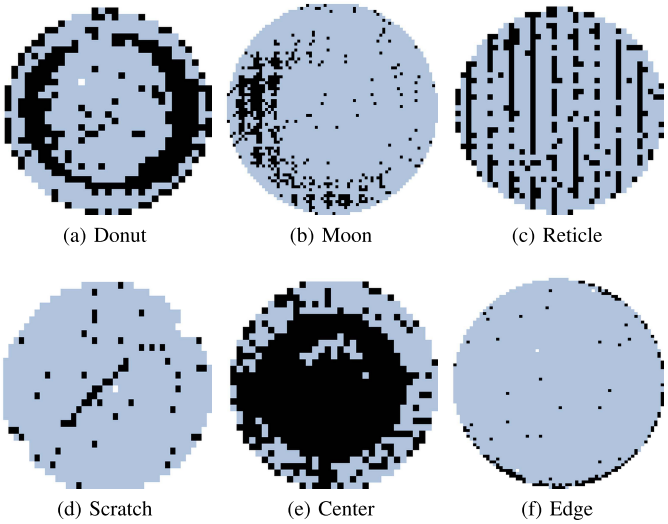


Fig. 1. Typical examples of wafer map failure patterns.

I. Introduction

THE SEMICONDUCTOR manufacturing process has become more and more complex which may involve hundreds of steps. Nowadays, the process is highly automatic and precisely monitored thanks to technological developments. However, defects are still unavoidable due to process problems or erroneous human operations. To ensure the good performance of each die, wafer testing is performed after each wafer fabrication stage. In this article, it is assumed that each die is assigned a binary value based on the wafer testing results, i.e., 0 for good dies and 1 for defective dies. A wafer map is used as the graphical representation of locations of the defects on the wafer. In general, defects on wafer maps have two categories: the first includes random defects and the second is composed of clustered defects. Random defects are often caused by uncertain environment fluctuation and variation of the process. Clustered defects attract more attention from engineers and researchers because they usually provide valuable information on specific manufacturing problems. Clustered defects may form different patterns, e.g., scratches, rings, repeats, etc. Fig. 1 shows several typical examples of wafer map failure patterns. The black pixels show locations of defective dies and the light blue pixels are normal dies on the wafer. For example, typical patterns like scratches are improperly made during material handling, while edge rings are usually caused by etching problems [1]. Therefore, if correctly recognized, the defect patterns will help identify and eliminate the problems in manufacturing processes, which in turn improves process yield and reduces costs.

To detect and classify wafer map defect patterns, numerous studies have been conducted. These studies aim to develop automatic defect detection and recognition techniques to cut down on cost resulting from visual recognition performed by experienced engineers. Current methods can be roughly divided into three categories. The first category involves constructing statistics to monitor the defect patterns on wafer maps. For instance, [2] used the multivariate Hotelling T^2 chart based on the number of defects and clustering index. Reference [3] proposed a step-down spatial randomness test for detecting abnormal wafers based on spatial correlogram. These methods can successfully separate normal and abnormal wafer maps, but they can hardly distinguish different spatial defect patterns.

The second category includes model-based clustering methods, which try to recognize failure patterns by assuming shape-specific distributions for failure regions. Reference [4]

Manuscript received July 31, 2019; revised September 16, 2019; accepted September 23, 2019. Date of publication September 27, 2019; date of current version October 29, 2019. *(Corresponding author: Nan Chen.)*

The authors are with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576 (e-mail: isecn@nus.edu.sg).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSM.2019.2944181

used Gaussian EM to detect elliptic and linear patterns, and spherical-shell algorithm to estimate ring patterns. Reference [5] modeled global defects using non-homogeneous Poisson process and local defects using bivariate normal distribution and principal curve. This kind of methods is advantageous in simultaneously identifying multiple failure patterns in a single wafer map. However, they can be computationally intensive to estimate parameters and the recognition is only limited to simple pre-defined shapes.

The last category is related to machine learning methods. These include both unsupervised and supervised learning methods based on the prior knowledge of class labels. In unsupervised learning methods, adaptive resonance theory network (ART1) [6], self-organized map (SOM) [7] and K-means [8], [9] have been developed to construct clusters of wafer maps. These methods have the advantage that new failure patterns can be introduced and identified. When class labels and enough training samples are available, the performance of supervised learning is usually superior to unsupervised learning methods. Typical examples are support vector machine (SVM) [10], K-nearest neighbors (KNN) [11], neural networks [12], [13]. To ensure the performance of supervised methods, large amount of high-quality data is required.

Most of the studies mentioned above used raw wafer maps directly as inputs for supervised defect pattern recognition, but sometimes they are inappropriate because of unsatisfactory performance and high computation cost. Therefore, feature generation is usually seen as an important step to reduce computation time and to improve accuracy. Features extracted from wafer maps largely influence the performance of defect pattern recognition. Effective features would definitely improve the accuracy and computation time to a large degree. Ideally, the defect patterns are identical with respect to orientation so that the samples can be classified easily. It is observed that wafers have a round shape and it is hard to define the image orientation. Features are called rotation invariant if their output results are not affected by the rotations of input image. In the problem of wafer map detection and recognition, we would like the same type of defect patterns with variations to have similar feature vectors for classification. This means rotation invariant features are more preferred in this context.

For the extraction of features from wafer maps, one direction involves automatic extraction of features by deep learning approaches. For example, the variational autoencoder can be used to learn latent data representations of the wafer map [14], [15]. The other direction considers specifically engineered features for the classification of wafer maps. These extracted features may consist of geometric features, texture features, transformation-based features, etc. [1], [10], [16], [17]. Deep learning approaches are more generalized and the effectiveness benefits from the increasing amount of training data. But training requires a large dataset and the extracted features cannot be easily explained. When it comes to specific problems, pre-defined features could give better results because of the use of domain knowledge and are more interpretable for domain experts. If the features are designed properly, the computational cost will be largely reduced.

In the literature, various rotation-invariant feature descriptors have been developed. One of the most popular method is the scale invariant feature transform (SIFT) [18], which has illumination, scale, rotation, and affine invariant properties. Reference [19] proposed a rotation invariant histogram of oriented gradients (RIHOG) to overcome the sensitivity to image rotation of the classic histogram of oriented gradients (HOG) algorithm. Other typical examples of rotation-invariant feature descriptors are Zernike moments [20] and speed-up robust features (SURF) [21], etc. However, the existing methods require relatively long computation time. In addition, the determination of the descriptor parameters (e.g., window size and number of neighboring pixels) can be challenging, as the influence of the change on the final performance is not intuitive. Therefore, we aim to develop features that are interpretable and require low computation cost, while preserving the power in recognizing defect patterns.

In this study, we propose a wafer map defect recognition method based on feature extraction by designed features. This is accomplished by applying rotatable weight masks on the circular area of wafer maps. For each unique mask, maximum value of the results created by rotated version of the mask is returned to ensure rotation invariance. The extracted features are easily interpretable with our designed form of masks. Then the features are sent to a feedforward neural network for the classification of defect patterns. We also apply our method to the real-world wafer map data. Experiment results show the effectiveness of our proposed method.

The rest of this article is organized as follows. Section II describes the detailed procedure of feature extraction and classification for wafer map defect pattern recognition. Section III gives experimental results with an application in real-world data. Section IV concludes the article and discusses directions for future work.

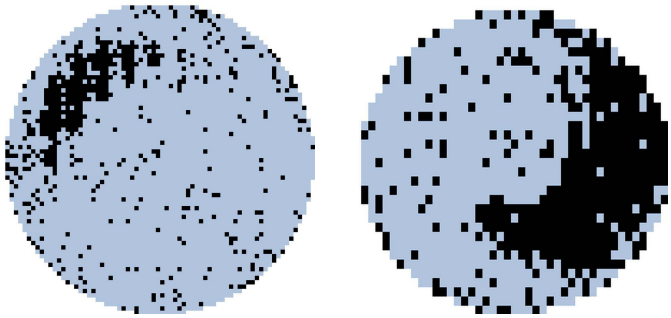
II. Methodology

In this section, we elaborate the framework of how to apply the proposed rotation invariant weight masks to detect wafer map failure patterns. The method consists of preprocessing, feature extraction and classification steps.

A. Preprocessing of Wafer Maps

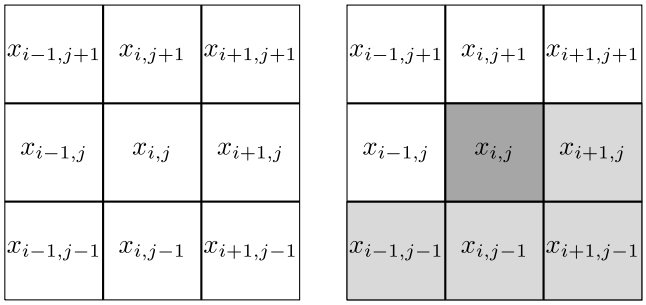
In this context, wafer map preprocessing contains two steps: denoising and resizing. Fig. 2(b) gives an example of real data. Both figures show the defect pattern “Moon”. There exists random defective points in the figure, these dots are not helpful for defect recognition and are seen as noise in the wafer map. Image denoising aims to remove these defects in the wafer map and let the defect pattern stand out. The dies on the wafers have different sizes, which is represented by the density of points in the wafer map. In order to detect failure patterns on wafer maps of different products, we should convert wafer maps into the same format.

Spatial filtering is a simple and widely used method to reduce random noise, see [1] for example. The spatial filter



(a) (b)

Fig. 2. Two examples of "moon" patterns with different die sizes.



(a) (b)

Fig. 3. (a) A 3×3 spatial filter (b) One example of dies covered by the filter at the edge.

computes a weighted sum of the binarized values of points in the neighborhood of each point. When this weighted sum is larger than a threshold, the die is regarded as defective after denoising. A 3×3 filter is shown in Fig. 3(a). However, it cannot be directly applied to wafer maps, as the wafer has a circular shape. When it comes to the edge of the wafer, the number of surrounded dies is fewer than that in the middle. Fig. 3(b) gives an example of dies masked at the edge by the 3×3 filter. The dark gray box is the die at the edge of the wafer, and light gray areas show the surrounding dies of the chosen die on the wafer. In this case, the filter only covers five dies centering at one edge die. Classic spatial filtering uses the same weighted sum for denoisng over the wafer map, which may remove non-random patterns near the edges. Moreover, spatial filtering is not effective in dealing with repetitive or connected long and thin patterns such as "Scratch" and "Reticle". The patterns will be largely removed using spatial filtering. We then make some modifications to existing spatial filtering to overcome the effects at edge and the difficulty for curvilinear patterns.

In a wafer map, the pattern formed by the defective dies is important. Thus, if a die is in normal state, it stays in normal state after denoising. If a die is defective, we choose a threshold to determine if it is an isolated defect or not. A natural choice is the percentage of failures in the surrounding of dies within the wafer area. We use a pair of coordinate (i, j) to denote the location of each pixel in the wafer map, x to denote the state of a pixel (1 for defective dies and 0 for normal dies and pixels outside the wafer area), and R to denote the weighted ratio. Based on the above assumption, we use $N(i, j)$ to denote the number of neighbors of a die at location (i, j), $\mathcal{M}$ to denote the set of all possible positions in the wafer map, and $2t + 1$ to denote the size of the filter. We define the weighted ratio for the central die as follows:

$$R(i,j) = \frac{1}{N(i,j)} \sum_{\substack{m=-t,\\ m\neq 0}}^{t} \sum_{\substack{n=-t,\\ n\neq 0}}^{t} x(i,j)x(i+m,j+n)I(i+m,j+n), \tag{1}$$

and

$$N(i,j) = \sum_{\substack{m=-t,\\ m\neq 0}}^{t} \sum_{\substack{n=-t,\\ n\neq 0}}^{t} I(i+m,j+n) \tag{2}$$

where $I(i + m, j + n)$ is the indicator function to represent whether the coordinate refers to a valid die on the wafer map, and

$$I(i,j) = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{M}, \\ 0, & \text{if } (i,j) \notin \mathcal{M}. \end{cases} \tag{3}$$

Thus, in Fig. 3(b), $N(i, j) = 4$. For a defective die, if one of the surrounding dies is defective, $R(i, j) = 1/4$. If we use L to denote the threshold, the state of the die after denoising x_d is:

$$x_d(i,j) = \begin{cases} 0, & \text{if } R(i,j) < L, \\ 1, & \text{if } R(i,j) \geq L, \end{cases} \tag{4}$$

Fig. 4 shows the comparison of denoising results by spatial filtering (thresholding at 4/9) and our method for "Reticle" and "Edge" patterns described in Fig. 1. The filter size 3×3 is used for comparison. Intuitively, we remove the "isolated" failure points in the wafer map, then $L = 1/8$. The principle of the spatial filtering is to replace the value of a die according to the average value of defective dies in the neighborhood. The weighted ratio is defined as $R(i, j) = \frac{1}{9} \sum_{m=-1}^{1} \sum_{n=-1}^{1} x(i + m, j + n)$. Similar to (4), if $R(i, j)$ is greater than the threshold value, the die is marked as defective. It can be seen that isolated defect points are mostly removed by our method. For the pattern "Edge", spatial filtering gives comparable results with our method except that our method is able to keep more defects at the edge when the pattern curve is thin. Our method works apparently better than spatial filtering for the denoising of the pattern "Reticle". Spatial filtering could change normal dies to defective state while erasing curves according to the neighborhood. In contrast, our method only performs thresholding on defective dies and that guarantees the stand out of curvilinear patterns.

To handle the difference in wafer sizes and die sizes, the wafer map images should be normalized. In this case, image interpolation is used to resize images. It works by using known data to estimate values at unknown locations. The most basic method commonly used is the nearest neighbor, a procedure that does not introduce any artificial data into the output. To resize an image to a predefined size, existing packages have been well developed.

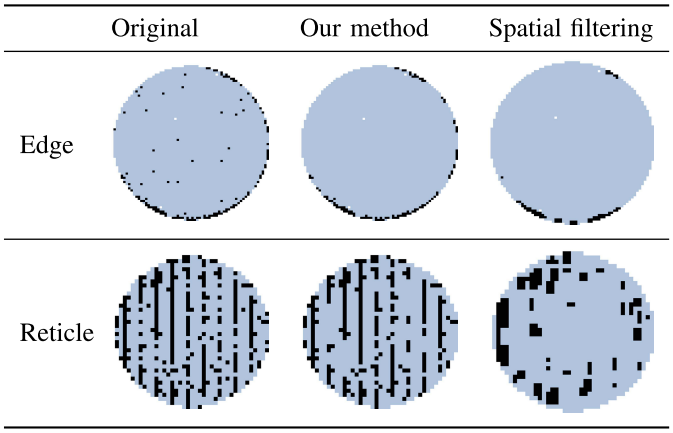


Fig. 4. Comparison of denoising results for the pattern "Reticle" and "Edge".

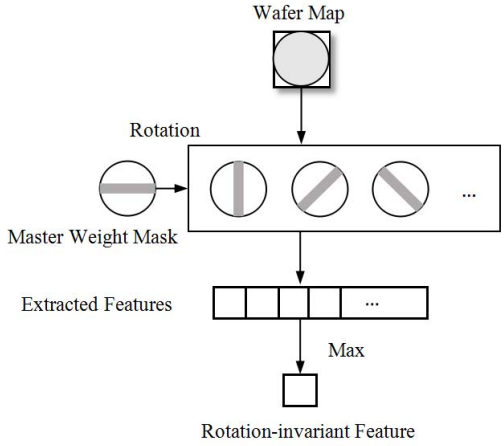


Fig. 5. Feature extraction to achieve rotation invariance.

B. Feature Extraction

A good group of features should be representative of the patterns, highly interpretable, and in the context of wafer map, rotation invariant. Features exhibiting rotation invariance is always of interest in real applications as object rotation is quite common in image data collection. Rotation-invariant features remain the same for wafer maps of different orientations. Considering the shape of wafer maps, we design a series of rotatable weight masks which all have the same shape as the wafer map. In this study, the proposed masks contain three types, namely, polar masks, line masks and arc masks. Polar masks extract features of concentric patterns, while line and arc masks mainly deal with eccentric patterns like scratches. The feature is calculated by summarizing the element-wise production results of the weight mask and the wafer map. Given a wafer map I and a weight mask M, the feature can be obtained as:

$$F_M(I) = \sum_{(i,j)\in R_I} I(i,j)M(i,j). \tag{5}$$

Rotation invariance is achieved by making several rotated copies of each master mask and only the max feature value is retained for each master mask, see Fig. 5. This ensures that the feature can best describe the pattern that the weight mask defines.

1) Polar Masks: First, we consider a group of polar coordinate based weight masks. This kind of masks is capable of capturing the concentric patterns of wafer maps. A binning method is considered to calculate the number of failures inside a bin and the difference in failure percentage between bins. We consider two binning types: angle binning and circle binning. Angle binning divide the angle of π into equally spaced positions. Fig. 6(a) shows angle binning of eight regions, their area are exactly the same. Circle binning draws concentric circles of the wafer, which separate the wafer map into annuli. Here the radius values of the created concentric circles are not necessarily be equipartition of the length of wafer radius. Fig. 6(b) illustrates that the wafer is divided into three parts by circles whose radiuses are 0.3 and 0.5 times the wafer radius. We can define the binning using polar coordinates. We

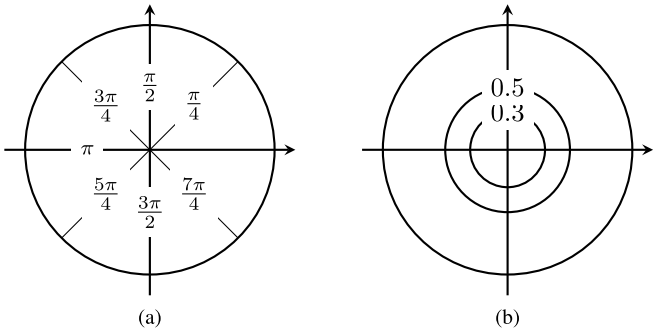


Fig. 6. Examples of binning: (a) angle binning (b) circle binning. The circle shape represents a wafer map.

denote N_a the sampling parameter of θ, and ρ to be partitioned into N_c intervals. In this case, angle binning divides 2π into equal intervals, i.e., $\theta = (\theta_0, \theta_1, \ldots, \theta_{N_a})$, where $\theta_0 = 0, \theta_1 = 2\pi/N_a, \ldots, \theta_{N_a-1} = 2(N_a-1)\pi/N_a, \theta_{N_a} = 2\pi$. Similarly, circle binning divides radius into intervals with $\rho = (\rho_0, \rho_1, \ldots, \rho_{N_c})$, and $\rho_0 < \rho_1 < \cdots < \rho_{N_c}$ with $\rho_0 = 0, \rho_{N_c} = R$, R is the radius of the wafer. ρ here can be given based on the definition of defect patterns.

Next we discuss how to determine the weights inside the proposed mask. For each bin, we assign the same weight for elements inside. To make it simple yet effective, we come up with a weight assignment method based on the structure of this mask. We choose the weights of each bin to be in the set of $\{0, -1, 1\}$. Weight value 1 count the number of failures inside bins, while the combination of $\{-1, 1\}$ describes the difference between different bins. The opposite weight values $1, -1$ are only allowed inside the same annulus, as it is meaningless to compare failure numbers when the size of the bins are different. Let w_{ij} denote weight assigned to the bin $(\theta_{i-1} \le \theta < \theta_i, \rho_{j-1} \le \rho < \rho_j), i = 1, 2, \ldots, N_a; j = 1, 2, \ldots, N_c$, then the weights (w_{ij}) forms a matrix $\mathbf{W}$ with size $N_a \times N_c$.

Here, we use the idea of bitwise operations to explain how to generate weight matrix $\mathbf{W}$. Our method starts from configuration for each dimension, then conducts pairwise production of the bits to get a 2-d weight matrix. First, we define weight candidates for angular bins $b_{\cdot j}$. To make it easy to

define the configuration method, here we only limit N_a to be $N_a = 2d, d \in \mathbb{N}^+$. The set for weights of angular bins A can be divided into three parts.

i $a_{1,1} = (\underbrace{1, \ldots, 1}_{N_a})$ is the bit array (base 2) of $2^{N_a} - 1$. For $i = 2, \ldots, N_a, i \neq N_a/2 + 1$, $a_{1,i} = a_{1,1} \ll (i-1)$.

ii $a_{2,1} = (\underbrace{0, \ldots, 0}_{N_a/2} \underbrace{1, \ldots, 1}_{N_a/2})$. For $i = 2, \ldots, N_a/2$, $a_{2,i} = a_{2,1}$ XOR $(a_{2,1} \ll (i-1))$.

iii For $i = 1, 2, \ldots, N_a/2$, each element in $a_{3,i}$ is given by $a_{3,i,*} = 2 \times a_{2,i,*} - 1$.

The notation "$a \ll n$" means to left-shift a by n bits. The bitwise XOR performs the logical OR operation on each pair of bits.

For instance, when $N_a = 4$, in case i, $(15)_{10} = (1111)_2$, so $a_{1,1} = (1, 1, 1, 1)$. $a_{1,i}$ is generated by left-shift $a_{1,1}$ by $i - 1$ bits. Thus, $a_{1,2} = a_{1,1} \ll 1 = (1, 1, 1, 0)$ and $a_{1,4} = a_{1,1} \ll 3 = (1, 0, 0, 0)$. $a_{2,3}$ is removed in this case because it represent the same configuration as $a_{2,1}$. For case ii, $a_{2,1}$ is $(0, 0, 1, 1)$. Based on the operation above, $a_{2,2} = a_{2,1}$ XOR $(a_{2,1} \ll 1) = (0, 0, 1, 1)$ XOR $(0, 1, 1, 0) = (0, 1, 0, 1)$. For case iii, $a_{3,*}$ actually replaces "0" in the corresponding array in A_2 with "-1", hence $a_{3,1} = (-1, -1, 1, 1)$ and $a_{3,2} = (-1, 1, -1, 1)$. Thus, for $N_a = 4$, the configuration becomes:

$$\mathcal{A}_1 = \{(1, 1, 1, 1), (1, 1, 1, 0), (1, 0, 0, 0)\},$$
$$\mathcal{A}_2 = \{(0, 0, 1, 1), (0, 1, 0, 1)\},$$
$$\mathcal{A}_3 = \{(-1, -1, 1, 1), (-1, 1, -1, 1)\}.$$

$\mathcal{A}$ is the union of the collection $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$. $\mathcal{A}_1$ counts the number of pixels of consecutive bins alongside angular bins, $\mathcal{A}_2$ takes care of symmetric regions about the center, while $\mathcal{A}_3$ compares the difference of defect pixels between central symmetric areas. The size of set $\mathcal{A}$ is $|\mathcal{A}_1| + |\mathcal{A}_2| + |\mathcal{A}_3|$, which equals $N_a - 1 + N_a/2 + N_a/2 = 2N_a - 1$. Thus, When $N_a = 4$, $|\mathcal{A}| = 7$.

Next, we define weight candidates for circular bins b_i.. This is done by setting some values to 1 and the rest to 0. Again we apply the idea of bit operation. N_c here can take both even and odd values, as symmetric property is no longer needed. For $i = 1, 2, \ldots, N_c$, let $\mathcal{C}_i = \{c_{i,1}, c_{i,2}, \ldots, c_{i,N_c+1-i}\}$ denote the derived set from each initial array $c_{i,1}$, which is the bit array (base 2) of $2^i - 1$. For $j \geq 2, j \leq N_c + 1 - i$, $c_{i,j} = c_{i,1} \ll (j-1)$. For example, when $N_c = 3$,

$$\mathcal{C}_1 = \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\},$$
$$\mathcal{C}_2 = \{(0, 1, 1), (1, 1, 0)\},$$
$$\mathcal{C}_3 = \{(1, 1, 1)\}.$$

$\mathcal{C}$ is the union of the collection $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{N_c}\}$. In this case, $\mathcal{C}$ calculates the number of defect pixels within the annulus determined by the combination of radiuses in ρ. The size of set $\mathcal{C}$, $|\mathcal{C}|$ is $|\mathcal{C}_1| + |\mathcal{C}_2| + \ldots, |\mathcal{C}_{N_c}|$, which equals $(1 + N_c)N_c/2$. Thus in the case of $N_c = 3$, $|\mathcal{C}| = 6$.

Once we have set $\mathcal{A}$ and set $\mathcal{C}$ available, we can construct the weight matrix of the masks by pairwise product the array elements from $\mathcal{A}$ and $\mathcal{C}$. Multiplication between a column vector and a row vector returns a matrix $\mathbf{W}$ with each element $w_{i,j} = x_i y_j$. Denote the set of weight matrix with $\mathcal{P}$, then $\mathcal{P} = \{x^T y | x \in \mathcal{A}, y \in \mathcal{C}\}$, where x and y are seen as row vectors in the equation. Thus, the total number of master weight masks for a given pair of (N_a, N_c) is:

$$|\mathcal{P}| = (2N_a - 1)(1 + N_c)N_c/2. \tag{6}$$

For the above example, the weight matrix of the master mask based on $a_{2,2}$ and $c_{2,1}$ is:

$$a_{2,2}^T c_{2,1} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} (0, 1, 1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \tag{7}$$

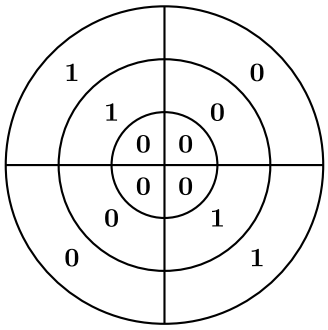


Fig. 7. Example of a polar weight mask with $N_a = 4, N_c = 3$.

This weight configuration of $N_a = 4, N_c = 3$ is illustrated in Fig. 7. With this combination of binning methods, we define 12 bins. The value inside each bin shows the weight given based on the matrix above. Each column in matrix represents weights of bins formed by different angles alongside an annulus. Similarly, each row gives the weights radially in the corresponding quarter of the circle.

2) Line Masks: The second group of features aims to represent line scratches from the wafer map. Here we use stripe areas in the weight mask as accumulators of defect points of linear patterns. The stripe area can be defined in the following way. A line in $x - y$ plane can be uniquely defined by its distance from the origin ρ and its angle θ as

$$x \cos\theta + y \sin\theta = \rho, \tag{8}$$

where θ is within $[0, \pi)$.

Thus, each (θ, ρ) pair uniquely define a line. For the mask, each stripe can be uniquely defined as $(\theta = \theta_i, \rho_{j-1} \leq \rho < \rho_j)$, see Fig. 8(a). Let the weight within each stripe be 1 and the weight outside the stripe be 0, then each mask count the number of defect points falling within this stripe. So for line patterns, the feature value should be large for the corresponding stripe which overlaps the most with the line. Here for a given θ, ρ is partitioned into N_l intervals by $\rho_0 < \rho_1 < \cdots < \rho_{N_l}$, satisfying $\rho_0 = 0, \rho_{N_l} = R$. Thus in this case, total number of master masks for this type is N_l.

3) Arc Masks: Previouly mentioned two categories of masks help detect concentric circular patterns and linear patterns, however, circular scratches are frequently seen which are usually eccentric with the wafer map. To improve recognition accuracy of this kind of pattern, we design arc masks to

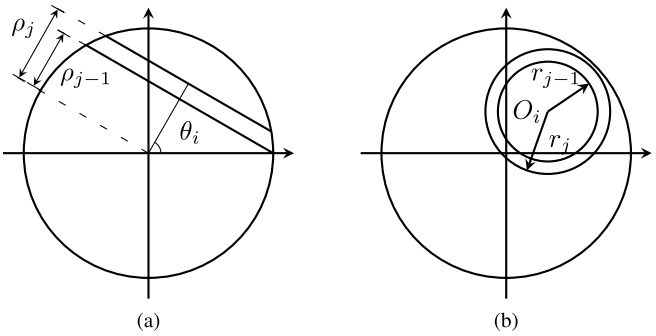


Fig. 8. (a) Line mask (b) Arc mask.

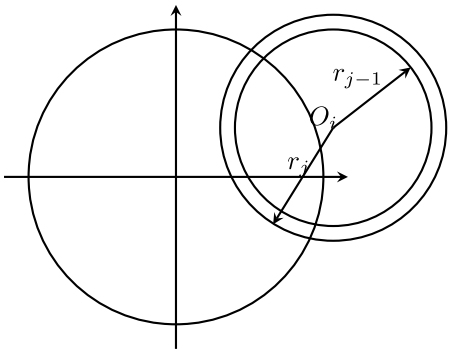


Fig. 9. An example of annulus center outside the wafer.

extract features representative of circular scratches. Basically, a circle can be parameterized as

$$(x-a)^2 + (y-b)^2 = r^2, \tag{9}$$

with its center $O(a, b)$ and radius r. Each pair of (O, r) defines an unique circle. Here we define rings on masks to accumulate the defect points inside, which can be written as $(O_i, r_{j-1} < r \leq r_j)$. Fig. 8(b) is an example of such mask. The annulus is determined by its center O_i ,its inner radius r_{j-1} and its outer radius r_j. We can then assign weights for the masks, 1 for the area within this annulus, 0 for the area outside this annulus. So the proposed arc mask calculates the number of defects within each annulus. Larger value is returned for annulus which has more overlapped area with the defect patterns.

As rotated copies are made later to guarantee rotation invariance, a simple way to choose the center of annulus is to put it on the x-axis, where $O(c, 0), c \geq 0$. Here the center of the annulus O is not necessarily within the area of wafer, because it is possible that only part of the annulus is within the wafer, see Fig. 9. Without loss of generality, we limit the annulus radius r to be $r_0 < r_1 < \cdots < r_{N_r}, r_0 = R_l, r_{N_r} = R_h$, where R_l and R_h respectively denote the lower and upper limit of the radius. r here can be equally spaced between $[R_l, R_h]$. For the location of center, c is chosen between $(0, R_c]$, with a total number of choices N_o. The resulting centers are $O_1(c_1, 0), O_2(c_2, 0), \ldots, O_{N_o}(c_{N_o}, 0)$, with $c_1 < c_2 < \cdots < c_{N_o}$. Similarly, we can simply divide $(0, R_c)$ into N_o equidistant intervals. Thus, considering the combination of center and radius groups, $N_r N_o$ master masks are created in this category. To this end, we have defined three types of masks. The total number of masks is

$$n = (2N_a - 1)(1 + N_c)N_c/2 + N_l + N_r N_o. \tag{10}$$

To ensure rotation invariance for these proposed round weight masks, for each of the master weight mask, we create several rotated versions of it, see Fig. 5. The rotation angle α is sampled at equal distance intervals from 0 to 2π. If we denote the number of total masks after rotation of one master mask to be N_t, then the set of rotation angles is $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_{N_t}\}$, where $\alpha_1 = 0, \alpha_2 = 2\pi/N_t, \ldots, \alpha_{N_t} = 2(N_t - 1)\pi/N_t$. Only the max value of features generated from each rotated group of the master mask is returned. This step ensures invariance to rotations and eliminates to influence of orientations of input images. The extracted features are then sent to feedforward neural network for classification.

C. Classification

Wafer map defect pattern recognition problem is a multiclass classification problem in this case. As for multiclass problems, things become more complicated. There exists three categories of methods to solve supervised multiclass classification problems [22]. The first category simply extend the binary classification algorithms to the multiclass case. This kind of algorithms is naturally extendable, e.g., neural networks, decision trees and k-nearest neighbors. The second category solves the multiclass classification problem by decomposing the problem into a set of binary classification, which can then be efficiently solved using binary classifiers. This involves combining some of the classes together and comparing the newly built classes in each group of two. One of the widely used methods in this category is the support vector machine. The last category uses a hierarchical classification idea that arrange the classes into a tree, then the process continues at each node that the classifier differentiates between the child class clusters. New patterns are added following a path from the root node to the upper-level leaves.

In our study, neural network is worth consideration for the recognition of defect patterns. Feedforword neural network has successfully been applied in pattern recognition and image processing. This class of networks consists of multiple layers of units that are interconnected in a feed-forward way. The number of hidden units and layers is typically defined subjectively based on the complexity of the problem. Optimal weights minimizing the loss function are acquired by applying supervised learning rules. Here, we choose classic multilayer perceptron as the classification model. The input of this network is the features extracted from previous steps. Dropout [23] can be used to prevent overfitting. In this study, we use neural network as classifier to do defect pattern recognition. It should be noted that other techniques may also be applicable.

III. Experiments and Results

In this section, we apply the proposed method to a real-world wafer map dataset, and present the performance analysis compared with baseline methods.

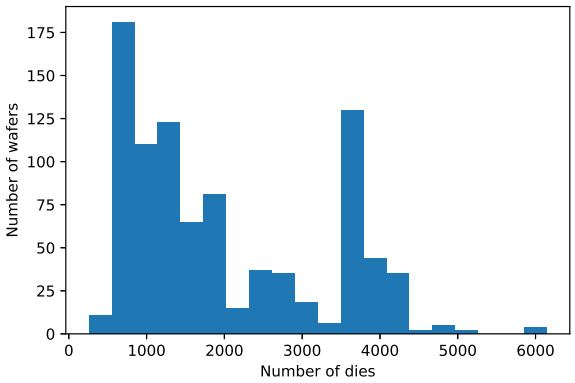


Fig. 10. Histogram of the number of dies for the wafer maps.

TABLE I
NUMBER OF INSTANCES IN EACH CLASS

Pattern	Donut	Moon	Reticle	Edge	Scratch	Center	Normal
Sample size	17	59	16	352	144	99	218

TABLE II
PARAMETERS FOR PREPROCESSING AND FEATURE EXTRACTION

Parameters	Value
Filter size (k)	3
Threshold for weighted ratio (L)	1/8
Sampling parameter of angle θ for polar masks (N_c)	5
Sampling parameter of radius ρ for polar masks (N_a)	4
Sampling parameter of radius ρ for line masks (N_l)	7
Higher limit of annulus center location for arc masks (R_c)	1.2R
Lower limit of annulus radius for arc masks (R_l)	0.5R
Higher limit of annulus radius for arc masks (R_h)	R
Sampling parameter of annulus center location for arc masks (N_o)	12
Sampling parameter of annulus radius for arc masks (N_r)	6
Number of rotated copies of the master mask (N_t)	16

TABLE III
COMBINED CONFUSION MATRIX FOR RECOGNITION RATE (%) FOR THE SEVEN DEFECT PATTERNS

Pattern	Donut	Moon	Reticle	Scratch	Center	Edge	Normal
Donut	100	0	0	0	0	0	0
Moon	0	90.7	0	1.3	0	6.7	1.3
Reticle	0	0	75.0	15.0	0	10.0	0
Scratch	0.6	0.6	1.6	85.6	0	1.6	10
Center	0	0	0	0	100	0	0
Edge	0	2.7	0.2	1.4	0	94.3	1.4
Normal	0	0	0	6.3	0	2.6	91.1

A. Data Description

The wafer map dataset comprises 904 wafer maps collected from real semiconductor manufacturing. It includes 218 normal wafer samples and 686 defective wafer samples. Information like die position, wafer radius and wafer functionality is included in the dataset. The faulty wafer has six patterns, i.e., "Donut", "Moon", "Reticle", "Edge", "Scratch', "Center", see Fig. 1. The histogram of the number of dies in each wafer in Fig. 10 shows that the number of dies (points) in the wafer maps varies a lot. This dataset is a mixture of wafer maps of different products. The number of samples for each class are shown in Table I. It is illustrated that the number of samples for each defect type in this dataset is unevenly distributed.

B. Performance Analysis

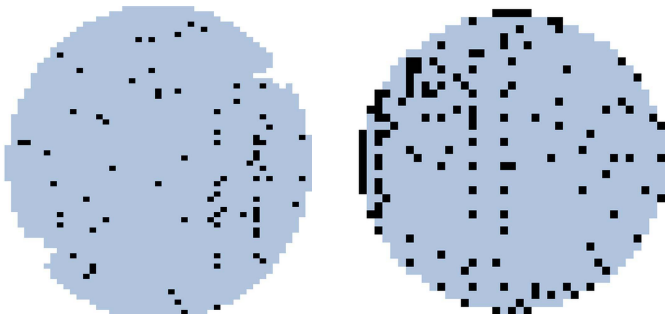
To test the performance of proposed method, we follow the proposed framework. We first denoise the wafer map by $k = 3$ and $L = 1/8$. In this study, $L = 1/8$ ensures the removal of isolated defects. Then we resize the image matrix to 32×32 guarantee the same format of input data. The choice of the mask parameters is dependent on the location and shape of the defect patterns. For polar masks, we choose $N_c = 5, N_a = 4$. Angle binning here divides 2π into equal intervals with angles $\theta = (0, \pi/2, \pi, 3\pi/2, 2\pi)$. We define the circle binning parameter $\rho = (0, 0.2R, \ldots, 0.8R, R)$, where R is the radius value of the wafer map in the input matrix. The choice of binning intervals agrees with the location of circular patterns. For example, "Edge" defect pattern appears mostly at radius interval $[0.8R, R]$, so circle binning at $\rho = 0.8R$ could help detect "Edge" pattern. For line masks, R is divided into 7 intervals $N_l = 7$, and $\rho = (0, 0.3R, 0.4R, \ldots, 0.8R, R)$, with the length of step larger for center and edge. Further, the upper limit for annulus center $R_c = 1.2R$ and the range of annulus radius $R_l = 0.5R, R_h = R$ for arc masks, $N_r = 6, N_o = 12$ with step $0.1R$. This binning method is consistent with the locations and areas of curvilinear patterns. The step $0.1R$ is determined by the width of curvilinear patterns. It guarantees that only one curve occurs in one strip, while the detected curve can be largely matched within the masked area. The number of rotated versions $N_t = 16$ (the mask is rotated every $\pi/8$ rad). The total number of master masks is 184 from (10), and the overall feature dimension equals 184 after selecting the best result for each group of rotated masks. We summarize all the parameters in Table II.

We choose to use two hidden layers (both with 400 hidden units) for the structure of multilayer perceptron in our experiment. The rectified linear units (ReLU) activation function, proving to be free from the problem of gradient divergence, is investigated instead of sigmoid active function for classification using multilayer perceptron model. To relieve the problem of overfitting, dropout is used (p is set 0.5) based on experiments in the multilayer perceptron model.

We use stratified K-fold cross validation to evaluate the performance of the proposed method. It first splits the dataset into K folds, and then averages the error rates created by K experiments that use $K-1$ folds for training and the left fold for validation. The folds are made in a way that the percentage of samples for each class is preserved. This ensures that folds are similar to each other. The next problem is how to determine the number of folds. Large number of folds lowers error rates, but meanwhile increases the variance and computation time. Lower K is usually cheaper but more biased. In this study, we empirically choose $K = 5$ based on our dataset.

Table III shows the combined confusion matrix for the proposed method on the test set with overall accuracy 92.3%. In the table, the annotations in the first row represent predicted patterns, and the first column shows true patterns. The values in the diagonal locations represent the average recognition rate of each defect type from cross validation.

From this table, the overall performance of the proposed method is quite good. This result also shows that this method



(a) (b)

Fig. 11. Examples of wrongly detected Reticle patterns: (a) Scratch (b) Edge.

can effectively differentiate between defect patterns and the normal pattern on a small dataset. Obviously, the most classification errors are from misidentifying "Reticle" as other defect patterns. This may be partially due to the small sample size of "Reticle" defects. Also, some of the patterns may be easily confused with others in real applications, see Fig. 11. Although the model misclassifies the patterns, it is still forgivable. This indicates that the performance is acceptable for users, as the boundaries between these patterns is not so clear.

C. Comparisons

To further investigate the performance of our method, we also test the dataset with other methods. We compare the proposed method with other classification methods together with other rotation-invariant features. The average of accuracy is calculated based on a five-fold cross validation for the evaluation of each of the method. Table IV summarizes the comparison results, and the values in parentheses are the standard errors.

First, we validate the effectiveness of each group of masks. The proposed three group of masks deals with different kinds of patterns, at the same time they help identify patterns jointly by providing more information on the shapes. Intuitively, polar masks work better at large and concentric patterns like "Center" and "Donut", while line and arc masks are more proficient at detailed and eccentric patterns like "Scratch". Therefore, here we use polar masks alone and a combination of line and arc masks to train the model respectively. To make it simple to represent, we use the initial of each kind of masks to stand for the features we use to train the model. We use NN to be short for feedforward neural network. For example, P-NN means to use only features extracted by polar masks to train the multilayer perceptron model. In addition, to test the effectiveness of feedforward neural network, we simply use the extracted features to train a boosting model, which is a classic method for classification. This method is called PLA-Boosting in this article. We also use random forest (RF), k-nearest neighbors algorithm (KNN) and support vector machine (SVM) to perform defect pattern classification, which is short for PLA-RF, PLA-KNN and PLA-SVM for comparison respectively. The algorithms are applied using Python scikit-learn package. The parameters of Boosting, KNN and SVM are setup as follows: For boosting, LogitBoost is applied with 200 estimators and the learning rate is set to 0.1. The number of estimators is also set to 200 for random forest. The number of neighbors to use for KNN is 10. For the SVM, we use the radial basis function kernel (γ=0.3) with C-Support Vector Classification algorithm (C=2.0). The left side of Table IV shows the comparison between the accuracy of the proposed method (PLA-NN) and other methods using different group of masks and other classifiers. For different mask groups, the same NN model is applied to avoid any uncertainties. It can be shown that using NN, to combine three category of masks (184) is better than use only polar masks (105) and a combination of line and arc masks (79). Moreover, line-arc masks perform slightly better than polar masks. With all the three proposed masks, NN performs the best for accuracy. NN also reaches comparable results with SVM for error, which outperforms all the other methods. Table V further compares the recognition rate for each pattern. It is generally consistent with our intuition. For NN model, P-NN performs better for "Center", "Dount" patterns compared with LA-NN, while LA-NN is better at the recognition of "Moon", "Reticle" and "Scratch". For each of the patterns, PLA-NN mostly performs better than using just one or two groups of masks. Using all PLA masks, Boosting and SVM give good recognition results for "Edge" and "Normal" patterns, but they perform quite worse at "Moon" and "Reticle" compared with PLA-NN. Therefore, we choose to use all three proposed masks accompanied with feedforward neural network for this study.

We then compare our methods with existing feature extraction methods in the literature. We consider existing solutions to wafer map defect pattern classification including rotation moments invariants and geometrical features (RMI-G) [16], Radon-based and geometry-based features (R-G) [10], geometrical, gray, textual and projection features (G-G-T-P) [1], and variational autoencoder (VAE) [15]. We also use other popular rotation-invariant features, i.e., scale invariant feature transform (SIFT) [18] and Zernike moments (ZM) [20] for the recognition of defect patterns. Boosting is applied for the classification task, due to its stable performance and less number of tunning parameters. Convolutional neural networks with

TABLE IV
ACCURACY COMPARISON FOR DIFFERENT METHODS

Method	Accuracy (%)	Method	Accuracy (%)
PLA-NN	**92.3** (1.8)	**RMI-G**	86.3 (2.1)
P-NN	87.7 (2.0)	**R-G**	86.6 (2.2)
LA-NN	91.2 (2.6)	**G-G-T-P**	82.3 (1.9)
PLA-Boosting	90.6 (3.7)	**VAE**	82.7 (2.2)
PLA-RF	90.2 (3.9)	**SIFT**	81.6 (2.7)
PLA-KNN	87.1 (4.2)	**ZM**	88.6 (1.4)
PLA-SVM	88.5 (1.4)	**A-CNN**	89.6 (1.2)

TABLE V
COMPARISON OF RECOGNITION RATE (%) FOR THE SEVEN PATTERNS

Method	PLA-NN	P-NN	LA-NN	PLA-Boosting	PLA-RF	PLA-KNN	PLA-SVM
Donut	100	95	90	90	95	100	85
Moon	92	78.7	88	74.7	62.7	54.7	73.3
Reticle	75	55	65	50	50	50	50
Scratch	85	77.2	82.2	84.4	77.8	74.4	75.6
Center	98.4	100	97.6	94.4	94.4	98.4	92.8
Edge	94.8	91.1	95.9	95.9	97.2	93.6	93
Normal	92.6	87.8	89.3	95.2	93	86.3	95.2

data augmentation (A-CNN) [13] is also applied to encode rotation invariance for the comparisons. The dataset is augmented with rotation every 30°. The right side of Table IV shows the performance of these existing features. The results indicate that the proposed method is superior to all the others. Results using the proposed features together with other classification methods shows that the proposed features is at least as good as existing features in the literature. It should be noted that our method can be seen as a CNN-based method with large self-defined filters in the convolutional layers. The training of conventional CNN is done by random assignment of initial filters and the weights in the filters are iteratively updated through backpropagation. But training requires a large dataset and the extracted features may not be optimal. When it comes to specific problems, pre-defined filters could give better results because of the use of domain knowledge. The superiority of our method can be explained by the proper design of the filters. Our method not only considers the special characteristics of wafer map, but also take the advantage of neural network. The extracted features are easily interpretable and understandable. In addition, it is quite effective even on a small dataset. We can see that without the application of neural network, the features could still have the potential in detecting wafer map defect patterns. This indicates that the features extracted from the proposed masks are also of importance in real applications and needs further studies.

IV. Conclusion

In this article, we propose a novel method for wafer map failure pattern detection based on feature extraction using feedforward neural network. Rotation-invariant features are extracted using proposed weight masks for classification. The proposed masks that are of the same shape with the wafer map could capture the characteristics of defect patterns. The experimental results show that the proposed method is applicable for real applications. The features developed from the masks are also meaningful for the description of wafer map defect patterns, as they are interpretable by intuition. For further studies, error analysis would help identify optimal group of filters for wafer map failure pattern detection problem. In addition, many other problems can be possibly solved based on the proposed masks. One direct application is to solve object detection problem in the architecture of CNN. For example, small size masks can be applied as filters in the convolutional layers to look into more detailed features, and to solve the problem of complex object recognition problems in other fields.

References

[1] 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. doi: 10.1109/TSM.2015.2497264.

[2] L.-I. Tong, C.-H. Wang, and C.-L. Huang, "Monitoring defects in IC fabrication using a hotelling T_2 control chart," *IEEE Trans. Semicond. Manuf.*, vol. 18, no. 1, pp. 140–147, Feb. 2005. doi: 10.1109/TSM.2004.836659.

[3] B. Kim, Y.-S. Jeong, S. H. Tong, I.-K. Chang, and M.-K. Jeong, "Step-down spatial randomness test for detecting abnormalities in DRAM wafers with multiple spatial maps," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 1, pp. 57–65, Feb. 2016. doi: 10.1109/TSM.2015.2486383.

[4] C.-H. Wang, W. Kuo, and H. Bensmail, "Detection and classification of defect patterns on semiconductor wafers," *IIE Trans.*, vol. 38, no. 12, pp. 1059–1068, 2006. doi: 10.1080/07408170600733236.

[5] J. Y. Hwang and W. Kuo, "Model-based clustering for integrated circuit yield enhancement," *Eur. J. Oper. Res.*, vol. 178, no. 1, pp. 143–153, 2007. doi: 10.1016/j.ejor.2005.11.032.

[6] C.-F. Chien, S.-C. Hsu, and Y.-J. Chen, "A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence," *Int. J. Prod. Res.*, vol. 51, no. 8, pp. 2324–2338, 2013. doi: 10.1080/00207543.2012.737943.

[7] J. H. Lee, S. J. Yu, and S. C. Park, "Design of intelligent data sampling methodology based on data mining," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 637–649, Oct. 2001. doi: 10.1109/70.964664.

[8] C.-H. Wang, S.-J. Wang, and W.-D. Lee, "Automatic identification of spatial defect patterns for semiconductor manufacturing," *Int. J. Prod. Res.*, vol. 44, no. 23, pp. 5169–5185, 2006. doi: 10.1080/02772240600610822.

[9] C.-F. Chien, W.-C. Wang, and J.-C. Cheng, "Data mining for yield enhancement in semiconductor manufacturing and an empirical study," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 192–198, 2007. doi: 10.1016/j.eswa.2006.04.014.

[10] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015. doi: 10.1109/TSM.2014.2364237.

[11] B. Kim, Y.-S. Jeong, S. H. Tong, I.-K. Chang, and M.-K. Jeongyoung, "A regularized singular value decomposition-based approach for failure pattern classification on fail bit map in a DRAM wafer," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 41–49, Feb. 2015. doi: 10.1109/TSM.2014.2388192.

[12] F. Adly, P. D. Yoo, S. Muhaidat, Y. Al-Hammadi, U. Lee, and M. Ismail, "Randomized general regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 2, pp. 145–152, May 2015. doi: 10.1109/TSM.2015.2405252.

[13] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 3, pp. 395–402, Aug. 2018. doi: 10.1109/TSM.2018.2841416.

[14] T. T. dos Santos and R. Kern, "Understanding wafer patterns in semiconductor production with variational auto-encoders," in *Proc. 26th Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn. (ESANN)*, 2018. [Online]. Available: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-41.pdf

[15] P. Tulala, H. Mahyar, E. K. Ghalebi, and R. Grosu, "Unsupervised wafermap patterns clustering via variational autoencoders," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–8. doi: 10.1109/IJCNN.2018.8489422.

[16] M. P.-L. Ooi, H. K. Sok, Y. C. Kuang, S. Demidenko, and C. Chan, "Defect cluster recognition system for fabricated semiconductor wafers," *Eng. Appl. Artif. Intell.*, vol. 26, no. 3, pp. 1029–1043, 2013. doi: 10.1016/j.engappai.2012.03.016.

[17] M. Piao, C. H. Jin, J. Y. Lee, and J.-Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 250–257, May 2018. doi: 10.1109/TSM.2018.2806931.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

[19] M.-K. Cheon, W.-J. Lee, C.-H. Hyun, and M. Park, "Rotation invariant histogram of oriented gradients," *Int. J. Fuzzy Logic Intell. Syst.*, vol. 11, no. 4, pp. 293–298, 2011. doi: 10.5391/IJFIS.2011.11.4.293.

[20] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 489–497, May 1990. doi: 10.1109/34.55109.

[21] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008. doi: 10.1016/j.cviu.2007.09.014.

[22] M. Aly, "Survey on multiclass classification methods," *Neural Netw*, vol. 19, pp. 1–9, Nov. 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.423.5993

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html