Two-Dimensional Principal Component Analysis-Based Convolutional Autoencoder for Wafer Map Defect Detection

Jianbo Yu and Jiatong Liu

***Abstract*—Due to the high complexity and dynamics of the semiconductor manufacturing process, various process abnormality could result in wafer map defects in many work stations. Thus, wafer map pattern recognition (WMPR) in the semiconductor manufacturing process can help operators to troubleshoot root causes of the out-of-control process and then accelerate the process adjustment. This article proposes a novel deep neural network (DNN), two-dimensional principal component analysis-based convolutional autoencoder (PCACAE) for wafer map defect recognition. First, a new convolution kernel based on conditional two-dimensional principal component analysis is developed to construct the first convolutional block of PCACAE. Second, a convolutional autoencoder is cascaded by considering the nonlinearity of data representation. The second convolutional block of PCACAE is constructed based on the encoding part. Finally, the pretrained PCACAE is fine-tuned to obtain the final classifier. PCACAE is successfully applied for feature learning and recognition of wafer map defects. The experimental results on a real-world case demonstrate that PCACAE is superior to other well-known convolutional neural networks (e.g., GoogLeNet, PCANet) on WMPR.**

***Index Terms*—Autoencoder, convolutional neural network (CNN), deep learning, semiconductor manufacturing, wafer map pattern recognition (WMPR).**

I. Introduction

In SEMICONDUCTOR manufacturing, the wafer fabrication process typically involves hundreds of steps to produce an integrated circuit (IC) on wafers. Any abnormality in the fabrication process may transfer various defects into final wafer maps. In addition to monitoring key process parameters in the IC fabrication system [1], detection and recognition of systematic defects on wafer maps are important to identify root causes of defects and provide appropriate remedies. The fault detection and recognition model can help operators to accelerate process adjustment, thereby improving production efficiency, reducing scrap rate, and avoiding huge cost losses caused by large-scale wafer defects.

Most IC-producing companies relied mainly on manual classification in the early years, which could no longer meet the actual demand due to the expansion of the manufacturing system. Therefore, some statistical-based methods were applied to wafer map pattern recognition (WMPR). Li [2] applied a 2-D wavelet transform to analyze wafer maps derived from X-ray inspection. Hwang and Kuo [3] used binary normal distribution and principal curve to recognize shapes of the defect cluster. Yuan and Kuo [4] combined spatial nonhomogeneous Poisson distribution, binary normal distribution, and principal curve model for further separation of various defect shapes. These methods have succeeded in separating different defect shapes on wafer maps. However, they have shown their weakness in detecting various defect patterns on wafer maps.

In recent years, WMPR is mostly implemented by using supervised and unsupervised machine learning methods. The supervised methods such as K-nearest-neighbor [5], support vector machine [6], artificial neural networks [7], etc., have been successfully applied in WMPR. For example, Tsai *et al.* [8] proposed a method based on independent component analysis to detect wafer defect patterns. Mean-shift technique based on a kernel density estimator was applied for detecting wafer defects [9]. A simplified subspace regression framework was introduced for the identification of defect patterns [10]. The unsupervised methods such as principal component analysis (PCA) [11], clustering [12], joint local and nonlocal linear discriminant analysis [13], etc., remove redundant information and obtain intrinsic information of images by projecting them to a lower dimension. Wang [14] recognized defect patterns in wafer maps by using filtering and spectral clustering. Alawieh *et al.* [15] clustered wafer maps according to their spatial signatures and employed binary tests to detect systematic failure patterns. Although these traditional machine learning methods have achieved some successes in WMPR, they basically focused on constructing classifiers that use predetermined features as inputs, rather than learning effective features from complex images.

In recent years, deep learning is becoming a hot topic in the area of machine learning, leading a series of breakthroughs

Manuscript received April 29, 2020; accepted July 8, 2020. Date of publication August 6, 2020; date of current version June 16, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 71777173, in part by Action Plan for Scientific and Technological Innovation of Shanghai Science and Technology Commission under Grant 19511106303, and in part by Fundamental Research Funds for the Central Universities. *(Corresponding author: Jianbo Yu.)*

The authors are with the School of Mechanical Engineering, Tongji University, Shanghai 201804, China (e-mail: jianboyu.bob@gmail.com; bang_liujt@163.com).

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

Digital Object Identifier 10.1109/TIE.2020.3013492

0278-0046 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

in both academia and industry [16]–[19]. As one of the most effective deep learning models in image recognition, convolutional neural networks (CNNs) use convolutional kernels to learn a hierarchy of features adaptively from a massive amount of image data without prior knowledge. Various CNNs, e.g., AlexNet [20], VGG [21], GoogLeNet [22], and ResNet [23], have been proposed to extract multiple levels of image representation. Meanwhile, CNNs have been widely applied in WMPR. Weimer *et al.* [24] proposed a deep CNN architecture for the wafer map defect detection. Nakazawa and Kulkarni [25] trained CNNs by using 28 000 synthetic wafer maps to achieve good recognition of wafer map defects. Kyeong and Kim [26] recognized mixed defect patterns on wafer images within the framework of several single classification models. Other deep neural networks (DNNs), e.g., stacked denoising autoencoders (SDAE) have been successfully applied in WMPR. Lee *et al.* [27] used SDAE to establish a standard fault detection and classification model. Yu [28] proposed an SDAE ensemble-based feature learning method. Although DNNs (e.g., CNN)-based feature learning methods show more effective performance than that of other methods, there are still some problems remained to be solved. First, all the above DNNs require a random initialization [29], [30], completely independent of the training samples, which results in high training cost and slow convergence. There are some other works that are applied to real-word product defects recognition [31], [32]. Second, these previous works do not consider the class imbalance of real-word datasets, and thus, the DNNs generalize not so well on the small class. Finally, the initialization of CNNs is still a challenging problem [33].

To solve the class imbalance and image feature learning problem, a novel CNN model, two-dimensional principal component analysis-based convolutional autoencoder (PCACAE) is proposed for feature learning and defect recognition of wafer maps. The main contributions of this article can be summarized as follows.

1) A novel semisupervised algorithm, conditional two-dimensional principal component analysis (C2DPCA) is proposed to extract effective features from the wafer maps with class imbalance. On one hand, C2DPCA can serve as a simple but effective projection algorithm for image dimension reduction; on the other hand, C2DPCA-based convolutional kernels have powerful feature extraction ability.
2) A new deep network structure, i.e., C2DPCA kernel-based convolutional autoencoder (CAE) that consists of a convolutional encoder and decoder is proposed to further extract abstract features from the input feature maps from C2DPCA-based kernels.
3) The layer-wise pretraining of PCACAE is an effective initialization method, which achieves the minimal initial error and has the rapid convergence in the training process. PCACAE performs comparable to or better than the state-of-the-art DNNs on WMPR, especially on the small classes, while maintaining a low computational cost.

The remainder of this article is organized as follows. Section II, the proposed DNN model, i.e., the network structure of PCACAE, the C2DPCA-based kernels, and the CAE-based kernels are presented. The WMPR method based on PCACAE is developed in Section III. The experimental results are presented in Section IV to demonstrate the effectiveness of PCACAE. Finally, Section V concludes this article.

II. Methodology

In this section, the new DNN model, i.e., PCACAE is proposed for WMPR online. PCACAE is a hybrid learning model, where C2DPCA-based convolution kernels are embedded into a stacked CAE model. C2DPCA-based convolution kernels are capable of filtering redundant information and then extracting orthogonal linear information of the images, which enables it to learn effective features from those noised images.

A. Network Structure

The network structure of PCACAE is quite different from those traditional CNNs for WMPR. The network structure of PCACAE is presented in Fig. 1. It contains an input layer, two convolutional modules, and an output module. Each convolutional module is composed of a convolutional layer, a batch normalization (BN) function, and a max-pooling layer, referring to those classic network structures such as GoogLeNet, ResNet, etc. The output module consists of two fully connected layers and a softmax classifier. The learning process of PCACAE is a layer-wise learn-update-remove loop.

In the C1 block, the convolutional layer is composed of N_{C1}, i.e., C2DPCA-based kernels with the size F_{C1} and stride S_{C1}, followed by a BN function. Given the input size $w \times h \times c$, normalized features with a size of $(\frac{w-F_{\mathrm{C1}}}{S_{\mathrm{C1}}}+1) \times (\frac{h-F_{\mathrm{C1}}}{S_{\mathrm{C1}}}+1) \times N_{\mathrm{C1}}$ can be obtained. After that, the pooling layer performs max-pooling operation with size F_{P1} and stride 1. Then, outputs of the C1 block with size $\frac{w-F_{\mathrm{C1}}+S_{\mathrm{C1}}}{S_{\mathrm{C1}}F_{\mathrm{P1}}} \times \frac{h-F_{\mathrm{C1}}+S_{\mathrm{C1}}}{S_{\mathrm{C1}}F_{\mathrm{P1}}} \times N_{\mathrm{C1}}$, denoted as $W_{\mathrm{C1}} \times H_{\mathrm{C1}} \times N_{\mathrm{C1}}$, can be obtained. In the C2 block, the convolutional layer is composed of N_{C1} i.e., CAE-based kernels with size $F_{\mathrm{C2}}$and stride S_{C2}. The outputs of the C1 block are input into the C2 block. After a stack of operations, the feature maps with size $(\frac{W_{\mathrm{C1}}-F_{\mathrm{C2}}}{S_{\mathrm{C2}}}+1) \times (\frac{H_{\mathrm{C1}}-F_{\mathrm{C2}}}{S_{\mathrm{C2}}}+1) \times N_{\mathrm{C2}}$will be extracted.

In the output block, the convolutional outputs are transformed into a 1-D vector through two fully connected layers, which integrates the local information contained in the 2D feature maps. Finally, the feature vector is fed into a Softmax classifier that outputs the final classification results. The dropout strategy is introduced to improve computational efficiency and reduce overfitting.

B. C2DPCA-Based Convolutional Kernels

PCA, reconstructing input data by mapping them into lower dimensional vectors, is a very effective method in dimension reduction and feature extraction. However, the 2-D image matrices must be previously transformed into a high dimensional 1-D image vectors for PCA. This makes it difficult to obtain the covariance matrix accurately. Thus, an image projection method, 2DPCA [34], is developed for image feature extraction. 2DPCA is based on 2-D matrices rather than 1-D vectors, that is, 2DPCA

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

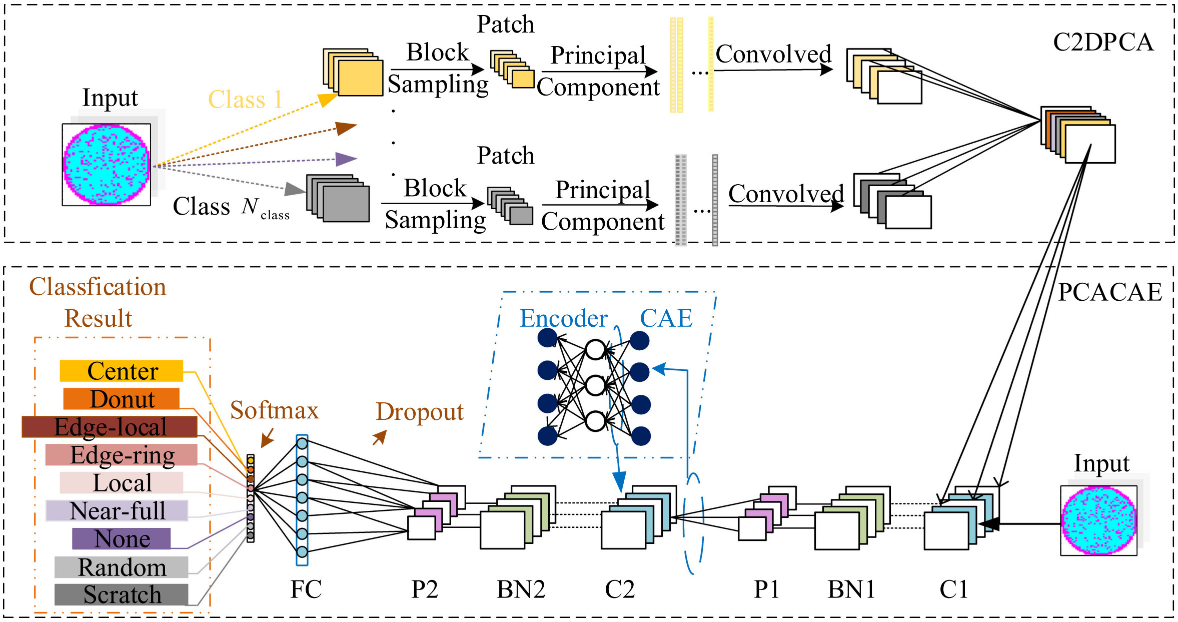

Fig. 1. Network structure of PCACAEA.

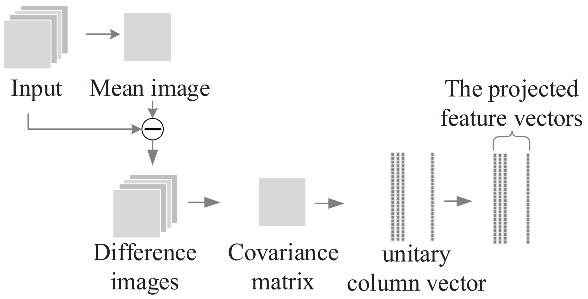

Fig. 2. Calculation procedure of 2DPCA.

can preserve the spatial characteristics of the input images effectively. The calculation procedure of 2DPCA is shown in Fig. 2.

2DPCA mainly focuses on the majority class and it is easy to miss valid information contained in the minority class. In order to alleviate the class imbalance problem, a new 2DPCA algorithm, i.e., C2DPCA transforms unsupervised 2DPCA into a semisupervised or supervised algorithm by considering class label distribution. Given M training image samples $\boldsymbol{I} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_M\}$ belonging to N_{class}classes$K = \{K_1, K_2, \ldots, K_{N_{\text{class}}}\}$, it is easy to obtain the mean image of class i

$$\boldsymbol{E}_i = \frac{1}{n_i}\sum_{i=1}^{N_{\text{class}}}\sum_{a=1}^{n_i}(\boldsymbol{I}_a \cdot P(\boldsymbol{I}_a|K_i)) \tag{1}$$

where $\boldsymbol{I}_a$ is the ath training sample, n_i denotes the total number of images belonging to the class i, and $P(\boldsymbol{I}_a|K_i)$ indicates the probability that the ath sample belongs to the class i

$$P(\boldsymbol{I}_a|K_i)=\begin{cases}1, & \boldsymbol{I}_a \in K_i\\ 0, & \boldsymbol{I}_a \notin K_i\end{cases}. \tag{2}$$

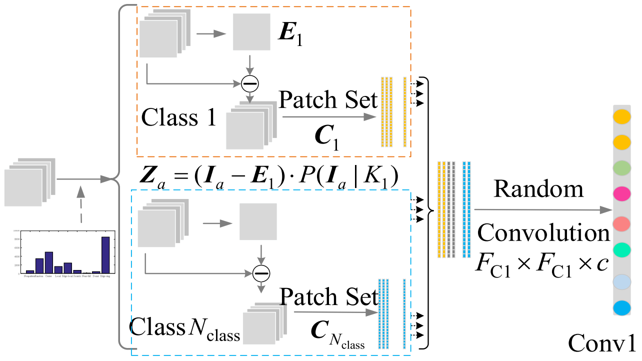
$\boldsymbol{E}_1$
Class 1
Patch Set
$\boldsymbol{C}_1$
$\boldsymbol{Z}_a = (\boldsymbol{I}_a - \boldsymbol{E}_1)\cdot P(\boldsymbol{I}_a|K_1)$
Class N_{class}
Patch Set
$\boldsymbol{C}_{N_{\text{class}}}$
Random
Convolution
$F_{C1}\times F_{C1}\times c$
Conv1

Fig. 3. C2DPCA-based convolutional kernels.

By computing the difference between each image and the mean images of each class, the difference image can be obtained

$$\boldsymbol{Z}_a = (\boldsymbol{I}_a - \boldsymbol{E}_i)\cdot P(\boldsymbol{I}_a|K_i). \tag{3}$$

Around each pixel of Z_a, a $F_{C1} \times F_{C1}$ patch is taken and these patches of Z_a:$\boldsymbol{X}_{ij} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{mn}\}$ are collected, where$m = w - [F_{C1}/2]$ and $n = h - [F_{C1}/2]$. Thus, a total patch set of class i: $\boldsymbol{X}_i = \{\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2}, \ldots, \boldsymbol{X}_{in_i}\}$will be obtained, and the image covariance matrix $\boldsymbol{C}_i$ for the class K_i can be expressed as

$$\boldsymbol{C}_i = \boldsymbol{X}_i{}^T\boldsymbol{X}_i. \tag{4}$$

Fig. 3 presents the calculation procedure of C2DPCA. In order to find the optimal projection direction, i.e., the object of C2DPCA is to minimize the reconstruction error, i.e.,

$$\min_{\boldsymbol{V}\in\boldsymbol{U}^{p\times p\times d}} ||\boldsymbol{X}_i - \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}_i||_{\mathrm{F}}^2 \tag{5}$$

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

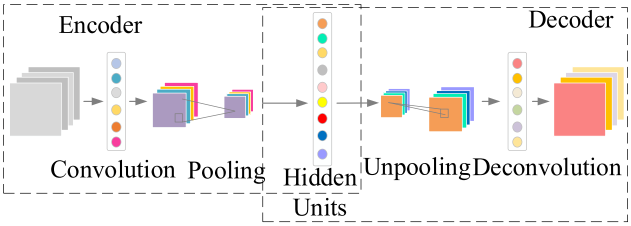

Fig. 4. Convolutional autoencoder.

where $\boldsymbol{V}$ is the identity matrix with size $d \times d$. The solution is known as the d principal eigenvectors of the covariance matrix $\boldsymbol{C}_i$. After transforming these one-dimension principal eigenvectors to a two-dimension matrix, d convolution filters of size $F_{\mathrm{C1}} \times F_{\mathrm{C1}} \times c$ can be obtained. Thus, C2DPCA-based kernels can be expressed as

$$\boldsymbol{w}_{id} = \mathrm{mat}_{p\times p}\left(\boldsymbol{q}_d({\boldsymbol{X}_i}^T \boldsymbol{X}_i)\right) \tag{6}$$

where $\mathrm{mat}_{p\times p}(\boldsymbol{v})$ is a function that maps $\boldsymbol{v} \in \boldsymbol{U}^{p\times p}$ to a projection space $\boldsymbol{U}^{p\times p}$, $\boldsymbol{q}_d({\boldsymbol{X}_i}^{\mathrm{T}}\boldsymbol{X}_i)$ denotes the dth principal component of $\boldsymbol{C}_i$ and $\boldsymbol{w}_{id}$ is the dth extractor learned from images belonging to class K_i. $\boldsymbol{W}_i = \{\boldsymbol{w}_{i1}, \boldsymbol{w}_{i2}, \ldots, \boldsymbol{w}_{id}\}$ denotes the filter set learned from the class K_i. The C2DPCA-based convolution kernels can finally be expressed as $\boldsymbol{W} = \{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_{N_{\mathrm{class}}}\}$.

C. CAE-Based Convolutional Kernels

C2DPCA-based convolutional layer has alleviated the class imbalance problem. However, it is easy to lead to dimension expansion of feature maps in the C1 block. Thus, a layer-wise network, i.e., convolutional autoencoder, is embraced as the second block of PCACAE to promote the feature learning from images. To further reduce computation cost, CAE is integrated with group convolutions.

Autoencoder is an unsupervised learning model that contains an encoder and a decoder. The encoder extracts latent features from original inputs and the decoder attempts to reconstruct input data from the hidden representations. Autoencoder minimizes the reconstruction error during the training process, ensuring that the reconstructed data can approximate the original input to a maximum extent. CAE integrates local convolutional connection with autoencoder, completing pretraining process of each convolution layer in the AE model. The operations in a CAE model are illustrated in Fig. 4.

The convolutional conversion from original inputs to abstract latent features is called a convolutional encoder. Given the N_{C1} feature maps $I = \{I_1, I_2, \ldots, I_{N_{\mathrm{C1}}}\}$, the latent feature vector (hidden units) of an autoencoder can be calculated as follows:

$$h_m(i,j) = a\left(\sum_{u=-k}^{k}\sum_{v=-k}^{k} \boldsymbol{F}_m^{(1)}(u,v) * \boldsymbol{I}(i-u, j-v) + b_m^{(1)}\right) \tag{7}$$

where $h_m(i,j)$ is the activation value of receptive filed centered on pixel (i,j) of the mth map, $a(\cdot)$ is a nonlinear activation function, $\boldsymbol{F}_m^{(1)}$ denotes the convolutional encode operators, $b_m^{(1)}$ is the encode bias corresponding to the mth image.

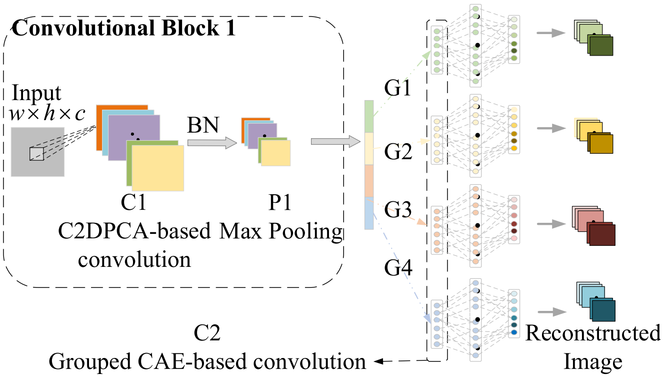

Fig. 5. CAE-based convolution layer.

The convolutional reconstruction from latent features to the original inputs I is called a convolutional decoder. Considering that the size of feature maps obtained after encode are smaller than the original inputs, the reconstruction cannot be implemented by the inverse convolution directly. Thus, the padding operation is performed to ensure that the reconstructed images have the same size as that of the original inputs. The Houtput from the encoder is decoded that the decoder producse $\tilde{\boldsymbol{I}}$ to approximate the original input I

$$\tilde{\boldsymbol{I}} = g(\boldsymbol{H} * \boldsymbol{F}_m^{(2)} + b_m^{(2)}) \tag{8}$$

where $\boldsymbol{H}$ is the latent feature maps, and $b_m^{(2)}$ is the decode bias corresponding to the mth activation map. The error minimization can be achieved by optimizing the mean square error

$$\min(L(\boldsymbol{I}, \tilde{\boldsymbol{I}})) = \min\left(\sum_{i=1}^{d} ||\boldsymbol{I} - \tilde{\boldsymbol{I}}||^2\right). \tag{9}$$

As shown in Fig. 5, the grouped CAE in PCACAE is to reconstruct the output from Convolution block C1. The feature maps obtained from C1 are divided into four groups and four independent CAEs are used to extract nonlinear features. The grouped CAE is computationally lightweight and, thus, accelerates the training process. The weights of the convolutional encoding part are used as the kernels of the second convolution layer in the block C2.

III. Application Procedure of PCACAE

The application procedure of PCACAE for WMPR mainly includes two parts: offline modeling and online recognition. The detailed procedure is presented in Fig. 6 and is further summarized as follows:

Part I: Off-line modeling

This part consists of the following five steps:

Step 1: Collect the normal and defect wafer map images to generate the training dataset.

Step 2: PCACAE uses C2DPCA to learn the principal components of various labeled images and implement convolutional kernel calculation.

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

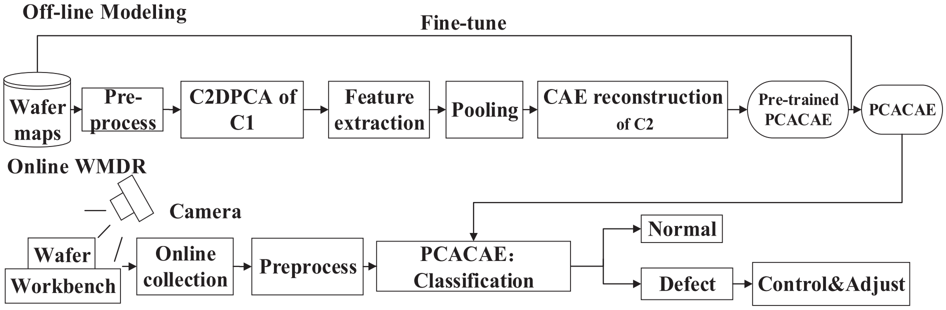

Fig. 6. Application procedure of PCACAE.

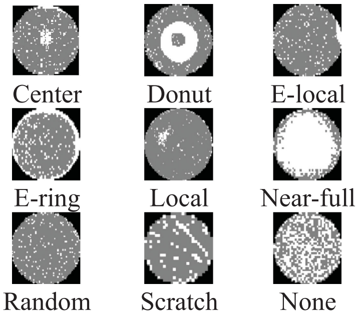

Fig. 7. List of wafer defect patterns.

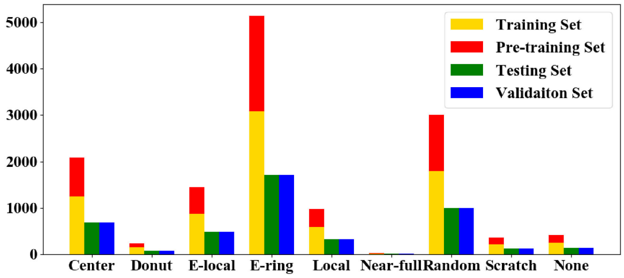

Fig. 8. Wafer map distribution of the dataset.

Step 3: The features obtained from the first convolution block are used as inputs to train CAE.

Step 4: The outputs of the encoding part are used as inputs of the second convolution layer.

Step 5: Based on the extracted features by PCACAE, the classifier layer of PCACAE is fine-tuned.

Part II: Online WMPR

This part consists of the following four steps:

Step 1: Preprocess wafer images and input them to the well-trained PCACAE.

Step 2: Extract the representative features by PCACAE from the wafer map.

Step 3: Input the extracted features to the classification layer of PCACAE to output the prediction.

Step 4: Based on the recognition result, identify the corresponding fault root cause to adjust the out-of-control process.

IV. Experiment and Result Analysis

In this section, the real-word industrial dataset (i.e., WM-811K [35]) is used to verify the effectiveness of the proposed PCACAE-based WMPR method. The CNN models are coded in Python 3.6 with TensorFlow and MATLAB, running on Windows10 with a GTX 1050Ti.

The WM-811K dataset, comprising 811 457 real-world labeled wafer maps, contains nine different wafer map patterns, as shown in Fig. 7, including a normal pattern (i.e., None) and 8 defect patterns (i.e., Center, Donut, Edge-local, Edge-ring, Local, Near-full, Random, and Scratch).

The 22 418 images are randomly selected from WM-811K to generate the dataset. Each original image is preprocessed to 96 × 96 pixels. The dataset is split into three parts based on a ratio of 6:2:2, including 13 451 training samples, 4483 validation samples, and 4483 testing samples. The training dataset is further separated into two parts, including samples for pretraining and samples for fine-tuning, named as S_1 and S_2,respectively. The class distribution of S_1 is the same as that of S_2, i.e., the samples are selected from each pattern with a fixed ratio of 0.2. The training and validation sets are used in the training stage, and the test set is used to evaluate the models. Fig. 8 shows the distribution of the wafer map patterns in the dataset. It is clear that the wafer map dataset is highly imbalanced, which affects the feature learning performance of DNNs.

A. Wafer Map Defect Recognition

To learn the C2DPCA-based kernels, the max training set of each pattern, i.e., $n_{\max}$, is set to 2000, the patch size is set to 7 × 7, and eight principal components are learned from each class. Since the input dataset includes nine different patterns, a total of 72 principal feature vectors can be obtained, and 64 convolutional kernels are used to construct the convolutional layer of the C1 block. To learn the grouped CAE, the decay parameters are set 0.001, and the dropout rate is set 0.5. PCACAE is optimized via stochastic gradient descent with decay parameters of 0.0005 and learning rate for the first 30 epochs and the last 10 epochs, 0.01 and 0.001, respectively, with the batch size of 64 for 40 epochs.

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

TABLE I
STRUCTURE PARAMETERS OF PCACAE

Block	Layer	Type	Filter size/stride	Output
C1	Conv1	Convolution	7×7/1	90×90×64
C1	BN1	Batch normalization	none	90×90×64
C1	Pool1	Max pooling	5×5/1	18×18×64
C2	Conv2	Convolution	5×5/1	15×15×256
C2	BN2	Batch normalization	none	15×15×256
C2	Pool2	Max pooling	3×3/1	5×5×256
O1	FC1	Fully connection		4096
O1	Dropout	Dropout (50%)		4096
O1	FC2	Fully connection		9×1

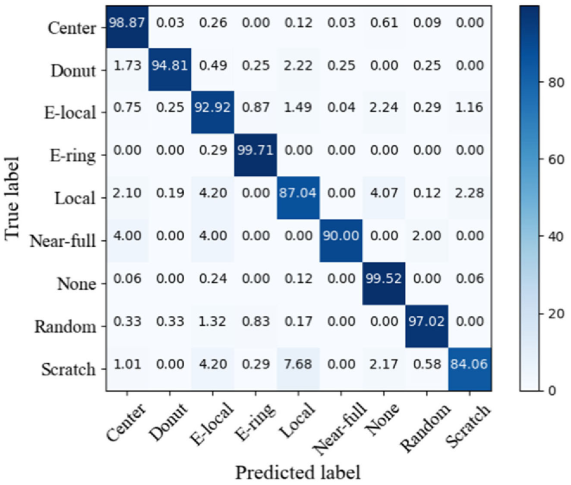

Fig. 9. Confusion matrix the PCACAE-based method.

After the calculation of the C2DPCA-based kernels on the pretraining set S_1 is finished, the weights of the Conv1 layer are frozen, and the extracted features are input into the C2 block to implement the layer-wise pretraining of PCACAE. The pretrained PCACAE is further fine-tuned on the dataset S_2 until it shows good generalization on the validation dataset. As shown in Fig. 13, PCACAE basically converges on the training and validation set when iteration times $= 10$ and obtains an overall accuracy rate 92% on the validation set.

The filter size and the dimensions of the input and output in each layer can be seen in Table I. The input layer receives the 96×96 wafer images, and then inputs them to the C1 block, obtaining feature maps of size 18×18, with 64 channels. These features are input into the C2 block, obtaining maps of size 5×5, with 256 channels. Finally, these maps are input into output block to implement the mapping from the $5 \times 5 \times 256$ matrix sets to the 4096-dimensional abstract features.

The confusion matrix of the test result of PCACAE is presented in Fig. 9. The rows stand for the actual label, and the columns stand for the predicted label for each pattern. It can be observed from the result that most patterns have been accurately recognized. The recognition accuracies on Local and Scratch are relatively low, but still nearly close to 90%. Despite the class imbalance problem, PCACAE still has achieved high prediction accuracies on the minority classes, i.e., 97% on Random, 90% on Near-full, and 94.81% on Donut.

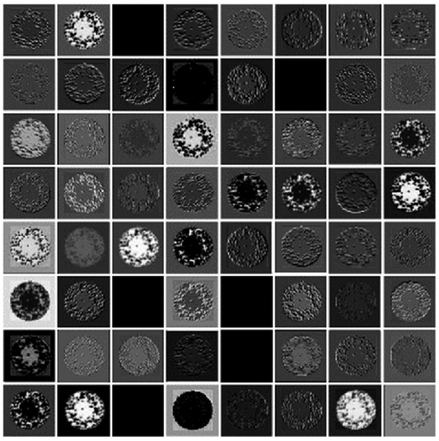

Fig. 10. Feature visualization of output of C1 block.

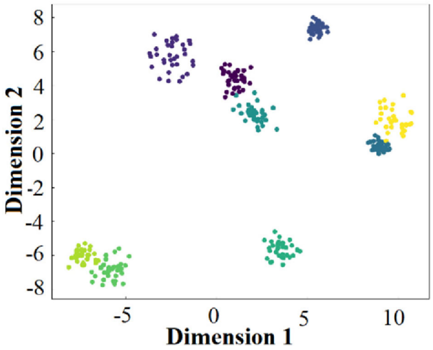

Fig. 11. Feature visualization of output of C2 block.

Furthermore, the 64-channel feature output of the C1 block is visualized in Fig. 10. It is obvious that C2DPCA-based convolution kernels show good feature learning performance, which preserves the principal information and eliminates most random noise and then extracts effective abstract features. T-distributed stochastic neighbor embedding is further used to map the outputs of the C2 block to a lower dimension space. A visual representation of these extracted features is shown in Fig. 11. It is clear that the features extracted by PCACAE effectively separate the nine wafer patterns, which illustrates that the outputs of the C2 block contain class discriminant information. This further explains the powerful feature extraction of PCACAE.

B. Parameter Sensitivity

The affection of the number of filters on the recognition performance of PCACAE is first investigated in this section. The filter sizes of the first and second convolutional layer are set as default. The number of filters (i.e., N_{C1}) in the first stage is varied from 16 to 256, corresponding to the number of the principal components learned from each pattern varied from 2 to 29. When there are two convolutional blocks in the deep network, N_{C2} is set to 64, 128, and 256, and N_{C1} is varied from 16 to 128, corresponding to the number of principal components

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

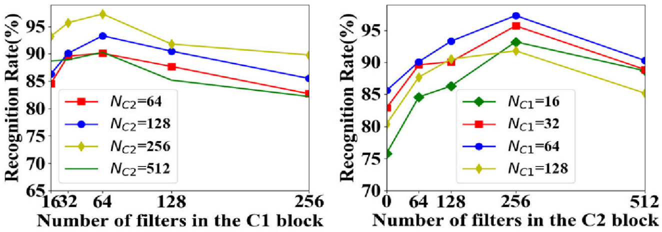

Fig. 12. Parameter sensitivity of number of filters of PCACAE.

TABLE II
RESULTS OF PCACAE WITH DIFFERENT COMBINATION OF MULTIPLE KERNELS (%)

Y	2DPCA	CAE	D_1	D_2	D_3	D_4
√	√	√	89.26	87.82	96.52	96.72
	√	√	88.39	88.01	94.34	93.61
		√	88.00	85.76	93.79	93.57
√	√		88.34	85.93	96.41	96.43
			88.15	85.71	96.37	96.54

learned from each pattern in the block C1 from 2 to 15. The recognition results during different training processes are shown in Fig. 12. It can be seen that the one-stage network achieves better validation performance for $N_{\mathrm{C1}} \geq 64$, and the two-stage cascade network is the best for all N_{C1}, except $N_{\mathrm{C2}} = 64$. Since the deeper network can extract more abstract features, the two-stage network generally outperforms the single-stage network. Moreover, the validation accuracy increases first and then decreases for the larger N_{C1} in the two-stage network, achieving the best performance when $N_{\mathrm{C1}} = 64$, and $N_{\mathrm{C2}} = 256$, i.e., the eight principal components are selected for C2DPCA in the first stage.

C. Ablation Studies

Ablation studies, typically referring to removing some "feature" of the model or algorithm and seeing how that affects the final performance, are often performed in computer vision [36]. In this study, 2DPCA with no label distribution and different cascade networks are considered to investigate the influence of key techniques in PCACAE, i.e., 2DPCA with auxiliary inputs, the C2DPCA-based pretraining method, and the cascade structure. To investigate the effect of combination of different pretraining methods, the four datasets with different scales are generated, i.e., $D_1 = \{\ S_1 = 11\,409,\ S_2 = 2\,282\}$, $D_2 = \{\ S_1 = 4\,564,\ S_2 = 2\,282\}$, $D_3 = \{\ S_1 = 11\,409,\ S_2 = 4\,564\}$, and $D_4 = \{\ S_1 = 4\,564,\ S_2 = 22\,818\}$, where S_1 and S_2 denote the pretraining set and the fine-tuning set, respectively.

Table II shows the ablation investigation on these four datasets. These results indicate that the good performance of PCACAE can be attributed to the cascade of C2DPCA-based kernels and encoding kernels of CAE. The following can be observed from Table II.

1) The recognition accuracies in the first row of the table are generally higher than those in the other rows. This illustrates the effectiveness of the cascade structure.
2) The comparison between the first and the second row indicates the effectiveness of C2DPCA-based convolutional kernels.
3) PCACAE can achieve a better performance on a pretraining and a fine-tuning dataset with a larger size. Moreover, the large size of pretraining dataset improves the feature learning performance of PCACAE.

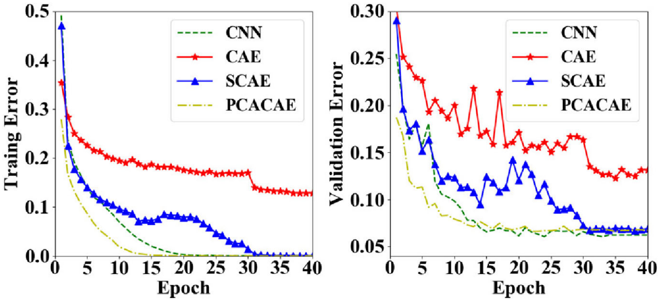

Fig. 13. Training and validation error curves of different networks.

TABLE III
CLASSIFICATION PERFORMANCE OF DIFFERENT NETWORKS

Network	Acc%(Top-1)	Acc%(Top-5)	Flops
PCACAE	97.3	100	143
CNN	96.4	100	185
SCAE	90.6	97.5	193
CAE	87.0	96.3	78

The typical DNNs, i.e., CNN, CAE, and SCAE [37] are further considered to verify the effectiveness of the cascade structure in PCACAE. The detailed information about the network structure of these DNNs is shown in Table I. The C1 block is replaced by CAE to construct the SCAE. The C1 block is removed to construct the one-stage CAE model. Xavier is used to initialize the network to construct the regular CNN model. The same hyperparameters settings are used for all models for a fair comparison. The training and validation curves for each network are shown in Fig. 13. These results show that first, PCACAE achieves the minimal error in the first epoch both on the training and validation sets, i.e. 28% on the training set and 18% on the validation set. This indicates the effectiveness of the pretraining process in PCACAE. Second, in the first 15 iterations, PCACAE reduces the Top-1 error on both the training and validation sets at a much faster rate than other networks, with Top-1 error reduced to around 10% in the fifth iteration. After the 40 training iterations, all networks achieve similar accuracy on the training and validation sets. The average classification performance of five tests on the same testing dataset is shown in Table III. These result comparisons further illustrate the effectiveness of the pretraining process in PCACAE.

D. Comparison With State-of-the-Art Works

Finally, the feature learning performance of PCACAE is compared with these state-of-the-art DNNs (i.e., AlexNet [20], GoogleNet [22], PCANet [38]), SDAE, and DBN. SDAE and DBN consist of two hidden layers with 64 units and 256 units

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
FIVEFOLD CROSS-VALIDATIONS (%) OF PCACAE AND OTHER DNNS

Classifier	PCACAE	GoogleNet	AlexNet	SDAE	DBN	PCANet
Center	98.87	98.63	98.25	98.49	97.13	98.65
Donut	94.81	91.72	90.34	87.63	86.32	88.79
Edge-local	92.92	90.83	88.65	87.51	86.21	87.43
Edge-ring	99.71	99.07	99.01	98.76	98.52	97.75
Local	87.04	89.68	85.79	83.49	81.87	83.6
Near-full	90.00	88.79	86.83	85.34	83.89	84.32
None	99.52	99.37	99.49	99.04	98.97	98.64
Random	97.02	96.21	95.62	93.27	91.36	92.32
Scratch	84.06	83.9	79.34	81.68	76.31	78.83
Accuracy	**97.27**	93.13	91.48	90.58	88.95	·90.04
Training time (sec/iter)	**0.58**	0.75	0.73	0.83	0.89	0.65

and a softmax layer is used to output the final prediction result. As for parameter setup of other CNNs, the initial learning rate is set to 0.001 and is reduced with a factor of 0.2 every 10 epochs, the batch size is set to 64, and Adam optimizer is used for optimization.

Table IV shows the recognition results of these DNNs, including the recognition accuracies on each pattern of the testing dataset, as well as the training time. It is obvious that PCACAE outperforms the state-of-art deep models. It can be observed that all DNNs perform relatively worse on the minority classes that have a smaller number of samples, e.g., Local, Near-full, and Scratch. However, PCACAE still obtains acceptable and the best recognition performance on these classes, achieving an accuracy of 87.04% on Local, 90% on Near-full, and 84.06% on Scratch. Second, PCACAE outperforms other DNNs, because it obtains the best recognition accuracy of 97.27% and spends the least training time per iteration. This comparison further demonstrates the effectiveness of PCACAE on WMPR.

V. CONCLUSION

In this study, a novel DNN, PCACAE, was proposed for the detection and recognition of wafer map defects. PCACAE was composed of an input layer, a semi-supervised convolutional module, an unsupervised grouped encoding module, and an output module. One of the objectives of this study was to cope with the class imbalance problem. A new 2DPCA, i.e., C2DPCA was developed as convolutional kernels to extract abstract features from wafer maps. The convolutional kernels were then cascaded with a convolutional autoencoder to further improve the feature learning performance. PCACAE reduces network convergence and training cost and shows strong feature extraction ability. A system based on PCACAE was applied in WMPR. The experimental results on the industrial case demonstrated that PCACAE outperforms other state-of-the-art DNNs.

Although the testing results are encouraging, there are still some potential limitations in the PCACAE-based WMPR method. First, the proposed method only focuses on the single defect pattern recognition on wafer maps. Second, C2DPCA is a linear projection algorithm and, thus, could not deal with nonlinear data very well. Third, the principal component selection affects the performance of convolutional filters. Finally, these disturbance factors, e.g., brightness, contrast, and blur perturbations on wafer maps should be considered to improve the applicability of the proposed method. Thus, the future work will focus on these issues to further improve the effectiveness of PCACAE in WMPR.

REFERENCES

[1] W. W. Tan, R. F. Li, A. P. Loh, and W. K. Ho, “RTD response time estimation in the presence of temperature variations and its application to semiconductor manufacturing,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 2, pp. 406–412, Jan. 2008.

[2] X. Li, “Improving automatic detection of defects in castings by applying wavelet technique,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 6, pp. 1927–1934, Dec. 2006.

[3] 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, Apr. 2007.

[4] T. Yuan and W. Kuo, “A model-based clustering approach to the recognition of the spatial defect patterns produced during semiconductor fabrication,” *IIE Trans.*, vol. 40, no. 2, pp. 93–101, Dec. 2007.

[5] Q. P. He and J. Wang, “Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes,” *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 345–354, Nov. 2007.

[6] R. Baly and H. Hajj, “Wafer classification using support vector machines,” *IEEE Trans. Semicond. Manuf.*, vol. 25, no. 3, pp. 373–383, Apr. 2012.

[7] C. Y. Chang, C. Li, J. W. Chang, and M. Jeng, “An unsupervised neural network approach for automatic semiconductor wafer defect inspection,” *Expert Syst. Appl.*, vol. 36, no. 1, pp. 950–958, Jan. 2009.

[8] D. M. Tsai, S. C. Wu, and W. Y. Chiu, “Defect detection in solar modules using ICA basis images,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 122–131, Jul. 2012.

[9] D. M. Tsai and J. Y. Luo, “Mean shift-based defect detection in multicrystalline solar wafer surfaces,” *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 125–135, Dec. 2010.

[10] F. Adly *et al.*, “Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps,” *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1267–1276, Sep. 2015.

[11] G. A. Cherry and S. J. Qin, “Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis,” *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 2, pp. 159–172, May 2006.

[12] M. P. L. Ooi, E. K. J. Joo, Y. C. Kuang, S. Demidenko, L. Kleeman, and C. W. K. Chan, “Getting more from the semiconductor test: Data mining with defect-cluster extraction,” *IEEE Trans. Instrum. Meas.*, vol. 60, no. 10, pp. 3300–3317, Mar. 2011.

[13] 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.

[14] C. H. Wang, “Recognition of semiconductor defect patterns using spatial filtering and spectral clustering,” *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1914–1923, Apr. 2008.

[15] M. B. Alawieh, F. Wang, and X. Li, “Identifying wafer-level systematic failure patterns via unsupervised learning,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 4, pp. 832–844, Apr. 2018.

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.

[16] L. Wen, X. Li, and L. Gao, “A new convolutional neural network-based data-driven fault diagnosis method,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Nov. 2017.

[17] G. Jiang, H. He, J. Yan, and P. Xie, “Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3196–3207, Jun. 2018.

[18] W. Long, X. Li, and L. Gao, “A new two-level hierarchical diagnosis network based on convolutional neural network,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 330–338, Feb. 2020.

[19] L. Guo, Y. Lei, and S. Xing, “Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data,” *IEEE Trans. Ind. Electron.*, vol. 66, no.9, pp. 7316–7325, Sep. 2019.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[21] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.

[22] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[24] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, “Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection,” *CIRP Annu. Manuf. Technol.*, vol. 65, no. 1, pp. 417–420, May. 2016.

[25] T. Nakazawa and D. V Kulkarni, “Wafer map defect pattern classification and image retrieval using convolutional neural network,” *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018.

[26] 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.

[27] H. Lee, Y. Kim, and C. O. Kim, “A deep learning model for robust wafer fault monitoring with sensor measurement noise,” *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 1, pp. 23–31, Nov. 2017.

[28] J. Yu. “A selective deep stacked denoising autoencoders ensemble with negative correlation learning for gearbox fault diagnosis,” *Comput. Ind.*, vol. 108, pp. 62–72, Jun. 2019.

[29] H. Shen, “Towards a mathematical understanding of the difficulty in learning with feedforward neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2018, pp. 811–820.

[30] Y. Jia *et al.*, “Caffe: Convolutional architecture for fast feature embedding,” in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, vol. 3, pp. 675–678.

[31] Y. Wu, B. Jiang, and N. Lu, “A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 2108–2118, Oct. 2019.

[32] A. Tjahjono, D. O. Anggriawan, A. Priyadi, M. Pujiantara, and M. H. Purnomo. “Overcurrent relay curve modeling and its application in the real industrial power systems using adaptive neuro fuzzy inference system,” in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl.*, 2015, pp. 1–6.

[33] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell, “Data-dependent initializations of convolutional neural networks,” 2015, *arXiv:1511.06856*.

[34] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, “Two-dimensional PCA: A new approach to appearance-based face representation and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2014.

[35] 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.

[36] S. Ren *et al.*, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[37] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, vol. 6791, pp. 52–59.

[38] T. H. Chan, K. Jia, and S. Gao, “PCANet: A simple deep learning baseline for image classification,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

Jianbo Yu received the B.Eng. degree from the Department of Industrial Engineering, Zhejiang University of Technology, Zhejiang, China, in 2002, the M.Eng. degree from the Department of Mechanical Automation Engineering, Shanghai University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Industrial Engineering and Management, Shanghai Jiaotong University, Shanghai, in 2009.

From 2009 to 2013, he worked as an Associate Professor with the Department of Mechanical Automation Engineering, Shanghai University, Shanghai. Since 2016, he worked as a Professor with the School of Mechanical Engineering, Tongji University, Shanghai. His current research interests include intelligent condition-based maintenance, machine learning, quality control, and statistical analysis.

Jiatong Liu received the B.Eng. degree in 2018 from the Department of Industrial Engineering, Tongji University, Shanghai, China, where, she is currently working toward the master’s degree from the Department of Industrial Engineering.

Her current research interests include computer vision, fault diagnosis, and machine learning.

Authorized licensed use limited to: Inha University. Downloaded on October 08,2024 at 07:24:21 UTC from IEEE Xplore. Restrictions apply.