Received: 17 July 2019 | Revised: 25 October 2019 | Accepted: 30 December 2019

DOI: 10.1002/qre.2627

RESEARCH ARTICLE

WILEY

Defect pattern recognition on wafers using convolutional neural networks

Rui Wang | **Nan Chen**

Department of Industrial Systems Engineering & Management, National University of Singapore, Singapore

Correspondence
Nan Chen, Department of Industrial Systems Engineering & Management, National University of Singapore, Singapore.
Email: isecn@nus.edu.sg

Abstract

In semiconductor manufacturing, wafer testing is performed to ensure the performance of each product after wafer fabrication. The wafer map is used to visualize the color-coded wafer test results based on the locations. The defects on the wafer map may be randomly distributed or form clustered patterns. The various clustered defect patterns are usually caused by assignable faults. The identification of the patterns is thus important to provide valuable hints for the root causes diagnosis. Solving the problems helps improve the manufacturing processes and reduce costs. In this study, we present a novel convolutional neural network (CNN)–based method to automatically recognize the defect pattern on wafer maps. Our method uses polar mapping before the training of CNN to transform the circular wafer map into a matrix, which can be processed within CNN architecture. This procedure also reduces the input size and solves variations in wafer sizes and die sizes. To eliminate the effects of rotation, we apply data augmentation in the training of CNN. Experiments using the real-world dataset prove the effectiveness and superiority of our method.

KEYWORDS

convolutional neural network, pattern recognition, polar mapping, semiconductor manufacturing, wafer map

1 | INTRODUCTION

Semiconductor industry has been growing tremendously in recent decades. Nowadays, the application of semiconductor has expanded to almost every electronic device. With the technological advances, the designs of semiconductor are becoming more and more complex, and the manufacturing process is thus lengthened and requires high accuracy. In the complex chip fabrication process, hundreds of steps are needed before finalizing the design and approving its functionality.[1] The defects can occur at any step. They are usually caused by human mistakes, particles from equipment, chemical stains, etc. The defects can be visualized using wafer maps, which may have some specific patterns on the wafer. Typical spatial patterns include, for example, rings, scratches, and semicircles. These spatial patterns contain valuable information on possible problems occurred during manufacturing processes. For instance, the edge ring occurs during the etching process, and the linear scratch is caused by the machine handling. Figure 1 gives examples of typical defect patterns on the wafer map. If we are able to detect the wafer map patterns, we can then identify the possible root causes of the problems. The elimination of the identified problems can thus improve yield and cut down on production costs. Hence, methods for detecting and recognizing wafer map spatial patterns are highly desirable.

Defect pattern recognition on wafer map has received an increasing attention from semiconductor industry. In most cases, visual inspection and identification of defect patterns are conducted by human experts. However, this human

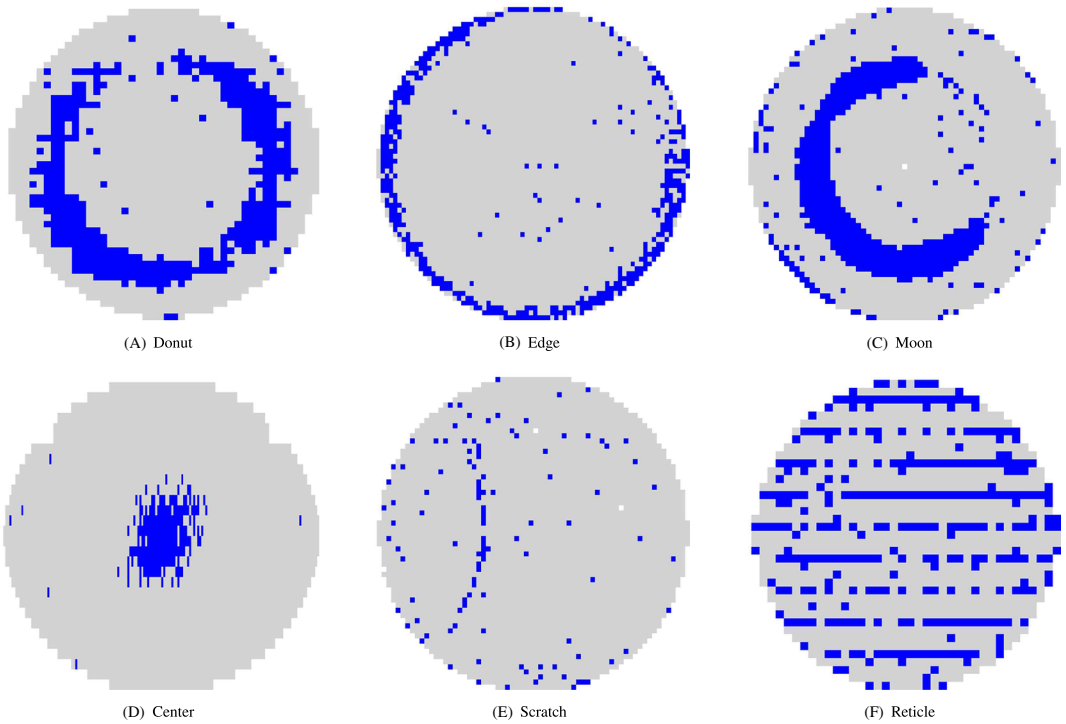


FIGURE 1 Typical examples of wafer map failure patterns [Colour figure can be viewed at wileyonlinelibrary.com]

inspection procedure is time consuming and highly subjective. Moreover, the cost is high using human operators. As a result, semiconductor manufacturing processes may seek the help of automated defect detection techniques, which are capable of quickly and accurately analyzing the data and finding out the corresponding root causes. Machine learning–based methods have been widely used to tackle the automatic pattern detection problem.

Unsupervised learning methods can be applied to detect wafer map defect patterns. This category of methods has the advantage that learning can be accomplished when dataset is large and the number of patterns cannot be identified beforehand. In this category, unsupervised learning neural networks seem to be the most widely used methods. Liu et al[2] and Hsu and Chien[3] implemented adaptive resonance theory network 1 (ART1) method to construct clusters of wafer maps. Choi et al[4] further improved the ART1 algorithm by proposing a multi-step ART1 model with a new similarity measure. Results show that this algorithm provides high accuracy and is quite robust in defect pattern recognition tasks. Lee et al[5] developed a self-organized map (SOM)–based data sampling method to extract the spatial defect features and then cluster the chip locations. Di et al[6] compared ART1 and SOM with more extensive simulated and real datasets. Tulala et al[7] used the variational autoencoder (VAE) to learn latent data representations of the wafer map for clustering analysis.

When the class labels are known, supervised learning is an ideal choice to achieve better recognition accuracy. Support vector machine (SVM) is one of the most widely used approaches to recognize defect patterns. SVM can efficiently perform nonlinear classification incorporating kernel functions. Xie et al[8] proposed a multiclass SVM method to detect rotation-, shift-, and scale-variant patterns. Wu et al[9] extracted a set of rotation- and scale-invariant features and then use SVM to do classifications. Other classification methods can be applied as well. For example, Ooi et al[10] generated a boosting on the alternating decision tree (ADTree) classifier to recognize defect clusters on wafers. Adly et al[11] combined a general regression neural network (GRNN) with a randomization technique. Yu and Lu[12] put forward a joint local and nonlocal linear discriminant analysis (JLNDA) method to discover manifold information and incorporated KNN to do defect pattern recognition. Ensemble classification methods were also used to overcome the limitation of individual classifiers. Piao et al[13] proposed a decision tree ensemble–based approach to aggregate the discrimination power for different feature types. Saqlain et al[14] combined the results of four machine learning classifiers with a soft voting ensemble (SVE) technique, which outperformed regular machine learning-based classifiers.

Recently, convolutional neural networks (CNN)[15] have proven to be a very powerful and successful machine learning approach for object detection.[16] It has the ability to automatically extract features from complex data and conduct classification tasks. The structure mainly consists of several convolutional layers for feature extraction, followed by pooling operations to combine the outputs of neuron clusters and discard irrelevant details. There are also fully connected layers

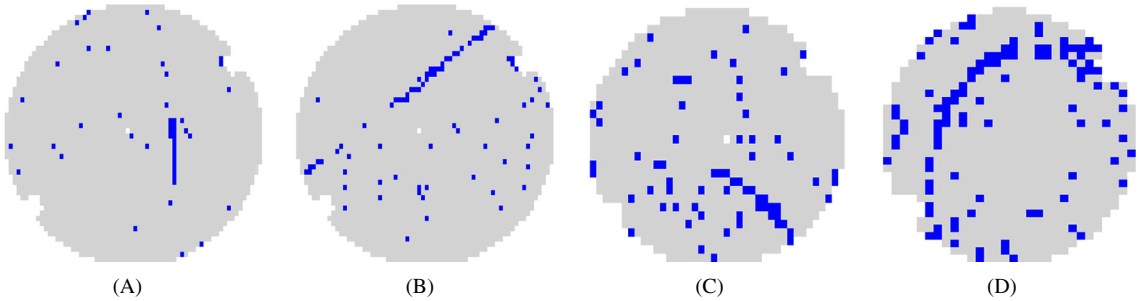


FIGURE 2 Examples of different die sizes, pattern sizes, and orientations for "Scratch" pattern type [Colour figure can be viewed at wileyonlinelibrary.com]

before the outputs, where each neuron is connected to all the neurons in the previous layer. There exists some challenges though, to apply CNN to the wafer map pattern recognition problem. First, the wafer map does not have a square shape, but rather forms a circular area. However, the input of conventional CNN needs to be a rectangle matrix. Second, the wafer map contains randomly distributed defects, which are seen as "noise." They may account for 10% to 50% of defects[10] on the wafer map. Third, the size of dies on the wafer map may be different, and the wafer size may also vary, which adds difficulty in applying classical algorithms. Last, we would like the method to achieve rotation invariance, ie, to give the same classification results for the same type of defect patterns with different orientations. The proposed method should be able to overcome these challenges and give correct recognitions.

Figure 2 gives an example of the "Scratch" pattern from real-world data. The blue pixels show locations of defective dies, and the gray pixels are normal dies on the wafer. In the following, we use "die" and "pixel" interchangeably in this article. For the "Scratch" pattern, both straight lines and arcs are considered. There exists random defective points in the figure that are not part of the pattern and not helpful for defect recognition. Thus, these defects are seen as noise in the dataset. They are usually caused by clean room environment problems. It is also observed that the size, orientation, and location of the patterns are quite different. In addition, the dies on the wafers have different sizes, which is represented by the pixel sizes in the wafer map. Obviously, dies on Figure 2C,D have larger size.

Existing studies have utilized CNN to detect defect pattern on wafers.[17-19] Despite the outstanding performance of these CNN-based methods, they still suffer from various limitations. For example, the high computational complexity is one of the primary concerns when handling large datasets. In these studies, wafer maps are directly processed with a CNN architecture. However, the direct application of CNN will cause the waste of computation resources. This is because the input of CNN needs to be a square matrix, but only the round part of the wafer map is actually used. Also, the size of wafer is growing larger for high production efficiency, increasing the training complexity. To further improve the performance and efficiency of CNN for the recognition of wafer map defect patterns, we propose a novel CNN-based method to transform the circular shape into a matrix, which reduces the input image size while keeping the patterns distinguishable. We combine the idea of polar coordinates mapping together with CNN to achieve rotation equivariance, which also overcomes the variation problems in pattern sizes and orientations. In the first step, we aim to reduce the effects of noise using a density-based clustering method called DBSCAN (density-based spatial clustering of applications with noise),[20] which tries to extend the cluster based on the density in the neighborhood. The dies that do not belong to any of the clusters are regarded as noise. We then transform the denoised circular wafer map into a matrix format, which is processable by CNN using the polar mapping technique. Polar mapping solves the problem of image size variation and rounded shape of wafers, while reducing the dimension of data. This approach also enables us to produce equivariant representations in rotation in the architecture of CNN. Data augmentation is applied to improve the performance of the CNN method. Experiments show that our approach can achieve a satisfactory performance even with a small input dataset. In addition, the shallow architecture of our proposed CNN reduces the computation costs. Results show that our method can be well applied in the wafer map defect pattern recognition problem.

The rest of this article is organized as follows. Section 2 describes the detailed procedure of how to utilize CNN with polar mapping in the context of wafer map failure patterns. To be specific, Section 2.1 explains the application of DBSCAN to filter out random defects. Then in Section 2.2, we propose the polar mapping method to effectively transform the circular wafer map into a rectangle matrix, while reserving the pattern information and reducing input size. Section 2.3 introduces the architecture of CNN and its linkage to the polar mapping method. Section 2.4 provides details for the application

of CNN with data augmentation in our problem. Section 3 gives experimental results with an application in real data. Section 4 concludes the article.

2 | METHODOLOGY

In this section, we explain the framework of the proposed CNN model to identify wafer map failure patterns. In the preprocessing step, the noise pixels in the wafer maps are removed to better recognize the defect patterns. Circular wafer maps are then transformed into a matrix format that can be processed within conventional CNN environments. Then we use the transformed matrix as input to the CNN model for the classification task. Data augmentation is applied to improve the robustness of our method to rotation changes.

2.1 | Denoising in wafer maps

Random noise on the wafer maps may seriously affect the recognition accuracy. Noise here refers to isolated defects that appear randomly and are not part of the pattern cluster. Denoising is commonly a key step in image preprocessing. After denoising, patterns are preserved while isolated defect points are removed. Current denoising methods include median filter[12] and spatial filter,[21] which update the die value with a function of the values of surrounding dies. These methods have shown good performance in many applications. However, they can hardly differentiate long and thin patterns like scratches from random defects. After filtering, scratches may be removed because the number of failures is small in the neighborhood. For this problem, we are more interested in connected pixels. To achieve the goal, we use DBSCAN, which builds connected dense components when clustering. DBSCAN is capable of forming clusters of arbitrary shapes while filtering out outliers that lie far from the clusters.

DBSCAN[20] basically groups together objects that are closely linked and forms a dense region. The application of DBSCAN takes two threshold parameters, radius ϵ and minimum number of points m. Let D be a dataset of points and p be a point, the ϵ-neighborhood of p is defined as $N_\epsilon(p) = \{q \in D | d(p,q) \leq \epsilon\}$. If there are at least m of points in an ϵ-neighborhood of p, ie, $|N_\epsilon(p)| \geq m$, then p is called the core point. In DBSCAN, it is required that for every point p in the cluster C, there is a point q in C such that p is inside the ϵ-neighborhood of q and q is a core point. This property is called point p directly density-reachable to point q, if $p \in N_\epsilon(q)$ and $|N_\epsilon(q)| \geq m$. A canonical extension is density-reachability. A point p is density-reachable from a point q if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i. Density-reachability cannot cover the relation of border points, because they do not satisfy core point condition. Thus, a further extension named density-connectivity is introduced. A point p is density-connected to a point q if there is a point o such that both p and q are density-reachable from o. A cluster is defined to be a set of density-connected points, which is maximal with respect to density-reachability. Noise is the set of points in D not belonging to any of the clusters.

In the application, the DBSCAN algorithm starts with a random point p in D and determine the ϵ-neighborhood. If core point condition is satisfied, it recursively collects density-reachable points in the neighborhood of p until no new points can be added to the cluster. DBSCAN repeatedly conduct the previous steps for the points in D until all the points are explored. Noise points are labeled if they are isolated points that do not belong to the clusters. The application of DBSCAN do not require the number of clusters for the clustering task; instead, only two parameters (ϵ and m) are needed. Besides, connected points are reserved for clusters with the density-connectivity rule. The noise points are also removed by DBSCAN, achieving the robustness to outliers.

In our study, DBSCAN is a suitable method for denoising. The isolated defective dies are removed while loosely connected dies are reserved based on the selection of parameters (ϵ, m). By setting the parameters, it is possible to optimally adjust the degree of denoising based on the performance. $\epsilon = 2, m = 3$ guarantees the random defects being removed while connected patterns standing out after denoising. In Jin et al,[22] $\epsilon = \sqrt{2}, m = 3$ is chosen to remove isolated outliers and twin outliers in their study. Considering the disconnected pattern "Reticle," we set the neighborhood threshold ϵ to 2, which ensures loosely connected patterns can be reserved after denoising. Figure 3 illustrates an example of the pattern "Edge." We can observe that after the denoising step, nonclustered defective pixels are largely removed while the failure pattern is reserved on the image.

2.2 | Polar mapping

The wafer map of circular shape is hard to be processed directly using conventional CNN, which requires a matrix format for the input. Our solution to this problem is to transform the original wafer patterns to a matrix according to the

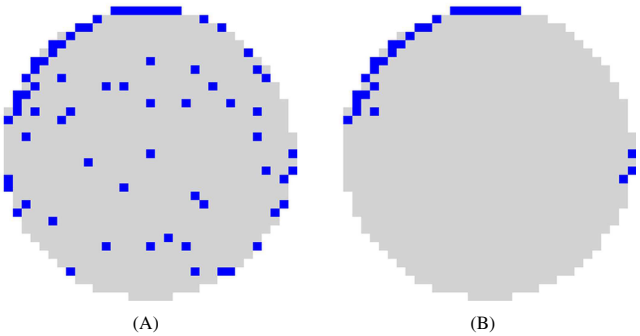


FIGURE 3 An example of denoising for the pattern "Edge": A, original; B, denoised [Colour figure can be viewed at wileyonlinelibrary.com]

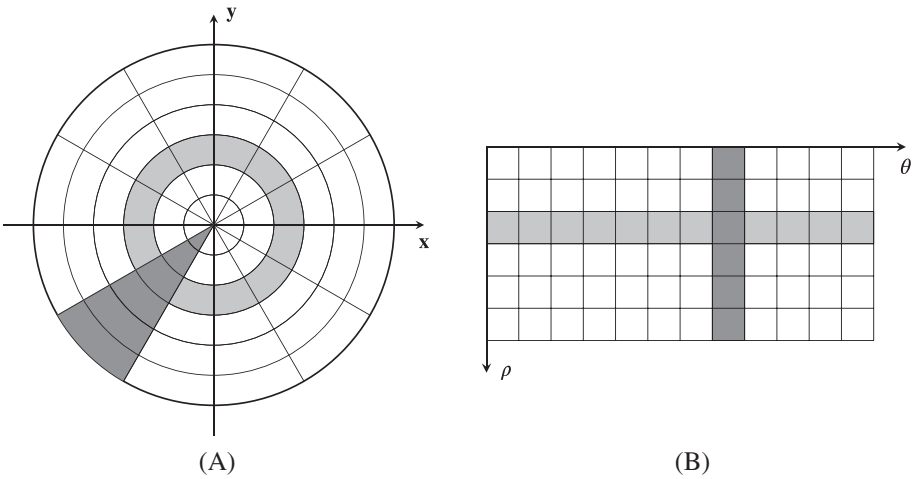


FIGURE 4 Schematic example of polar mapping

polar mapping rule. The point $P(x, y)$ in the Cartesian coordinate system will correspond to $P(\rho, \theta)$ in the polar coordinate system. ρ here represents the distance from the centroid, and θ is the counterclockwise angle from the x-axis. The coordinates ρ and θ are transformed by

$$\begin{cases} \rho = \sqrt{x^2 + y^2} \\ \theta = \arctan(y/x) \end{cases} \tag{1}$$

Thus, the location of every point (x, y) in the original wafer map can be represented by a pair of (ρ, θ).

The variation in wafer map sizes and pixel sizes also causes problems for CNN input. We can then apply a simple sampling structure on the polar coordinates to transform the circular wafer map into a rectangular matrix of the same size. For θ, we simply separate it into N_θ equidistant intervals, ie, $\theta = (0, 2\pi/N_\theta, \ \dots\ , 2\pi(N_\theta - 1)/N_\theta, 2\pi)$, and $\theta_0 = 0, \theta_{N_\theta} = 2\pi$. We also divide radius into intervals with $\rho = (\rho_0, \rho_1, \ \dots\ , \rho_{N_\rho})$, and $\rho_0 < \rho_1 < \ \dots\ < \rho_{N_\rho}$, with $\rho_0 = 0, \rho_{N_\rho} = R$, R is the radius of the wafer map. Then each bin in the polar coordinates system corresponds to a cell in the matrix of size $N_\theta \times N_\rho$. The corresponding value is computed by the average of pixel values (1 for defective pixels, 0 for normal pixels) inside each bin. If we denote the transformed matrix as T, the element value t_{ij} is computed by the percentage of failure pixels inside bin $(\theta_{i-1} \le \theta < \theta_i, \rho_{j-1} \le \rho < \rho_j), i = 1, \ \dots\ , N_\theta; j = 1, \ \dots\ , N_\rho$.

Figure 4 shows the application of polar mapping to the wafer maps with $N_\theta = 12, N_\rho = 6$. Following this transformation, the light gray ring on the wafer surface is mapped into one row, and the dark gray area is mapped into one column in the matrix. Polar mapping has some advantages. This mapping method is intuitively ideal for circular areas to be transformed into a rectangular shape. The transformed matrix with a rectangular shape can then be processed in the CNN architecture. Local information is also preserved, as regular patterns like concentric circles and radial lines result in similarly regular patterns after mapping. Besides, the resolution and size differences of images are solved by setting proportional binning intervals of the wafer radius. The transformed matrix is constant in size with respect to N_θ and N_ρ, which would help reduce the input size if properly set. Smaller size of input data shortens the computation time of the classification task. Also, the rotation in the Cartesian coordinates is equivalent to the translation along the angular axis in the polar coordinates. This means if the pattern rotates around the origin, its polar mapping only moves horizontally.

Figure 5 shows the polar mapping of each pattern given in Figure 1 with $N_\theta = 16, N_\rho = 10$. The mapping computes the percentage of defects within each binning area. Thus, in Figure 5, darker color means higher proportion of detects. It can be observed that polar mapping effectively differentiates among different patterns. After polar mapping, circular patterns,

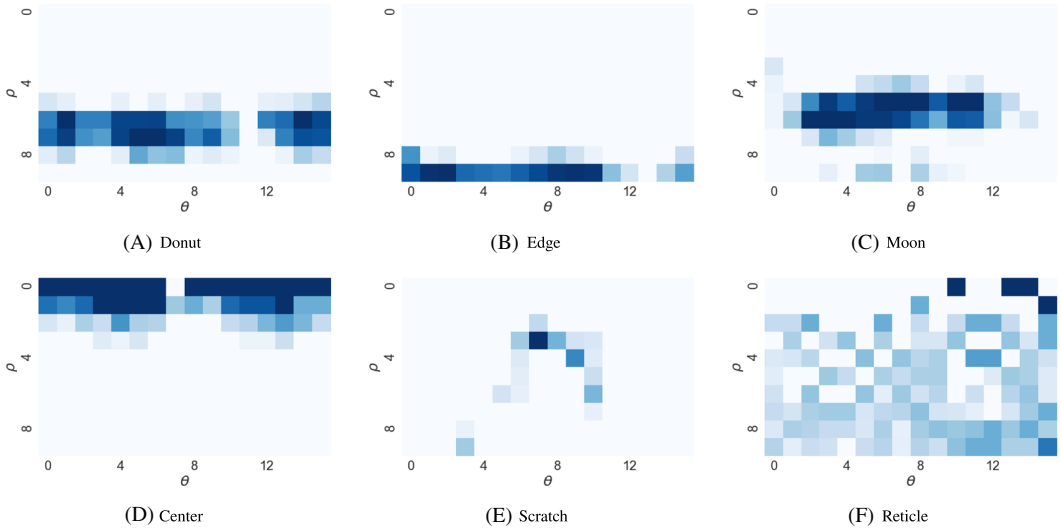


FIGURE 5 Polar mapping results of wafer map failure patterns [Colour figure can be viewed at wileyonlinelibrary.com]

ie, "Donut," "Moon," "Center," and "Edge" become horizontal strips along θ-axis, while "Scratch" changes into a curve in the transformed image. The "Reticle" pattern is converted into a sparsely distributed pattern, similar to the "Reticle" pattern in the original wafer map.

It should be noted that the binning along radial axis can take unevenly distributed intervals based on real applications. The intervals at the middle could take larger values to ensure enough number of dies within each bin. Besides, the lengths of intervals are also adjustable based on the locations of different patterns. For example, the pattern "Edge" is often located in the radius interval $[0.75R, R]$, and the corresponding binning intervals could be adjusted accordingly. Similarly, instead of the direct binning method, other space variant sampling structures[23] can be considered.

2.3 | Convolutional neural networks

CNN has exhibited strong performance in image classification tasks. As one of the machine learning methods, CNN has the ability to automatically learn important features from images. It is designed to take advantage of the two-dimensional structure of an input image based on the idea of neural networks. CNNs are obtained by stacking several layers of feature-detecting neurons to learn some characteristics of the input data. These layers are, namely, convolutional layer, pooling layer, and fully connected layer. CNNs vary in the configuration of layers and the training methods for different problems.

1. Convolutional layer: Convolution is done by sliding the filter along the image to compute the summation of the element-wise multiplications of the filter and the covered square of the image, while preserving the spatial relationship between pixels. The input to the convolutional layer is a three-dimensional array $x \in \mathbb{R}^{m\times n\times c}$, where each dimension represents the height, width, and the number of channels of the image. Since a wafer map only contains binary values, we limit the input to the two-dimensional single channel case $x \in \mathbb{R}^{m\times n}$ for simplicity. Suppose the input to a convolutional layer has size $m \times n$, if we use $k \times k$ filter $w \in \mathbb{R}^{k\times k}$ for convolution, the output should be of size $(m-k+1)\times(n-k+1)$. The computation in the convolutional layer can be formulated as

$$y_{ij} = f\left(\sum_{a=0}^{k-1}\sum_{b=0}^{k-1} w_{ab}x_{i+a,j+b} + l\right), \tag{2}$$

where $f(\cdot)$ denotes a nonlinear function applied to each component of the convolved feature map. One commonly chosen function is the rectified linear unit (ReLU)[24]:

$$f(x) = \max(0, x). \tag{3}$$

2. Pooling layer: The pooling layer aims to reduce the dimension of features while reserving important information. Features after pooling is robust against noise. The pooling layer subsamples the output feature from the previous layer using a small rectangle block. In most cases, the feature is divided into non-overlapping regions based on the size of the block, and one value is returned for each region. In this work, we consider max pooling in the CNN structure, and then the maximum response is returned in each $k \times k$ region, which can be written as

$$y_{ij} = \max_{\substack{0<a\leq k-1,\\ 0<b\leq k-1}} x_{i+a,j+b}. \tag{4}$$

3. Fully-connected layer: After several convolutional and max pooling layers, fully connected layers are often used as the final layers of a CNN. A fully connected layer takes all neurons in the previous layer and connects them to every neuron on the current layer. The output of the previous layers is flattened to a 1D feature vector as input to the fully connected layers. For classification, softmax function is used to perform multiclass logistic regression and squashes the outputs of each neuron to be between 0 and 1. The output of the softmax function indicates the probability that any of the classes are true. If we have a K-class classification problem, denote $z = (z_1, \ldots, z_K)$ as the vector of input to final output layer, we have

$$f(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, i = 1, 2, \ldots, K. \tag{5}$$

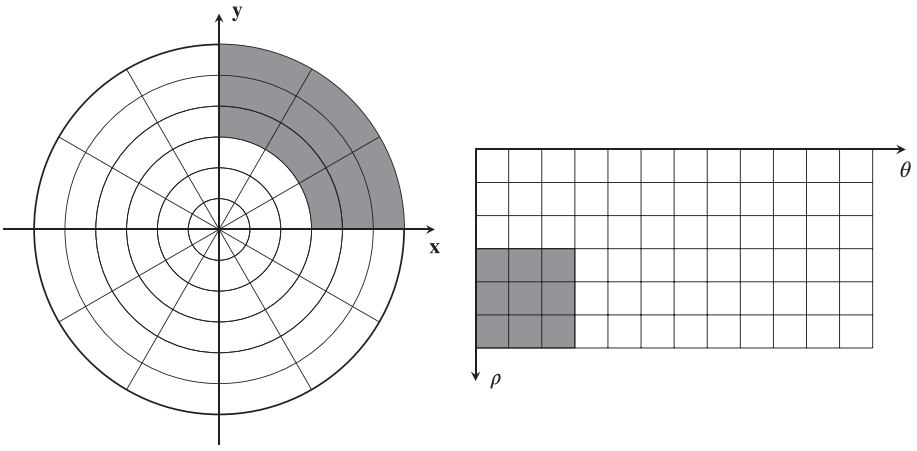
Dropout[25] can be applied to prevent overfitting in fully connected layers. The parameters to be estimated are filters in convolutional layers and weights connecting adjacent layers. In this study, we train the parameters of the CNN with stochastic gradient descent and back-propagation.

We use the polar mapping results as the input for CNN. Then the rectangular filter of the input matrix for convolution is equivalent to the filter with a shape of angular sector inside an annulus; see Figure 6. In this figure, the gray area shows the size of a filter. To slide the filter horizontally in the polar-mapped matrix is the same as rotating the augular-shape filter in the wafer map. Similarly, moving the rectangular filter in the vertical way means sliding the corresponding filter radially in the original image.

One advantage of using polar mapping matrix as input to CNN is the equivariance property to rotation. Equivariance is defined in the following way. First, let G denote a group of transformations. For $t \in G$, $L_t I$ denote the result of transformation on an image I. A mapping Ψ is equivariant to group G if

$$\Psi(L_t I) = L'_t(\Psi(I)). \tag{6}$$

Equivariance means there exists a relationship between transformation L_t of input I and transformation L'_t of features. If L'_t is the same as L_t, equivariance means features are transformed the same way as the input. When L'_t is the identity function, equivariance becomes invariance $\psi(I)$. Equivariance is often preferred as it enables the prediction of the response features given a transformation of the input. After polar mapping, the rotation of the original wafer map is sim-



(A) (B)

FIGURE 6 Equivalent convolution operation in A, original wafer map and B, matrix after polar mapping

ply obtained by the translation of the transformed matrix along the θ-axis. We show that convolutions are inherently equivariant to translation in the following.

Let $f(h)$ and $\psi(h)$ be continuous functions over a bounded 2D region $\Omega \subset \mathbb{Z}^2$, ie, $f, \psi : \Omega \to \mathbb{R}$. The convolution of feature map f and filter ψ can be written as

$$(f \star \psi)(x) = \int_h f(h)\psi(h-x)dh, \tag{7}$$

where $x \in \Omega$ are 2D vectors of the indexes. For a transformation t acting on a set of feature maps, we introduce the following notation:

$$[L_t f](x) = [f \circ t^{-1}](x) = f(t^{-1}x). \tag{8}$$

This relationship also holds if we replace x by an element h in the region. When t represents a pure translation, then $t^{-1}h$ simply indicates $h - t$. It can be proven that convolution is translation equivariant by definition:

$$\begin{aligned}
[[L_t f] \star \psi](x) &= \sum_h f(t^{-1}h)\psi(h-x) \\
&= \sum_h f(h-t)\psi(h-x) \\
&= \sum_h f(h)\psi(h-(x-t)) \\
&= (f \star \psi)(x-t) \\
&= [L_t[f \star \psi]](x).
\end{aligned} \tag{9}$$

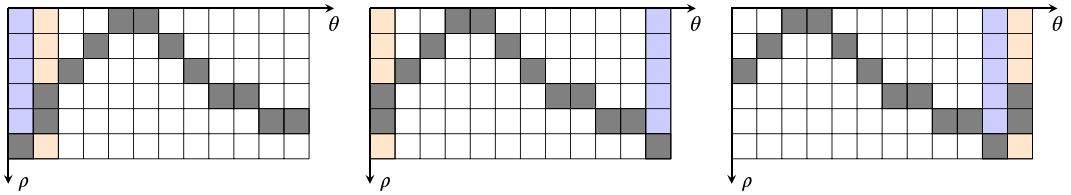
This is obtained by using the substitution $h \to h+t$. So here, we have $[L_t f] \star \psi = L_t[f \star \psi]$, which means a translation followed by a convolution is the same as a convolution followed by a translation. Rotation becomes translation in the polar mapping matrix, and CNN is inherently equivariance to translation. Therefore, polar mapping guarantees equivariance to rotation in CNN. It should be noted that in the special case of binning following the log-polar coordinates, CNN also guarantees scale equivariance as scaling of the image becomes traslation along ρ-axis.

2.4 | Training CNN with data augmentation

For wafer defect pattern recognition, one of the problems is that defect patterns can occur at different orientations on the wafer. Wafers have a round shape, which makes it hard to define the image orientation. The defects can be located at any spots on a wafer and should be treated as the same pattern if they form the same shape, typical examples are "Moon" and "Scratch" patterns. Ideally, CNN could give us the correct classification results for occurrence of pattern at every possible direction. However, the data used for training are always relatively scarce if we would like to take into consideration all the orientations of the patterns.

In order to improve the robustness of our method, we apply data augmentation operations in the training of CNN. Data augmentation creates plausible transformations of existing samples as additional training data, which aims to improve the performance of classifiers. By training on augmented data, the network learns more information about the transformations of the same pattern, which in turn improves accuracy for recognition. In this study, we create several rotated copies for each wafer sample to relieve the effects of rotation in pattern classification task. The rotation angle α is sampled from 0 to 2π with intervals of equal distance. If we denote the number of rotated copies created for one master image to be N_r, then the set of rotation angles is $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_{N_r}\}$, where $\alpha_1 = 0, \alpha_2 = 2\pi/N_r, \dots, \alpha_{N_r} = 2(N_r-1)\pi/N_r$. This step eliminates the influence of orientations of input images and clustered patterns. To make it simple, we choose N_r to be a divisor of N_θ, ie, $N_\theta = cN_r, c \in \mathbb{N}^+$.

As we already transformed data into matrix by polar mapping, rotation becomes quite simple and convenient in this case. The rotation of the wafer map reduces to the rolling of the transformed matrix along the θ-axis. In the previous example, the rotation of the sample by 30° can be accomplished by moving the leftmost column to the right side; see Figure 7. Similarly, the 60° rotation of the original image can be simply done by cutting the left two columns and then pasting to the rightmost. This rotation operation comes with almost no extra computation cost, which is obviously superior to conventional augmentation methods.



(A) (B) (C)

FIGURE 7 Rotation with polar mapping matrix A, before rotation, B, rotation by 30°, and C, rotation by 60° [Colour figure can be viewed at wileyonlinelibrary.com]

To evaluate the effectiveness of the proposed method, stratified k-fold cross-validation is applied to measure the performance of the model. The folds are selected so that each fold contains roughly the same proportions of class labels. This ensures that each fold is representative of all classes of the data.

3 | EXPERIMENT AND RESULTS

In this section, we apply the proposed method to a real wafer map dataset, and compare with several different methods.

3.1 | Data description

The wafer map dataset was collected from a semiconductor company. It contains 904 wafer maps, including 218 normal wafer samples and 686 defective wafer samples. Information like die positions and wafer testing results are included in the dataset. The defective wafers have six patterns, ie, "Donut," "Moon," "Reticle," "Edge," "Scratch," and "Center"; see Figure 1. On each defective wafer map, there exists both random defects and systematic defects. The number of dies in each wafer varies a lot, from hundreds to over six thousand, which means this dataset is a mixture of wafer maps of different products.

3.2 | Performance analysis

We evaluate the proposed framework for the wafer defect pattern recognition. First, we remove the noise of the wafer map by DBSCAN with $\epsilon = 2, m = 3$. Next, to apply the CNN model, the parameters to be determined are N_θ, N_ρ, and N_r. We set $N_\theta = 16, N_\rho = 10$ in the polar mapping step, which means wafer maps are transformed into 16×10 matrices. For data augmentation, N_r is chosen to be a divisor of N_θ for simplicity, ie, 2, 4, 8, and 16 in this case. Here, the number of rotated versions is chosen to be $N_r = 8$; the wafer map is rotated every $\pi/4$ rad for augmentation.

We use a small network with two convolutional layers for our problem. The model structure is introduced in Table 1. Note that the input image only takes binary values; thus, the input channel is 1 instead of 3 for colorful images. We choose to use two convolutional layers in the structure, the first one learns 32 feature maps using 3×3 filters, while the

TABLE 1 Architecture of CNN model used in this study

Layer	Size
conv1	3×3, 32
conv2	5×5, 32
pool1	2×2
fc1	400
fc2	400

TABLE 2 Recognition rate (%) for the seven patterns

Defect Pattern	Center	Moon	Edge	Scratch	Donut	Reticle	Normal
Center	99	0	0	1	0	0	0
Moon	0.1	94.6	3.3	1.9	0	0	0.1
Edge	0	0.6	96	1.4	0	0	2
Scratch	0	4.9	9	74.1	0	3.5	8.5
Donut	0	0	0	0	100	0	0
Reticle	0	0	6.7	23.3	0	70	0
Normal	0	0	6.7	5	0	0	88.3

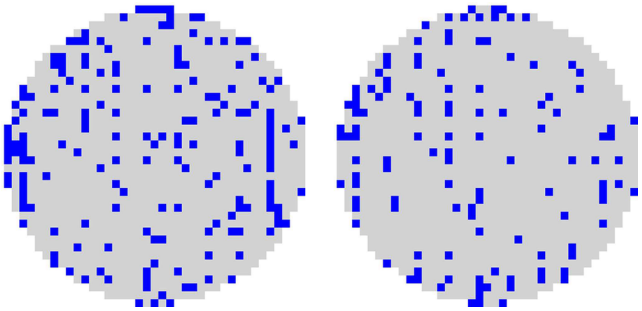


FIGURE 8 Examples of "Reticle" patterns wrongly detected as "Scratch" [Colour figure can be viewed at wileyonlinelibrary.com]

next layer creates the same number of feature maps but using 5×5 filters. We add a padding of zeros at the edges to maintain the same size of the output matrix as the input matrix, which avoids the loss of information at the wafer map edges during the convolution operation. The following max-pooling layer reduces feature dimension with subsampling region 2×2. The ReLU activation function, proving to be free from the problem of gradient divergence, is investigated instead of sigmoid active function for convolutional and fully connected layers. We use two fully connected layers (both with 400 hidden units) in our experiment. To relieve the problem of overfitting, dropout is used in the fully connected layer. Fivefold cross-validation is applied to measure the performance of the model.

Table 2 shows the combined confusion matrix for the proposed method with overall accuracy 90.0%. In the table, the annotations in the first row represent predicted patterns, and the first column shows original patterns. The diagonal values represent the accurate recognition rates of each pattern. The lowest recognition rate is 70% for "Reticle," and it is frequently confused with "Scratch," which can be explained by intuition; see Figure 8. We effectively solve the problem with a relatively shallow network, while the size of input is also reduced with the selection of $N_\theta = 16$ and $N_\rho = 10$.

To gain further insight into the influence of polar mapping on the performance of pattern recognition, we present the difference in accuracy using different parameters for polar mapping matrix size $N_\theta \times N_\rho$ in Table 3. The results shown here is by training on the polar transformed dataset directly to eliminate the influence of augmentation. The column "Accuracy" shows the mean and standard deviation of the accuracy of each fold in cross-validation, with the standard deviation shown in the parentheses after the mean. The column "CV" represents the coefficent of variation, which is defined as the ratio of the standard deviation to the mean. Intuitively, larger value of N_θ and N_ρ should give better results, which provides more information of the patterns on the wafer map. Results indicate that the size of 16×10 gives relatively good performance and the increase in both values cannot significantly improve the accuracy in terms of both mean and variation. Thus, 16×10 is a good choice for matrix size of polar mapping in this study. It effectively reduces input size for CNN, which may contains thousands of dies on the wafer map, and it in turn results in network size reduction and learning simplification.

3.3 | Comparisons

We compare our method with other machine learning methods, including support vector machine (SVM), multilayer perceptron (MLP), probabilistic neural network (PNN), and generalized regression neural network (GRNN). The training of these methods share the same procedure with our method, with polar mapping and data augmentation. For the SVM, we use the radial basis function kernel with C-support vector classification algorithm ($C = 2.0$). For the MLP, we assume two hidden layers with 200 hidden units in each layer. For activation functions, we use the ReLu for hidden layers and

TABLE 3 Accuracy comparison for different matrix sizes after polar mapping

Size	Accuracy, %	CV, %
32×20	89.7 (1.0)	1.1
16×20	89.1 (1.9)	2.1
8×20	75.7 (2.9)	3.8
32×10	89.4 (1.0)	1.1
16×10	89.2 (1.3)	1.5
8×10	83.7 (2.2)	2.6
32×5	65.9 (2.6)	4.0
16×5	65.4 (1.2)	1.8
8×5	62.0 (3.1)	5.0

TABLE 4 Accuracy comparison for different machine learning methods

Method	Accuracy, %	CV, %
CNN	90.0 (1.3)	1.4
SVM	82.6 (2.7)	3.3
MLP	80.7 (3.5)	4.3
GRNN	83.2 (2.1)	2.5
PNN	82.6 (2.8)	3.4

TABLE 5 Accuracy comparison for different CNN-based methods

Method	Accuracy, %	CV, %
A-P	90.0 (1.3)	1.4
A-C	89.6 (1.2)	1.3
CS	88.9 (2.4)	2.7
P	89.2 (1.6)	1.8
C	88.3 (2.4)	2.7

TABLE 6 Comparison of recognition rate (%) for the seven patterns

Method	A-P	A-C	P	C	CS
Center	99.0	99.0	98.0	98.0	98.0
Moon	88.3	72.7	88.0	82.9	87.9
Edge	96.0	94.0	93.2	94.0	92.3
Scratch	74.1	77.7	69.2	64.3	68.6
Donut	100	100	95.0	100	100
Reticle	70.0	80.0	68.3	73.3	73.3
Normal	89.0	90.4	93.1	92.2	94.9

softmax for output layers. For PNN and GRNN, we set standard deviation for the probability density function (PDF) to be both 0.2. Fivefold cross-validation is used for the evaluation of each of the method. The comparison results are illustrated in Table 4. CNN shows the best result in terms of both mean accuracy and the degree of variation.

We also run the dataset on some existing CNN-related methods for comparison.

- Conventional CNN[19] (C): The image size is resized to 30×30 to form a standard size using bilinear interpolation for image processing. The same CNN structure as Table 1 is applied.
- Conventional CNN with data augmentation (A-C): Conventional CNN incorporates the same data augmentation step as our proposed method.
- CNN using circular symmetric filters[26] (CS): Circular symmetric filters are applied in CNN to achieve rotation invariance, which is guaranteed with a constraint in the loss function.
- Polar CNN (P): Our proposed method without data augmentation in training the model. Polar mapping is applied to transform the wafer map into square matrix.
- Polar CNN with data augmentation (A-P): Our proposed method which performs polar mapping at the wafer center and conduct data augmentation during training.

We do not apply data augmentation to CS because the architecture is inherently invariant to rotation. Table 5 shows the comparison between the accuracy of the proposed method (A-P) and that of a A-C, P, C, and CS. The results indicate that the proposed method is superior to all the others. Without data augmentation, P gives comparable results with C and CS, but P successfully reduces input size (16×10) compared with C and CS (30×30). Also, P guarantees rotation equivariance property in the application of CNN. Data augmentation generally slightly improves the classification results in both of our proposed method P and conventional CNN, and the error rate drops at the same time. Accompanied with data augmentation, our method A-P is also superior to A-C, which performs the best among all them. The comparison of recognition rates for each pattern is shown in Table 6. Overall, polar-mapping based method is better at the recognition of "Moon," while conventional CNN gives better performance at the recognition of pattern "Reticle." Compared with P-CNN, data augmentation helps improves performance in all patterns especially "Reticle" and "Scratch."

4 | CONCLUSION

In this article, we propose a novel method for wafer map failure pattern detection in the framework of CNN. DBSCAN is first adopted for the removal of random defects. Compared with existing methods such as the spatial filter, DBSCAN

that builds connected clusters of arbitrary shapes helps keep long and thin patterns while removing isolated defects. Polar mapping is then applied as a preprocessing step to solve the inconvenience of the circular shape of wafers, as well as the pixel size difference. The wafer map is transformed into a matrix, and the patterns are still differentiable after the transform. The use of the transformed matrix as input to CNN achieves the equivariance property to rotation. At the same time, the size of image data has been reduced by the polar mapping, which simplifies the problem and shortens the network training. We also apply data augmentation to eliminate the influence of rotation for the patterns.

The experimental results on real-world data show that the proposed method outperforms the conventional CNN. The results indicate that proper preprocessing of wafer maps helps improve the recognition performance of the defect patterns. Also, computation resources can be saved with the dimension reduction by polar mapping. Obtaining good performance with data of smaller size is a considerable advantage, as the training time is one of the major concerns when using CNN. The limitation of this method lies in the need to preprocess the wafer map data instead of using raw data directly. When the size of raw data is small, the advantage of small dataset weakens considering the tradeoff between training time and preprocessing time. For further studies, our method may be extended to mixed-type defect pattern recognition problems. In addition, polar mapping may also be applicable to other problems under the framework of CNN, when the rotation equivariance property is preferred.

ORCID

Rui Wang https://orcid.org/0000-0001-5958-2234
Nan Chen https://orcid.org/0000-0003-2495-5234

REFERENCES

1. Dong H, Chen N, Wang K. Wafer yield prediction using derived spatial variables. *Qual Reliab Eng Int*. 2017;33(8):2327-2342.
2. Liu SF, Chen FL, Lu WB. Wafer bin map recognition using a neural network approach. *Int J Prod Res*. 2002;40(10):2207-2223.
3. Hsu S-C, Chien C-F. Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. *Int J Prod Econ*. 2007;107(1):88-103.
4. Choi G, Kim S-H, Ha C, Bae SJ. Multi-step ART1 algorithm for recognition of defect patterns on semiconductor wafers. *Int J Prod Res*. 2012;50(12):3274-3287.
5. Lee JH, Yu SJ, Park SC. Design of intelligent data sampling methodology based on data mining. *IEEE Trans Robot Autom*. 2001;17(5):637-649.
6. Di Palma F, De Nicolao G, Miraglia G, Pasquinetti E, Piccinini F. Unsupervised spatial pattern classification of electrical-wafer-sorting maps in semiconductor manufacturing. *Pattern Recogn Lett*. 2005;26(12):1857-1865.
7. Tulala P, Mahyar H, Ghalebi E, Grosu R. Unsupervised wafermap patterns clustering via variational autoencoders. In: 2018 International Joint Conference On Neural Networks (IJCNN); 2018; Rio de Janeiro, Brazil:1-8.
8. Xie L, Huang R, Cao Z. Detection and classification of defect patterns in optical inspection using support vector machines. In: International Conference on Intelligent Computing. Springer; 2013; Nanning, China:376-384.
9. Wu M-J, Jang Jyh-Shing R, Chen J-L. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Trans Semicond Manufact*. 2015;28(1):1-12.
10. Ooi MP-L, Sok HK, Kuang YC, Demidenko S, Chan C. Defect cluster recognition system for fabricated semiconductor wafers. *Eng Appl Artif Intel*. 2013;26(3):1029-1043.
11. Adly F, Alhussein O, Yoo PD, et al. Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps. *IEEE Trans Ind Inform*. 2015;11(6):1267-1276.
12. Yu J, Lu X. Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. *IEEE Trans Semicond Manufact*. 2016;29(1):33-43.
13. Piao M, Jin CH, Lee JY, Byun J-Y. Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features. *IEEE Trans Semicondu Manufact*. 2018;31(2):250-257.
14. Saqlain M, Jargalsaikhan B, Lee JY. A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. *IEEE Trans Semicond Manufact*. 2019;32(2):171-182.
15. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. 1097–1105; 2012.
16. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Cham: Springer; 2014:818-833.
17. Cheon S, Lee H, Kim CO, Lee SH. Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Trans Semicond Manuf*. 2019;32:163-170.
18. Kyeong K, Kim H. Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. *IEEE Trans Semicond Manufact*. 2018;31(3):395-402.
19. Nakazawa T, Kulkarni DV. Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Trans Semicond Manufact*. 2018;31(2):309-314.

20. Ester M, Kriegel H-P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*. 1996;96:226-231. https://doi.org/10.5120/739-1038
21. Wang C-H. Recognition of semiconductor defect patterns using spatial filtering and spectral clustering. *Expert Syst Appl*. 2008;34(3):1914-1923.
22. Jin CH, Na HJ, Piao M, Pok G, Ryu KH. A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map. *IEEE Trans Semicond Manufact*. 2019;32:286-292.
23. Araujo H, Dias JM. An introduction to the log-polar mapping. In: Proceedings II Workshop on Cybernetic Vision; 1996; Sao Carlos, Brazil, Brazil:139-144. https://doi.org/10.1109/CYBVIS.1996.629454
24. Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10); 2010:807-814.
25. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Machine Learn Res*. 2014;15(1):1929-1958. http://jmlr.org/papers/v15/srivastava14a.html
26. Kohli D, Das BC, Gopalakrishnan V, Iyer KN. Learning rotation invariance in deep hierarchies using circular symmetric filters. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017; New Orleans, LA, USA:2846-2850. https://doi.org/10.1109/ICASSP.2017.7952676

AUTHOR BIOGRAPHIES

Rui Wang is a PhD candidate in the Department of Industrial Systems Engineering and Management, National University of Singapore. She received the BE degree in Mechanical Design Manufacture and Automation from Harbin Institute of Technology in 2014. Her research interests include image processing, machine learning, and data mining for semiconductor manufacturing processes.

Nan Chen is an Associate Professor with the Department of Industrial Systems Engineering and Management, National University of Singapore. He received the BS degree in Automation from Tsinghua University in 2006, the MS degree in Computer Science in 2009, the MS degree in Statistics and the PhD degree in Industrial Engineering in 2010, all from University of Wisconsin-Madison. His research interests include statistical modeling and surveillance of engineering systems, simulation modeling design, condition monitoring, and degradation modeling.

How to cite this article: Wang R, Chen N. Defect pattern recognition on wafers using convolutional neural networks. *Qual Reliab Engng Int*. 2020;36:1245–1257. https://doi.org/10.1002/qre.2627