Computers in Industry 109 (2019) 121–133

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map

Jianbo Yu*, Xiaoyun Zheng, Jiatong Liu

School of Mechatronics Engineering, Tongji University, 4800 Caoan Road, 200084, Shanghai, PR China

ARTICLE INFO

Article history:
Received 1 February 2019
Received in revised form 2 April 2019
Accepted 26 April 2019
Available online 7 May 2019

Keywords:
Semiconductor manufacturing
Wafer map
Deep learning
Convolutional neural network
Pattern recognition

ABSTRACT

In semiconductor manufacturing systems, those defects on wafer maps tend to cluster and then these spatial patterns provide important process information for helping operators in finding out root-causes of abnormal processes. Deep learning has achieved many successes in image and visual analysis. This study concentrates on developing a hybrid deep learning model to learn effective discriminative features from wafer maps through a deep network structure. This paper proposes a novel feature learning method, stacked convolutional sparse denoising auto-encoder (SCSDAE) for wafer map pattern recognition (WMPR) in semiconductor manufacturing processes, in which the features will be extracted from images directly. Different from the regular stacked denoising auto-encoder (SDAE) and convolutional neural network (CNN), SCSDAE integrates CNN and SDAE to learn effective features and accumulate the robustness layer by layer, which adopts SDAE as the feature extractor and stacks well-designed fully connected SDAE in a convolutional way to obtain much robust feature representations. The effectiveness of the proposed method has been demonstrated by experimental results from a simulation dataset and real-world wafer map dataset (WM-811K). This study provides the guidance to applications of hybrid deep learning in semiconductor manufacturing processes to improve product quality and yields.

1. Introduction

In the semiconductor manufacturing, wafer maps are often used to visualize various defect patterns and identify potential process problems. The semiconductor manufacturing process is highly complicated in nature with a high possibility to introduce various defects into the final products [1–7]. Detection and recognition of wafer map defects provide important process information to prevent a large number of defect wafers and identify root-causes of the out-of-control process, which will accelerate adjustment procedure of these processes. Prompt detection of abnormal wafers is an effective way for increasing yield and product quality. Some typical defect patterns (e.g., ring, scratch, semicircle) often exhibit on wafer maps in the manufacturing process. These typical defect patterns generally connect those possible causes of failure or process variations. For instance, the edge ring arises from the etching's problem, the linear scratch is created by the machine handling, and the center (cluster) generally is generated by the thin film deposition [2]. These exhibited patterns on wafer maps provide some important clues on which step of the manufacturing process that is responsible for those process failures [8,9]. In general, semiconductor fabs employ regular control charts to monitor the total number of defects found on wafers. However, this method is not adequate for process variation detection and fault diagnosis. Therefore, wafer map patterns recognition (WMPR) in semiconductor processes have become increasingly important as an effective means of quickly identifying abnormal sources that impact product quality and yields [1,10,11].

Traditionally, the defect patterns on product surfaces can be recognized by using various visual inspection and image processing techniques with the help of high resolution camera [12]. However, this method lacks flexibility because it requires prior knowledge of all the possible defect shapes. As a result, various machine learning methods have been used effectively in detection and recognition of wafer map defects in recent years, which can be divided into unsupervised and supervised learning categories.

Unsupervised learning without using class labels is an effective method to allow computer to capture various patterns from dataset automatically when those new defect patterns are exhibited and need to be added online [2]. Artificial neural

* Corresponding author.
E-mail addresses: jbyu@tongji.edu.cn (J. Yu), xiaoyuncloudy@163.com (X. Zheng), bang_liujt@163.com (J. Liu).

https://doi.org/10.1016/j.compind.2019.04.015
0166-3615/© 2019 Elsevier B.V. All rights reserved.

networks (ANNs) are popular due to their simplicity and powerful ability to deal with nonlinear and multi-dimensional problems. Those typical unsupervised learning-based ANNs, e.g., self-organizing-maps (SOM) [3], adaptive resonance theory network (ART) [9], multistep ART [13], contextual-Hopfield ANN [14] have been employed to recognize defect patterns existing on wafer maps. Other clustering methods, e.g., spatial statistics and hierarchical clustering [15], hidden Markov tree [16], mean-shift [17] were also used to identify various defects of wafer maps. Wang [18] proposed a spatial defect pattern diagnosis model that used a spatial filter to extract defect clusters and then constructed a decision tree (DT) to identify specific defect types on wafer maps. A hybrid approach [19] was proposed to separate composite defect patterns of wafer maps with support vector clustering. Kim et al. [20] proposed a systematic approach for detecting and clustering mixed-type defect patterns, which can effectively cluster complex shapes of defect patterns on wafer maps. The advantages of unsupervised learning are clear: no subjective recognition is needed and, being completely automatic, the method is able to recognize unsuspected defects that could have been overlooked by a human operator.

Supervised learning-based recognizers constructed by the labeled training data have been applied widely for WMPR. These typical recognizers consist of back-propagation network (BPN) [21], support vector machines (SVM) [22,23]. Other recognizers (e.g., K-nearest-neighbor (KNN) [24], decision tree [25] are also used for defect classification. Although these typical supervised recognizers (e.g., ANN, SVM) achieved some successes in recognition of wafer map defects, they still suffer from various limitations in real-world applications.

In general, it is essential to implement feature generation from wafer maps to avoid dealing with high dimensional image. Currently, some studies focus on generation of various features from wafer maps, which generally consists of geometric features, texture features, etc. [2,8,26]. However, the feature set generated from wafer maps is still high-dimensional and consists of much noise that affects effectiveness of recognizers. Thus, feature selection is often performed for recognizers to provide discriminative information for various defect patterns on wafer maps. However, the effective features are manually selected according to the specific wafer maps patterns recognition issue, but this brings generally many difficulties for real-world applications. Thus, feature learning from image signals directly is critically needed to automatically capture the effective pattern features, so that the recognizers can be generalized to different cases without making significant modifications.

Recent progress in the development of machine learning, i.e., deep learning has been receiving an ever-growing attention in that they are increasingly able to automatically extract features with multiple levels of abstraction from large amounts of data. Deep learning, also known as deep neural networks (DNNs), is an effective feature extraction processing model with multiple hidden layers of representations. Considering the capability of DNNs to address large scale data and learn high-level representations, various DNNs, e.g., convolutional neural network (CNN), deep belief network (DBN), and autoencoder (AE) have been used widely for solving many challenging problems, e.g., image classification [27], face recognition [28], speech recognition [29].

CNN is one of the effective deep learning models for various image pattern recognitions, which is able to learn a hierarchy of features from the image input by automatically updating the filters during training on massive amounts of the image data. A deep CNN architecture [30] was proposed to detect defects in industrial inspection, which takes various types of defect free and defective samples together as the inputs. Nakazawa and Kulkarni [31] employed CNN for wafer map defect pattern classification and image retrieval. Kyeong and Kim [32] used CNNs to classify mixed-type defect patterns in wafer bin maps. Lee et al. [33] proposed a CNN model with automatic feature extraction for fault diagnosis in semiconductor manufacturing processes. It is appropriate for chemical process control in wafer fabrication such as etching and chemical vapor deposition. However, the hybrid deep learning with unsupervised pre-training along with supervised DNNs has not been tried on WMPR so far.

Sparse auto-encoder can be effectively used for unsupervised feature learning on a dataset when the class label information is not available. Meanwhile, denoising autoencoder can minimize the error in reconstructing the input from a stochastically corrupted transformation of the input. Lee et al. [34] proposed the use of a stacked denoising autoencoder (SDAE) to establish a fault detection and classification model for wafer defect detection with sensor measurement noise. Both sparse auto-encoder and denoising autoencoder are common fully connected networks, which cannot scale well to high-dimensional inputs in terms of computational complexity.

CNN can be combined with AEs to obtain global features without training on full size images. Masci et al. [35] proposed convolutional auto-encoders that directly takes the 3-D image data as the input and trains the auto-encoder convolutionally. Adjacent convolutional auto-encoders is combined by the convolution and pooling operations. The convolutional kernels are learned to convolve input feature maps of each layer into more abstract features. Luo et al. [36] proposed a convolutional sparse auto-encoder, in which the structure of convolutional auto-encoders was leveraged and the max-pooling was incorporated to heuristically sparsify the feature maps for feature learning. Except for AEs, other types of unsupervised DNNs with convolution structure have achieved great performance for learning representations in visual tasks. Lee et al. [37] proposed convolutional DBN that uses convolution structure to combine the layers to construct hierarchical models [38]. Compared to the regular DBN [39], convolutional DBN preserves information of local relevance and improves the capability of feature representation.

This important issue (i.e., feature learning) is not still investigated well on WMPR. In this study, we develop a wafer map defect detection and recognition system based on a new hybrid deep learning method called stacked convolutional sparse denoising auto-encoder (SCSDAE), which stacks well-designed full connected SDAE in a convolutional way to obtain much more effective feature representations. The main contributions of the paper are following: (1) The hybrid deep learning structure will improve feature learning performance of SCSDAE for WMPR. This enables SCSDAE to generate effective discriminant features from wafer maps directly; (2) Different from SDAE and CNN-based feature learning, SCSDAE uses unsupervised learning-based convolutional auto-encoder to finish noise filtering and feature extraction tasks; (3) The effectiveness of the proposed system is investigated by the experimental results on simulation and real-world wafer maps with a comprehensible performance evaluation. The experimental results show that SCSDAE outperforms these regular DNNs (e.g., CNN, SDAE, DBN) and other typical classifiers (e.g., KNN, BPN, SVM, DT) for WMPR. This study provides the guidance for applications of hybrid deep learning in semiconductor manufacturing process control.

The rest of the paper is organized as follows. Section 2 introduces the deep learning techniques. Section 3 proposed the SCSDAE-based WMPR method. In Section 4, a simulation case-based experimental analysis is performed to verify the effectiveness of the proposed method. The proposed method is further applied to an industrial case in Section 5. Finally, concluding remarks are given in Section 6.

2. Deep learning

Originated from ANN, DNN involved a set of models that attempt to learn high-level representations from the given data with very deep neural networks, typically deeper than three layers. A DNN essentially consists of an input layer, an output layer and some hidden layers stacked between input layer and output layer. The network is firstly layer-wise initialized via unsupervised training, and then tuned in a supervised manner. DNNs use various layers with a limited number of nodes to realize highly nonlinear functions, which can capture essential statistical regularity presentation from the data itself. The representation features can be generated by DNNs for classification, regression and specific problems in information retrieval. The independence from the prior knowledge and human effort in feature design is a major advantage for deep learning.

2.1. Convolutional neural network

CNN is a type of feed-forward neural network defined by a set of convolutional and fully connected layers. CNN can be trained by using large collections of diverse images. It is capable of learning important feature representations from a large benchmark dataset consisting of more than one million images, such as ImageNet [40]. These feature representations often improve performance of those recognizers in image and video recognition.

As shown in Fig. 1, a typical structure of CNN consists of a feature extractor that is composed of several convolutional layers followed by pooling layers and a softmax classifier. The lower-layers are composed to alternating convolution and pooling layers. The convolutional layer extracts signal features, whereas the pooling layer reduces the input dimension and thus further reduces the computation time. The upper-layers however are fully-connected and correspond to a traditional multi-layer perceptron (i.e., hidden layer and logistic regression). The extracted features are then put into the top softmax layer for classification.

2.2. Stacked sparse denoising autoencoder

AE is a symmetrical neural network that can learn effective features in an unsupervised manner by minimizing reconstruction errors [41]. AE consists of an encoder and a decoder. The encoder encodes an input by mapping it to a hidden representation through a deterministic nonlinear function, and then the decoder reconstructs the input by mapping the hidden representation back to the original input space. Sparse auto-encoder [42] is an effective unsupervised feature learning algorithm that optimizes the network weights by minimizing the network reconstruction error between the input data and the reconstructed data. Sparse auto-encoder can learn relatively sparse features by introducing a sparse penalty term inspired by the sparse coding [43] into the auto-encoder. The SDAE is a simple but effective extension of sparse auto-encoder. Fig. 2 presents the structure of SDAE. It reconstructs the input from a corrupted version by manual addition with random noise, which not only prevents AE from just simply learning an identity mapping between input and reconstructed output, but also captures more informative hidden patterns and obtains effective representations from noisy data. Several DAEs can be stacked to form a deep hierarchy and learn high-level representations by feeding the outputs of the l -th layer as inputs to the $(l+1)$ -th layer.

2.3. Dropout

Generally, when the known training data set is small, the overfitting problem will occur in an ANN model. Dropout is an effective technique that can help to reduce overfitting when training an ANN with a limited training dataset [44]. Technically, the dropout can be realized by setting the output of some hidden neurons to zero so that these neurons will not be involved in the forward propagation training process. It should be noted that there are some differences between the training process and testing process with dropout. The dropout is turned off during testing, which means the outputs of all hidden neurons will not be masked during testing. This will help to improve the feature extraction and classification performance.

3. Stacked convolutional sparse denoising auto-encoder

In recent years, feature learning in the context of deep learning has attracted considerable attentions in the machine learning field [45]. Feature extraction plays a key role in improving the performance of the recognizers. SCSDAE is an unsupervised feature extraction and classification method for WMPR, which stacks well-designed fully connected SDAEs in a convolutional way to obtain robust feature representations. It takes the full connection between the features and the convolved feature maps, and then extracts effective features for convolution in an unsupervised manner. In this study, the dropout technique is applied to train the SDAE of each layer, which can prevent overfitting, and improve the subsequent classification performance.

The overall architecture of the SCSDAE is optimized by layer-wise training. An input image is first sent to the convolutional layer to be convolved into feature maps with filters learned by the SDAE, and sub-sampled by pooling operation to get smaller feature maps. The feature maps are further sent to the next convolutional layer. By layers of mapping, the final pooled feature maps are reshaped

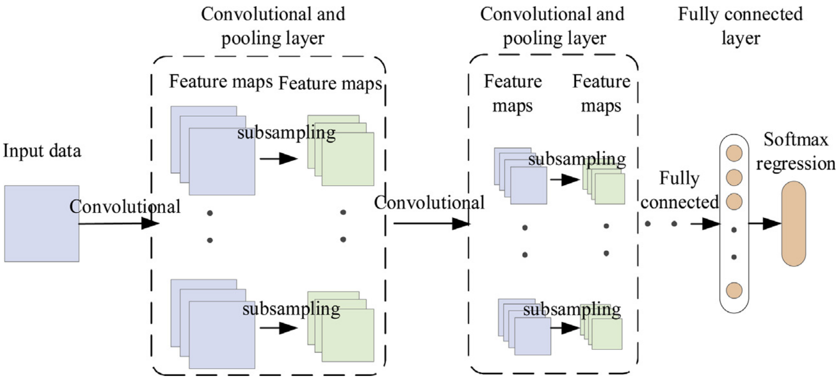


Fig. 1. A typical architecture of CNN.

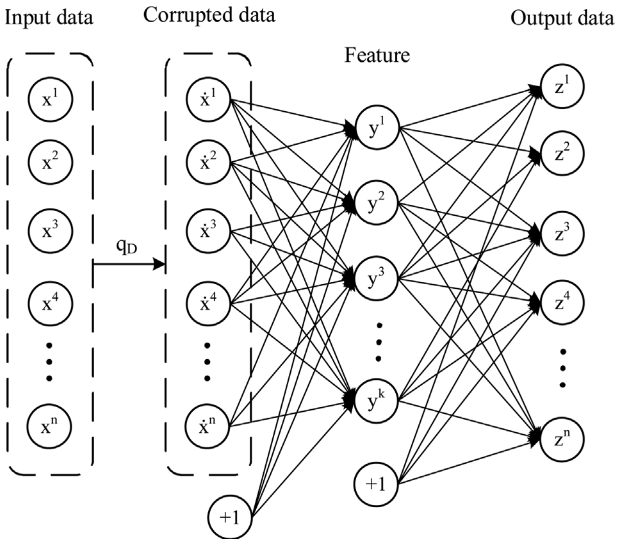


Fig. 2. A typical structure of SDAE.

into discriminative input vectors of the SVM classifier. Fig. 3 illustrates structure of SCSDAE. The big blue grid represents the original input image, and the black square represents the features extracted by SDAE. The pooled feature map 1 is the output of the first CSDAE, also used as the input image to convolved by the second CSDAE feature extractors.

The proposed SCSDAE can be divided into three main steps: two unsupervised CSDAE learning steps and a classification step, and each unsupervised step can be subdivided into three sub-stages, including feature extraction, convolution, and max pooling.

3.1. Feature extraction of the first CSDAE

In this step, SDAE is performed as a feature extractor that maps an input vector x^i to a new feature vector with the K features, where K is the number of hidden units of SDAE. SDAE can make the hidden layer to learn important features and prevent it from simply discovering the sparsity. Given an image of size $m \times m \times 3$, the input vector $x^i \in R^D, D = n \times n \times 3$, and the corresponding corrupted version, the input vector is processed by a linear deterministic mapping and a nonlinear activation function. During the encoding stage, the input data are processed as follows:

$$\alpha^i = f(W_1 x^i + b_1) \tag{1}$$

where $W_1 \in R^{K \times D}$ is a weight matrix with K features, and $b_1 \in R^K$ is the encoding bias. The rectified linear units (ReLU) activation function $f(x) = \max(0, x)$ is considered in this study, which has received extensive attention in the field of deep learning [46]. During the decoding stage, a separate linear decode matrix is utilized for the training to make SDAE more applicable and robust.

$$z^i = W_2 \alpha^i + b_2 \tag{2}$$

where $W_2 \in R^{D \times K}$ and $b_2 \in R^D$ express the tied weight matrices of SDAE with K features and decoding biases, respectively.

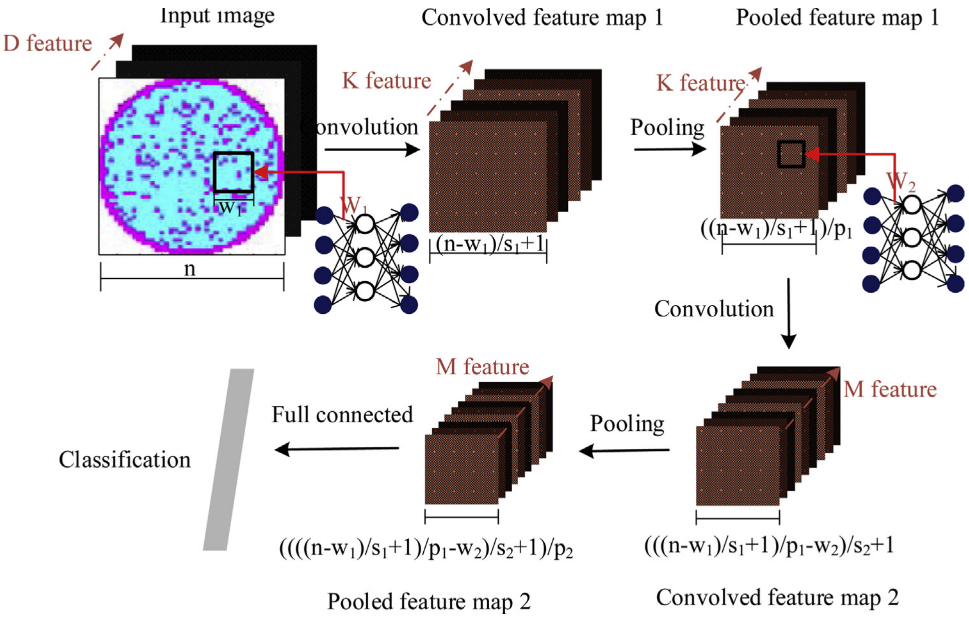


Fig. 3. Illustration of the SCSDAE.

Inspired by the virtues of sparse coding, SDAE is trained to minimize the reconstruction loss function with a sparsity regularization term [47]:

$$J_{SDAE}(X,Z) = \frac{1}{2}\sum_{i=1}^{M}\left|\left|x^i - z^i\right|\right|^2 + \frac{\lambda}{2}(||W_1||^2 + ||W_2||^2) + \beta\sum_{j=1}^{K}KL(\rho||\hat{\rho}_j) \quad (3)$$

$$KL(\rho||\hat{\rho}_j) = \rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \quad (4)$$

where X and Z denote the input and reconstructed data, respectively, λ is the weight decay coefficient, the weight decay term $||W_1||^2 + ||W_2||^2$ is to prevent over-fitting, the norms on $x^i - z^i$, W_1, and W_2 are the L2 norms. β is the sparsity penalization parameter, the sparsity regularization term $KL(\rho||\hat{\rho}_j)$ is Kullback-Leibler divergence [48], $\hat{\rho}_j = \frac{1}{M}\sum_{i=1}^{M}[\alpha^i(x^i)$ is the averaged activation of hidden unit α^i over the training set $X^{D\times M}$, and ρ is the sparsity parameter close to zero. In this stage, the dropout strategy is used to improve the computational efficiency and reduce the over-fitting of SDAE.

Using the feature extraction function $f : R^D \rightarrow R^K$, a new feature representation $f(x) \in R^K$ are transformed from the input patches $x^i \in R^D$. Given an image of size $m \times m$ with 3 channels, and the filter size $w_1 \times w_1$ by step size stride s_1, we can obtain a convolved image with the size $((m - w_1)/s_1 + 1) \times ((m - w_1)/s_1 + 1) \times K$.

To reduce the high dimensionality of the convolved feature maps and save memory, a max pooling [49] is introduced in this stage to select the most representative information within the receptive field. Suppose that the pooling size is p_1, the pooled feature map $Size_{(L1)} \times Size_{(L1)} \times K$ is obtained finally in the first-layer of SCSDAE, where $Size_{(L1)} = ((m - w_1)/s_1 + 1)/p_1$.

3.2. Feature extraction of the second CSDAE

In order to adequately utilize the information of the pooled feature maps from the first layer of CSDAE, the patches of pooled feature maps extracted from the first CSDAE are encoded in the second CSDAE. Assume that the patch sampling strategy and data preprocessing are performed to the input vector $x_1 \in R^K$ for the second layer, the activation of the hidden units in SDAE is computed as follows:

$$a_1 = g(W_{11}x_1^i + b_{11}) \quad (5)$$

In the decoding stage of SDAE, the reconstruction value is obtained with a linear activation function:

$$z_1 = g(W_{22}a_1^i + b_{22}) \quad (6)$$

where W_{11} and W_{22} are the tied weights.

After feature extraction of the second SDAE, the new feature maps are obtained via convolution. Suppose that the filter size in the second-layer CSDAE is $w_2 \times w_2$ with stride s_2, the input size of the second-layer CSDAE is $Size_{(L1)} \times Size_{(L1)} \times K$, and the convolved feature maps $((Size_{(L1)} - w_2)/s_2 + 1) \times ((Size_{(L1)} - w_2)/s_2 + 1) \times M$ with M channels are obtained.

Then max pooling in the second-layer CSDAE is imposed to obtain the pooled feature maps $Size_{(L2)} \times Size_{(L2)} \times M$ with the pooling size of p_2, where $Size_{(L2)} = [((Size_{(L1)} - w_2)/s_2 + 1)/p_2]$.

3.3. Classification of SCSDAE

Finally, the pooled feature maps are used to construct the SVM classifier to predict the class label of the input wafer maps. SVM uses various kernels to transform the original input space into high dimensional feature space. The common kernel functions are linear, polynomial, radial basis function (RBF, also called Gaussian kernel), and sigmoid. Because the number of features is very large, we use a linear kernel for the SVM in the classification layer of SCSDAE.

3.4. Application procedure of SCSDAE

The whole system framework based on SCSDAE for wafer maps detection and recognition is presented in Fig. 4. The off-line modeling phase is to learn the discriminative representative features and train a SCSDAE recognizer through unsupervised representation learning and supervised training with class labels. In the online testing phase, the input wafer maps are fed into the well-trained SCSDAE to provide final detection and recognition results. The detailed procedure of the proposed method is summarized as follows:

Off-line modeling: This part consists of the following six steps:

Step 1: Collect wafer maps for each of the normal and defect patterns to generate the training dataset.

Step 2: Perform feature learning based on SDAE in an unsupervised learning manner, and then obtain high-level feature representations from the training dataset;

Step 3: Convolution is utilized to extract appropriate and sufficient features;

Step 4: Max pooling is utilized to obtain representative features;

Step 5: Pooled feature maps in the first CSDAE are passed to the second CSDAE in an unsupervised manner;

Step 6: Finally, the pooled feature maps are used to construct the SVM classifier.

Online defect detection and recognition: This part consists of the following four steps:

Step 1: Sample the wafer maps from the real fabrications;

Step 2: Input the wafer map to the SCSDAE model constructed in the off-line phase;

Step 3: After feature learning of SCSDAE in unsupervised manner, the pooled features are fed into the SVM classifier to predict the class label;

Step 4: Report the defect pattern by using the constructed SVM.

4. A simulation case

It is often difficult to assess the performance of recognition models based on the real wafer data because the real wafer patterns may be unknown. Thus, the effectiveness of SCSDAE is verified firstly on a simulated dataset that consists of nine different wafers patterns. Simulation does provide an effective platform from which investigations into potential problems associated with various wafer map patterns in a semiconductor manufacturing process can begin.

By following the method provided in [50], the wafer map defect patterns are generated using a probabilistic model of the spatial occurrence of electrical failures. Assumed that a single device independently occurs electrical failure of other ones with a probability, which is a function of the location of the device in the wafer: $P(failure) = p(x,y)$, where x and y are the planar coordinates of the center of the device. The different wafer map patterns can be generated by the different profiles of the failure probability $p(x,y)$ as a function of the planar coordinates (x,y). The expressions of the spatial probability $p(x,y)$ corresponding to nine typical patterns are reported in [50]. We generate random noise on the surface of all wafer maps.

Fig. 5 presents the wafer map examples with normal and defect patterns. The simulation dataset consists of one normal pattern and eight typical defect patterns i.e., Center, Donut, Edge-local,

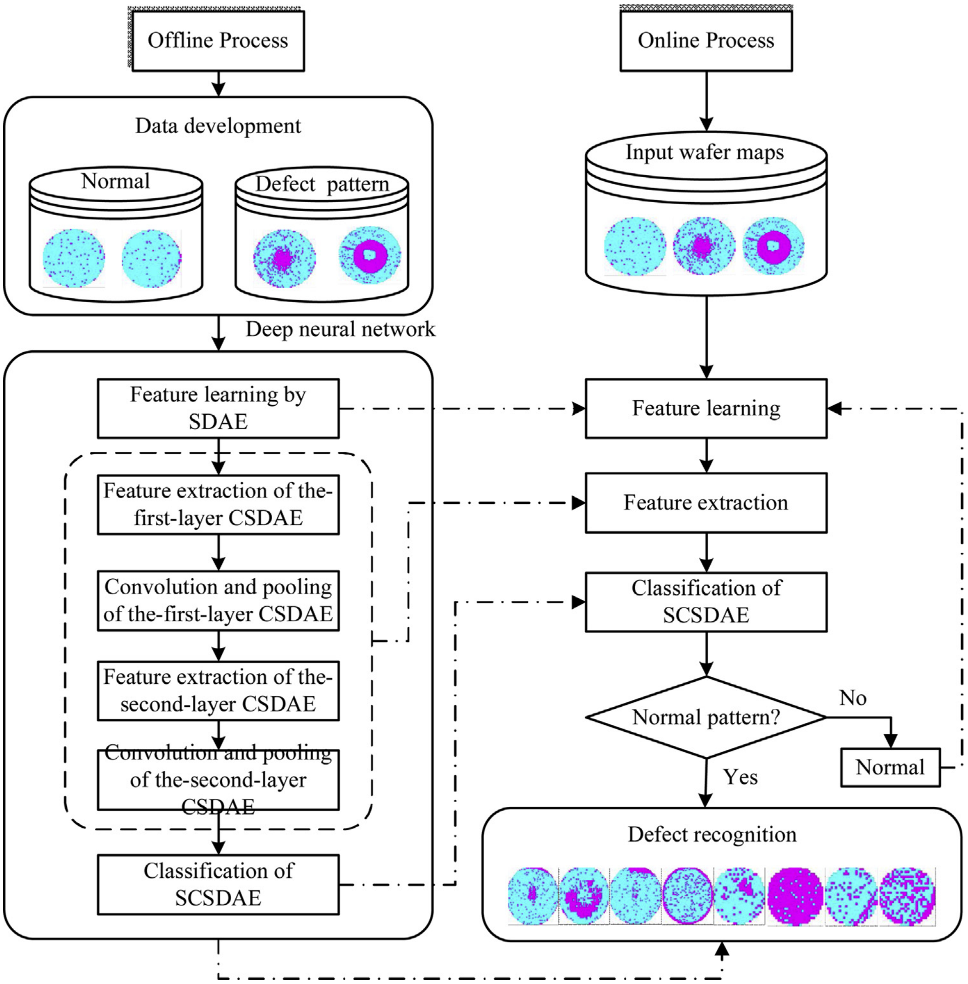


Fig. 4. Schematic diagram of the proposed wafer map defect detection and recognition system.

Edge-ring, Local, Near-full, Random, Scratch and the normal none-pattern. To verify the effectiveness of the proposed method, 1300 samples for each wafer map are generated by using the simulation method. Finally, total 11,700 (1,300*9) wafer maps will be generated as the dataset.

4.1. Recognition performance analysis

This section will evaluate effectiveness of SCSDAE for WMPR. The size of each image is preprocessed to 128 × 128 pixels. Then, we separate these images randomly into training dataset with 6,300 maps, validation dataset with 2,700 maps, and testing dataset with 2,700 maps. After achieving the desired training/validation accuracy on the training and validation images, the testing images are used to evaluate the WMPR performance of SCSDAE.

We trained the SCSDAE model with two layers for WMPR. The detailed structure information and parameters about the SCSDAE model is shown in Table 1. The 5,000 samples from the training set were used to train the first-layer of SCSDAE with a total of 30 epochs and learning rate $\delta = 0.001$. We trained the second-layer based on the $14 \times 14 \times 32$ outputs of the first layer with a total of 20 epochs and learning rate $\delta = 0.0001$. Finally, we used the training and validation dataset for supervised training. In this study, these key parameters are determined by using the try-error method on the training and validation in advance. We fine-tuned the network on the training set until the recognition error on the validation set is stabilized (see Fig. 6). We finally retrained the network from scratch on the 6,300 samples with the same learning rates and epochs and test its performance on the testing set.

Table 2presents the overall recognition rate (%) in the combined confusion matrix of SCSDAE on the testing dataset. The diagonal elements in this matrix are the recognition rates of each WMP. SCSDAE obtains an overall accuracy rate 94.81%. Except for donut and local defect, SCSDAE recognizes more than 94% WMPs for other defect patterns. This good recognition performance of SCSDAE owes to the deep learning structure and hybrid feature learning. It is clear that the most errors are from misrecognizing Edge-local into Edge-ring. It can be seen from Fig. 7 that Edge-ring and Edge-local were frequently confused with other defect patterns. In general, we will accept these misclassification results because these wafer maps seem to saddle across the boundary of two patterns. This indicates that SCSDAE would satisfy with the requirements of users, as indicated by the overall recognition accuracy.

4.2. WMPR performance comparison

These typical recognizers, i.e., C4.5, KNN, BPN, SVM and DNNs (i.e., SDAE, CNN and DBN) that are often employed for WMPR were also considered for performance comparison with SCSDAE. The 6, 300 wafer maps were used as the training data of these

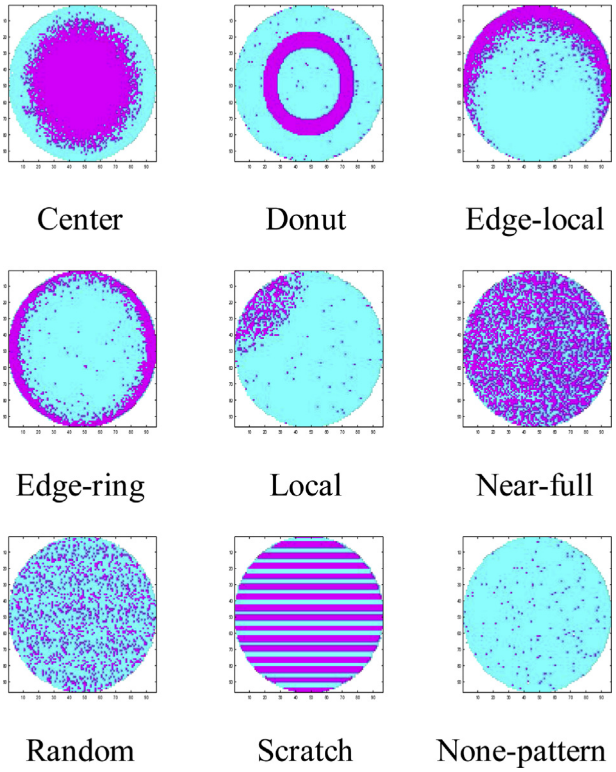


Fig. 5. List of simulation wafer map defect patterns.

Table 1
Network architecture of SCSDAE.

Input	Layer 1				Layer 2				Full connect	Output
	Filter size	Feature maps	Stride	Pooling size	Filter size	Feature maps	Stride	Pooling size		
128 × 128	7 × 7	32	3	3	5 × 5	64	3	3	256	9

recognizers. The 53 features [2] generated from wafer maps were used as the inputs of SVMs.

The deep learning toolbox, i.e., Tensorflow [51] is used to train AlexNet [52]. The structure of SDAE is setup as follows: the number of the input, two hidden and output nodes is 9216, 4000, 1000 and 9, respectively. Learning rate is set to 0.0001 and training epochs are 50. The BPN used the same network structure of SDAE. The number of iterations and learning rate of BPN were 500 and 0.1, respectively. The multiclass SVM was used from LIBSVM. SVM uses various kernels to transform the original input space into high dimensional feature space. The common kernel functions are linear, polynomial, radial basis function (RBF, also called Gaussian kernel), and sigmoid. To further assess the recognition performance of SVM, we conducted two SVMs by using different types of kernel functions, i.e., SVM with a linear kernel (SVML), and SVM with a Gaussian kernel (SVMG). For KNN, the value of K is set to 10 according to a prior validation on the training dataset. For C4.5, incorrectly assigned sample at a node was set to 8.

Table 3 presents performance comparison between SCSDAE and other recognizers based on five-fold cross validation. The ACC and SD is the average recognition rate and the corresponding standard deviation of the different recognizers based on the five-fold cross validation, respectively. It can be seen from Table 3 that SCSDAE and other DNNs (i.e., SDAE, CNN and DBN) show the obvious better result than that of these typical recognizers (i.e., C4.5, KNN, BPN, SVM). SCSDAE outperforms all other recognizers significantly. These comparison results illustrate that SCSDAE is very powerful for WMPR. The main reasons for success of the proposed SCSDAE-based WMPR model are that: the hybrid deep learning technique is capable of adaptively learning the representative features from the images through multiple non-linear transformations; It is able to learn the complex non-linear relationships between the wafer maps and the different patterns through encoding higher level network structure with unsupervised and supervised learning.

Apart from the recognition accuracy comparison, these evaluation measures (i.e., rand accuracy, precision, recall and F1 score) on the five-fold cross validation are also performed and the results are listed in Table 4. The ACU is the rand accuracy that defines how many of the positive samples are true positive samples. The PRE is the precision that defines the number of correct positive results divided by the number of all positive results returned by the classifier, while REC is the recall rate that defines the number of correct positive results divided by the number of all relevant samples. The F1 score $F1 = (2 \cdot PRE \cdot REC)/(PRE + REC)$ is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. It can be found that SCSDAE has better performance than that of the other recognizers.

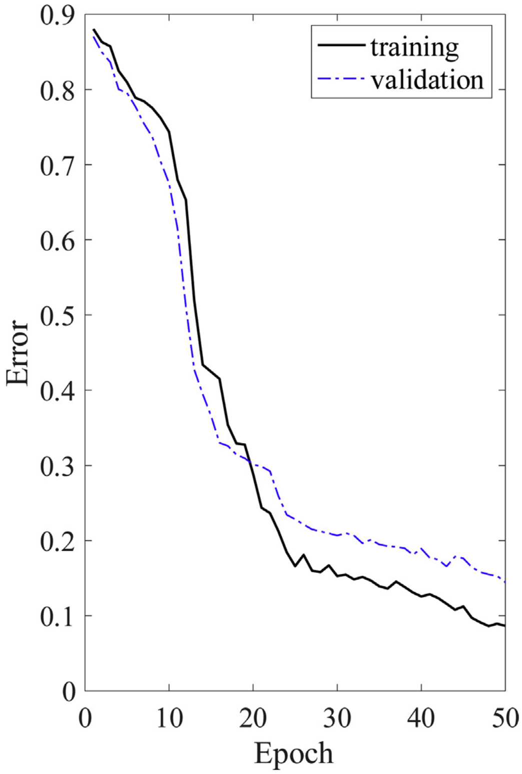
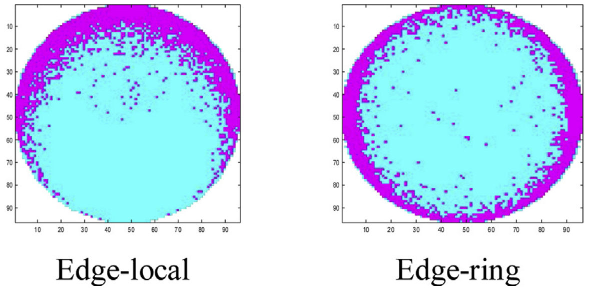


Fig. 6. The training and validation process of SCSDAE.

4.3. Parameter sensitivity analysis

The performance of SCSDAE is affected significantly by the structure and parameters of the network itself. The parameter sensitivity analysis is further performed to illustrate influence of these key parameters (i.e., activation units and network structure) on the SCSDAE model.

(1) Activation units: SCSDAE explicitly sparsifies feature maps by retaining the maximum value in each local subregion. The activation units that are capable of preserving much information of the input are preferred to explore the main factors of variation embedded in the image data. The two activation units, i.e., rectified linear units (ReLU) and LgSU are considered in SCSDAE. ReLU allows a network to easily obtain sparse representations. Fig. 8 presents the reconstruction images to analyze the encoding ability of different activation units in SCSDAE. It can be seen that ReLU has a strong ability to capture the main features of variation embedded in the image in comparison with LgSU that fails to learn important features. ReLU shows good reconstruction ability, because the sparsity operation is implemented in each local subregion.



Edge-local Edge-ring

Fig. 7. The wafer maps with Edge-local and Edge-ring pattern.

(2) Network structure. In general, the network structure (in particular, the depth of the architecture) of SCSDAE has a significant affection on its generalization performance. Thus, a sensitivity analysis to different network structures is performed to determine the appropriate structure. Table 5 compares the performance of single-layer, two-layer CSDAE and AlexNet with different dropout rates. The dropout is used in the fully-connected layer of the three models.

From Table 5, we find that the two-layer CSDAE can achieve comparable performance with that of the single-layer CSDAE and AlexNet on the simulation wafer dataset. The results show that the best recognition performance of SCSDAE was obtained at a dropout rate close to 0.9 and the performance would decrease when the dropout rate is more than 0.9. This indicates that dropout can improve the performance of SCSDAE, but too much dropout may loss some important neurons for feature representation. It confirms that SCSDAE is capable of learning more effective features than the typical CNNs, e.g., AlexNet.

(3) Classifier. In general, a typical CNN structure consists of a feature extractor that is composed of several convolutional layers usually followed by pooling layers and a Softmax classifier. However, these studies [53] consider other classifiers (e.g., SVM) in the classification layer of a CNN. Thus, a comparison between Softmax and SVMs that are used as classifier in the classification layer of SCSDAE is performed to determine the appropriate classifier. In SCSDAE$_{LK}$, SCSDAE$_{RBF}$, and SCSDAE$_{PK}$, SCSDAE is constructed by feeding an SVM classifier with a linear, RBF, and polynomial kernel on the 256 pooled features respectively. In SCSDAEnorm, Softmax is used as the classifier in the last layer of SCSDAE. Table 6 compares the performance of SCSDAE$_{norm}$, AlexNet, SCSDAE$_{LK}$, SCSDAE$_{RBF}$ and SCSDAE$_{PK}$ with dropout rates equal to 0.9, From Table 6, we find that the SCSDAE using SVM can achieve comparable performance with that of the other methods on the simulation wafer dataset. The results show that the best recognition performance of SCSDAE$_{LK}$ was obtained by using SVM with a linear kernel. This indicates that SCSDAE improves the accuracy of SCSDAE$_{norm}$ by using SVM as a classifier.

Table 2
Simulation case: confusion matrix of SCSDAE for WMPR.

	Center	Donut	Edge-local	Edge-ring	Local	Near-full	None	Random	Scratch
Center	98.57	0	0	0	0	1.43	0	0	0
Donut	2.86	92.86	1.43	2.86	0	0	0	0	0
Edge-local	0	0	95.71	4.29	0	0	0	0	0
Edge-ring	1.43	0	2.86	95.71	0	0	0	0	0
Local	0	0	2.86	2.86	92.86	0	0	1.43	0
Near-full	1.43	0	0	0	0	95.71	0	2.86	0
None	0	0	0	1.43	2.86	0	95.71	0	0
Random	2.86	0	0	0	1.43	1.43	0	94.29	0
Scratch	0	0	1.43	0	0	2.86	0	1.43	94.29

Table 3
Simulation case: Five-fold cross validation of SCSDAE and other recognizers.

Classifier	SCSDAE	AlexNet	SDAE	DBN	SVML	SVMG	BPN	KNN	C4.5
ACC (%)	**95.13**	92.83	90.37	88.89	89.76	91.32	90.02	88.03	88.65
SD	0.25	0.37	0.27	0.09	0.60	0.54	0.63	0.50	0.64

Table 4
Simulation case: Evaluation measures (%) of SCSDAE and other recognizers.

		Center	Donut	Edge local	Edge ring	Local	Full	None	Random	Scratch
SCSDAE	ACU	98.74	99.04	98.74	98.37	98.74	98.89	99.52	98.74	99.37
	PRE	90.80	100.00	92.09	90.25	95.55	94.41	100.00	94.33	100.00
	REC	98.67	91.33	97.00	95.67	93.00	95.67	95.67	94.33	94.33
	F1	94.57	95.47	94.48	92.88	94.26	95.03	97.79	94.33	97.08
AlexNet	ACU	98.19	98.78	98.07	97.67	98.19	98.37	99.26	98.11	99.00
	PRE	87.69	100.00	89.24	86.92	92.83	91.83	100.00	91.09	98.58
	REC	97.33	89.00	94.00	93.00	90.67	93.67	93.33	92.00	92.33
	F1	92.26	94.18	91.56	89.86	91.74	92.74	96.55	91.54	95.35
SDAE	ACU	97.22	98.19	97.52	96.93	97.85	97.48	98.96	97.78	98.81
	PRE	82.99	96.31	85.85	83.38	92.01	87.42	100.00	90.54	99.26
	REC	94.33	87.00	93.00	90.33	88.33	90.33	90.67	89.33	90.00
	F1	88.30	91.42	89.28	86.72	90.14	88.85	95.10	89.93	94.41
DBN	ACU	96.81	98.19	96.93	96.67	97.44	97.22	98.89	97.30	98.33
	PRE	80.92	98.08	83.38	82.01	89.42	85.94	100.00	88.47	97.40
	REC	93.33	85.33	90.33	89.67	87.33	89.67	90.00	87.00	87.33
	F1	86.69	91.27	86.72	85.67	88.36	87.77	94.74	87.73	92.09
SVML	ACU	97.00	98.48	97.33	96.96	97.56	97.26	98.96	97.48	98.52
	PRE	81.74	99.24	86.08	83.64	89.53	85.99	100.00	88.67	98.15
	REC	94.00	87.00	90.67	90.33	88.33	90.00	90.67	88.67	88.33
	F1	87.44	92.72	88.31	86.86	88.93	87.95	95.10	88.67	92.98
SVMG	ACU	97.44	98.78	97.85	97.48	97.93	97.48	99.15	97.74	98.81
	PRE	84.07	100.00	89.03	86.02	90.94	86.71	100.00	90.24	98.91
	REC	95.00	89.00	92.00	92.33	90.33	91.33	92.33	89.33	90.33
	F1	89.20	94.18	90.49	89.07	90.64	88.96	96.01	89.78	94.43
BPN	ACU	97.19	98.52	97.44	96.96	97.59	97.30	98.96	97.56	98.56
	PRE	82.75	99.24	86.67	83.64	89.56	86.26	100.00	88.74	98.15
	REC	94.33	87.33	91.00	90.33	88.67	90.00	90.67	89.33	88.67
	F1	88.16	92.91	88.78	86.86	89.11	88.09	95.10	89.04	93.17
KNN	ACU	96.56	97.93	96.70	96.33	97.33	96.96	98.81	97.22	98.22
	PRE	79.83	96.92	82.46	80.55	89.04	84.71	100.00	88.14	96.67
	REC	92.33	84.00	89.33	88.33	86.67	88.67	89.33	86.67	87.00
	F1	85.63	90.00	85.76	84.26	87.84	86.64	94.37	87.39	91.58
C4.5	ACU	96.67	98.04	96.81	96.52	97.52	97.15	98.85	97.44	98.33
	PRE	80.17	97.68	82.82	81.60	89.76	85.62	100.00	89.42	96.70
	REC	93.00	84.33	90.00	88.67	87.67	89.33	89.67	87.33	88.00
	F1	86.11	90.52	86.26	84.98	88.70	87.44	94.55	88.36	92.15

4.4. Feature visualization

The lower-dimension visualization of the features extracted by SCSDAE is further performed t-distributed stochastic neighbor embedding (t-SNE) [54] is used to transform the extracted features into two-dimension space. The 50 samples were selected for each WMP. Finally, the total 450 samples are selected randomly from the testing dataset for feature visualization of SCSDAE.

Feature visualization is carried out on the original images, images of the two convoluted layers and the last fully-connected layer of SCSDAE. Visualization of the first two vectors extracted by t-SNE is presented in Fig. 9. We can see from Fig. 9 that the raw wafer maps from the nine patterns randomly spread in the dimension mapping with a large overlap under the different patterns. The features of the last fully-connected layer present stronger clustering characteristics than the raw wafer maps. The features become more and more divisible as the layer goes deeper and deeper in SCSDAE, which exhibits very good discriminant characteristics. It is clear that the second convoluted layer separates the nine patterns well. The good performance of SCSDAE

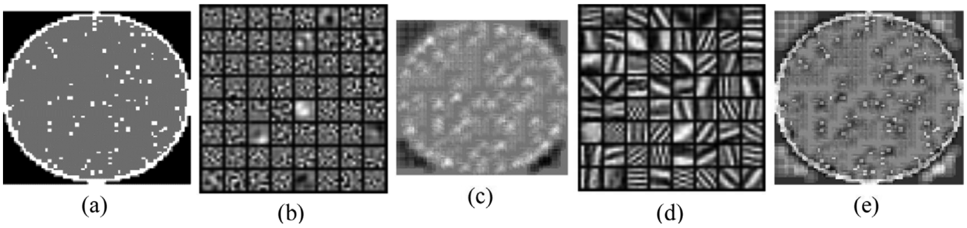


Fig. 8. Feature learned by SCSDAE with different activation units, (a) Original image. (b) LgSU, (c) Reconstructed images using features from (a), (d) ReLU, (e) Reconstructed images using features from (d).

Table 5
Recognition accuracy of SCSDAE and AlexNet with different dropout rates.

Model	Dropout rate				
	0.5	0.7	0.75	0.9	0.95
AlexNet	64.71	75.34	76.36	89.58	93.45
One-layer CSDAE	88.28	86.87	87.43	92.28	88.67
Two-layer CSDAE	91.35	89.26	90.17	94.39	92.31

Table 6
Performance comparison with different classifiers (%).

Model	Accuracy (%)
$\text{SCSDAE}_{\text{norm}}$	92.63
AlexNet	89.58
$\text{SCSDAE}_{\text{LK}}$	94.39
$\text{SCSDAE}_{\text{RBF}}$	93.04
$\text{SCSDAE}_{\text{PK}}$	92.37

is because of the strong power of feature learning of the hybrid deep structure. Thus, these visualization results indicate that the proposed model provides the powerful ability in adaptively learning the defect features from the wafer maps.

5. An industrial case

In this section, we further test the WMPR performance of SCSDAE on a real-world industrial dataset (i.e. WM-811 K) that were collected from 46293 lots in a real-world fabrication [8]. A scanner system is used to collect wafer bin maps for WM-811 K. Based on the scanned values on the position of the pixel in the wafer map, cyan, magenta white colors are used for normal, defective and empty elements of each wafer map, respectively. The WM-811 K dataset consists of the normal and eight defect patterns, i.e., Center, Donut, Edge-local, Edge-ring, Local, Near-full, Random, Scratch, and the normal none-pattern (see Fig. 10). The detailed information about the used wafer maps for performing fault detection and recognition is presented Fig. 11. It is clear that the class label imbalance exists in the WM-811 K dataset, which will result in a big challenge for WMPR. The dataset was divided into a

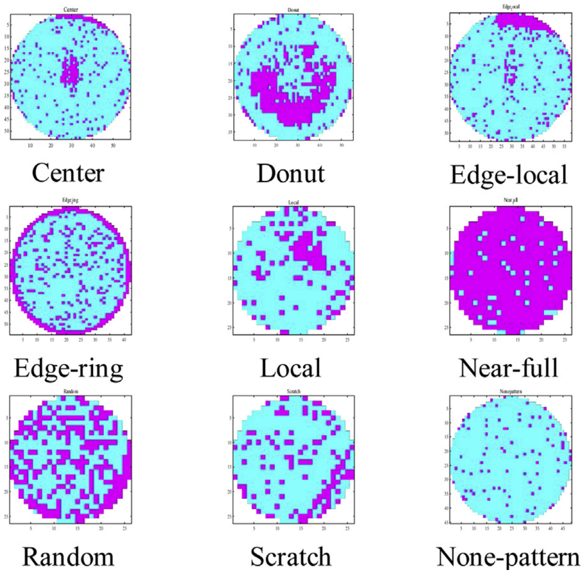
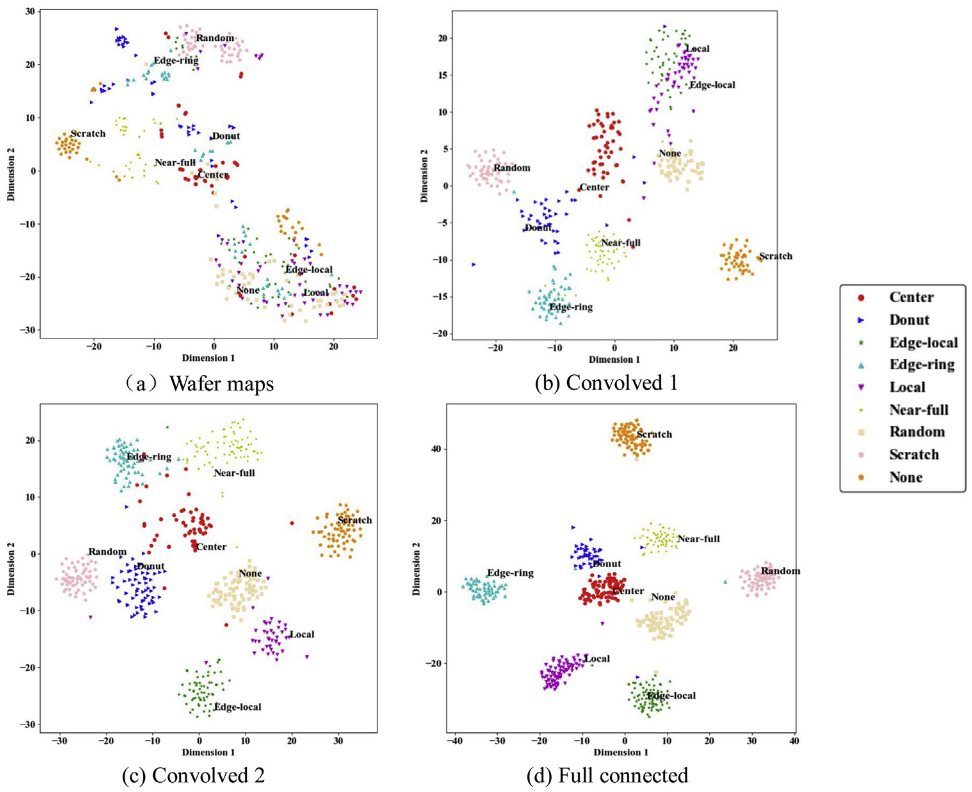


Fig. 10. List of wafer map defect patterns.



（a） Wafer maps

(b) Convolved 1

(c) Convolved 2

(d) Full connected

Fig. 9. Feature visualization via t-SNE for the wafer maps, and learned features of the two convoluted layers and last fully-connected layer of SCSDAE, (a) wafer maps, (b) the first convoluted layer, (c) the second convoluted layer, and (d) the full connected layer.

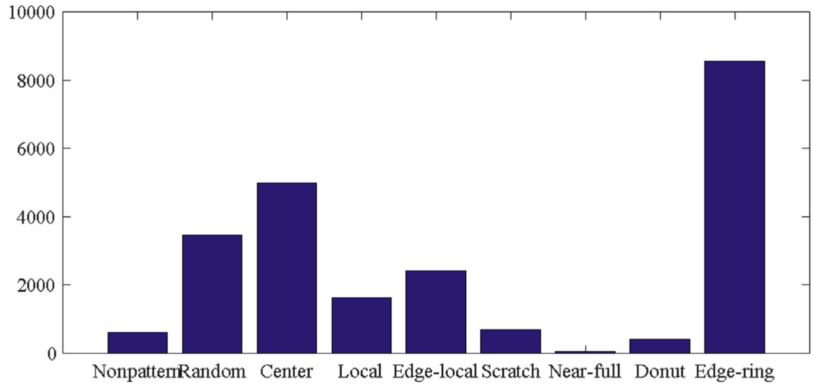


Fig. 11. Wafer map distribution of various patterns in the WM-811 K dataset.

Table 7
Industrial case: Confusion matrix (%) of SCSDAE for WMPR.

	Center	Donut	Edge-local	Edge-ring	Local	Near-full	None	Random	Scratch
Center	95.57	0	2.50	1.15	0.48	0	0.29	0	0
Donut	5.69	82.93	0.00	6.50	0	0	0	4.88	0
Edge-local	1.10	0	96.14	1.38	0.28	0	1.10	0	0
Edge-ring	0.86	0	0.70	96.18	0.78	0	1.48	0	0
Local	0.17	0	0.12	0.19	95.30	0	0.41	0.06	0
Near-full	0	0	0	0	0	87.50	0	12.50	0
None	0.07	0	0	3.00	0	0	96.73	0	0.20
Random	2.73	1.64	2.19	1.09	1.64	0.55	1.64	88.52	0
Scratch	2.40	0	1.92	1.92	0	0.48	2.40	0.96	89.90

training and testing set (with the radio of 7:3) to construct the SCSDAE model and then to test its performance, respectively.

Table 7 presents the overall recognition rate (%) for WMPR results of SCSDAE. The results are calculated by the average of the diagonal elements of a respective confusion matrix. SCSDAE shows a very good performance for WMPR because it achieves 94.75% accuracy rate. Although a big imbalance in the 9 classes existed, SCSDAE shows very good performance for those small classes (e.g. Donut, Near-full). In particular, the 87.50% of the Near-full defects can be recognized effectively based on the only 54 wafers for training. This indicates that SCSDAE would satisfy with the requirements of users, as indicated by the overall recognition accuracy. In SCSDAE, the hybrid learning scheme, i.e., convolutional auto-encoder-based feature learning might improve the recognition performance of the SVM classifier for the small classes in this dataset.

Table 8 shows the recognition accuracy and the corresponding standard deviation of SCSDAE and other typical DNNs as well as the other regular recognizers. It can be observed from Table 8 that SCSDAE performs better than the other classification methods. These comparison results illustrate that SCSDAE is very effective for WMPR in very complex working conditions.

Table 9 lists the evaluation measures results among SCSDAE and other recognizers. It can be observed from Table 9 that SCSDAE performs better than the other classification methods. This further demonstrates that SCSDAE is capable of learning abstract and complicated orientation, structural, and detailed information to identify wafer map patterns in a real-world semiconductor manufacturing process.

In order to handle imbalanced classification problem, the under-sampling and over-sampling methods are generally used to against highly imbalanced datasets [55]. Under-sampling randomly eliminates majority class examples, while over-sampling increases the number of instances in the minority class. Both of them aim to obtain approximately the same number of instances of the classes.

The two methods are further used to discuss the effectiveness of SCSDAE for solving the class imbalance problem. Based on the learned features by SCSDAE from the training dataset, in the case of under-sampling, we take 60 samples without replacement from majority patterns (i.e., Center, Donut, Edge-local, Edge-ring, Local, Random, Scratch, and None pattern), then combine them with minority pattern (i.e., Near-full pattern). Total samples in the new dataset after under-sampling is 540 (60*9). In this case of over-sampling, we replicate minority patterns (i.e., Near-full and Donut) for 600 sample, then randomly take 600 samples from other wafer patterns. Total samples in the new dataset after over-sampling are 5,400 (600*9).

The recognition rate comparison of SCSDAE that uses the original dataset, under-sampling dataset, and over-sampling dataset is presented in Table 10. It is clear that these features extracted from the last fully connected layer of SCSDAE improve the recognition performance. It can be observed from Table 8 that the performance of SCSDAE with over-sampling outperforms SCSDAE with under-sampling and the original dataset. Over-

Table 8
Industrial case: Five-cross validation of SCSDAE and other recognizers.

Classifier	SCSDAE	AlexNet	SDAE	DBN	SVML	SVMG	BPN	KNN	C4.5
ACC (%)	**94.75**	92.08	91.91	90.69	89.56	90.53	90.02	87.16	88.91
SD	0.20	0.29	0.43	0.28	0.45	0.36	0.58	0.55	0.48

Table 9
Industrial case: Evaluation measures (%) of SCSDAE and other recognizers.

Model	Evaluation metrics	Center	Donut	Edge local	Edge ring	Local	Full	None	Random	Scratch
SCSDAE	ACU	98.44	99.65	98.70	97.17	98.48	99.94	97.98	99.47	99.65
	PRE	94.21	97.14	91.95	96.26	93.21	87.50	94.22	91.53	98.42
	REC	95.57	82.93	96.14	96.18	84.77	87.50	96.73	88.52	89.90
	F1	94.89	89.47	94.00	96.22	88.79	87.50	95.46	90.00	93.97
AlexNet	ACU	97.71	99.24	98.12	95.73	98.13	99.88	97.11	99.07	99.18
	PRE	91.84	83.81	89.31	94.34	90.50	75.00	92.27	83.62	90.00
	REC	93.17	71.54	93.38	94.27	82.30	75.00	94.73	80.87	82.21
	F1	92.50	77.19	91.30	94.31	86.21	75.00	93.49	82.22	85.93
SDAE	ACU	97.65	99.15	98.06	95.62	98.10	99.91	97.08	99.04	99.15
	PRE	91.65	80.95	89.05	94.19	90.27	81.25	92.21	83.05	89.47
	REC	92.97	69.11	93.10	94.12	82.10	81.25	94.67	80.33	81.73
	F1	92.31	74.56	91.03	94.15	85.99	81.25	93.42	81.67	85.43
DBN	ACU	97.30	98.89	97.79	95.35	97.81	99.91	96.58	98.83	98.92
	PRE	90.51	72.38	87.86	93.84	88.01	81.25	91.10	79.10	85.26
	REC	91.82	61.79	91.86	93.76	80.04	81.25	93.53	76.50	77.88
	F1	91.16	66.67	89.82	93.80	83.84	81.25	92.30	77.78	81.41
SVML	ACU	97.06	98.74	97.50	94.83	97.60	99.88	96.14	98.63	98.71
	PRE	89.75	67.62	86.54	93.14	86.43	75.00	90.13	75.14	81.58
	REC	91.05	57.72	90.48	93.06	78.60	75.00	92.53	72.68	74.52
	F1	90.40	62.28	88.47	93.10	82.33	75.00	91.32	73.89	77.89
SVMG	ACU	97.36	98.98	97.74	95.12	97.84	99.94	96.47	98.83	98.80
	PRE	90.70	75.24	87.60	93.53	88.24	87.50	90.84	79.10	83.16
	REC	92.01	64.23	91.59	93.45	80.25	87.50	93.27	76.50	75.96
	F1	91.35	69.30	89.55	93.49	84.05	87.50	92.04	77.78	79.40
BPN	ACU	97.24	98.92	97.59	94.89	97.72	99.94	96.26	98.77	98.71
	PRE	90.32	73.33	86.94	93.21	87.33	87.50	90.39	77.97	81.58
	REC	91.63	62.60	90.90	93.14	79.42	87.50	92.80	75.41	74.52
	F1	90.97	67.54	88.87	93.18	83.19	87.50	91.58	76.67	77.89
KNN	ACU	96.36	98.51	96.79	93.49	97.31	99.91	95.30	98.31	98.33
	PRE	87.48	60.00	83.38	91.34	84.16	81.25	88.25	68.93	74.74
	REC	88.74	51.22	87.05	91.27	76.54	81.25	90.60	66.67	68.27
	F1	88.10	55.26	85.18	91.31	80.17	81.25	89.41	67.78	71.36
C4.5	ACU	96.95	98.86	97.24	94.51	97.49	99.94	95.88	98.60	98.36
	PRE	89.37	71.43	85.36	92.71	85.52	87.50	89.55	74.58	75.26
	REC	90.66	60.98	89.24	92.63	77.78	87.50	91.93	72.13	68.75
	F1	90.01	65.79	87.26	92.67	81.47	87.50	90.72	73.33	71.86

Table 10
Recognition rate comparison (%) of SCSDAE that uses the original dataset, under-sampling dataset, and over-sampling dataset.

Dataset	Evaluation metrics	Center	Donut	Edge local	Edge ring	Local	Full	None	Random	Scratch
Original dataset	ACU	98.44	99.65	98.70	97.17	98.48	99.94	97.98	99.47	99.65
	PRE	94.21	97.14	91.95	96.26	93.21	87.50	94.22	91.53	98.42
	REC	95.57	82.93	96.14	96.18	84.77	87.50	96.73	88.52	89.90
	F1	94.89	89.47	94.00	96.22	88.79	87.50	95.46	90.00	93.97
Under-sampling dataset	ACU	98.12	99.44	98.47	96.49	98.28	99.91	97.55	99.36	99.42
	PRE	93.17	90.48	90.90	95.36	91.63	81.25	93.25	89.27	94.21
	REC	94.51	77.24	95.03	95.28	83.33	81.25	95.73	86.34	86.06
	F1	93.84	83.33	92.92	95.32	87.28	81.25	94.47	87.78	89.95
Over-sampling dataset	ACU	98.58	99.68	98.90	96.99	98.69	99.94	98.01	99.65	99.80
	PRE	94.69	98.10	92.88	96.02	94.80	87.50	94.29	94.92	100.00
	REC	96.05	83.74	97.10	95.95	86.21	87.50	96.80	91.80	92.31
	F1	95.37	90.35	94.94	95.98	90.30	87.50	95.53	93.33	96.48

sampling obtains the balanced classes in the training data and improves the performance of classifiers. In addition, SCSDAE of the original data performs better than that of under-sampling. This indicates that feature extraction plays a key role in handling imbalanced classification problem.

6. Conclusions

In this study, we provided a hybrid deep learning model, i.e., SCSDAE for WMPR, which adopts SDAE as a feature extractor and exploits the full connection between the SDAE features and the subsequent convolved images in an unsupervised manner. The robust and discriminative features from wafer maps through this deep network architecture can be extracted by SCSDAE for WMPR improvement. SCSDAE achieves better recognition results on the simulation and real-word wafer map datasets than those traditional WMPR methods and the other deep learning models, which verifies that SCSDAE is capable of learning effective features from wafer maps. This study provides a new way for WMPR using hybrid deep learning in semiconductor manufacturing process control. However, the problem of limited training data will lead to over-fitting of SCSDAE. Moreover, due to labeled data scarcity for some patterns in wafer maps, it is a challenging issue for SCSDAE to implement WMPR on imbalanced dataset. In the future, a further research about the patch size and pooling mechanism analysis will be carried out to disclose the properties of SCSDAE. In addition, adaptive recognition of SCSDAE for those novel wafer map defect patterns online will be further considered.

Acknowledgement

This work was supported by National Natural Science Foundation of China (71777173), Fundamental Research Fund for the Central University.

References

[1] S.H. Huang, Y.C. Pan, Automated visual inspection in the semiconductor industry: a survey, Comput. Ind. 66 (2015) 1–10.

[2] J. Yu, X. Lu, Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis, IEEE Trans. Semicond. Manuf. 29 (1) (2016) 33–43.

[3] F. Adly, O. Alhussein, P.D. Yoo, Y. Al-Hammadi, K. Taha, S. Muhaidat, Y.S. Jeong, U. Lee, M. Ismail, Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps, IEEE Trans. Ind. Inf. 11 (6) (2015) 1267–1276.

[4] S. Kang, Joint modeling of classification and regression for improving faulty wafer detection in semiconductor manufacturing, J. Intell. Manuf. (2018) 1–8.

[5] N.G. Shankar, Z.W. Zhong, Defect detection on semiconductor wafer surfaces, Microelectron. Eng. 77 (3-4) (2005) 337–346.

[6] C.H. Wang, S.J. Wang, W.D. Lee, Automatic identification of spatial defect patterns for semiconductor manufacturing, Int. J. Prod. Res. 44 (23) (2006) 5169–5185.

[7] T. Yuan, W. Kuo, Spatial defect pattern recognition on semiconductor wafers using model-based clustering and Bayesian inference, Eur. J. Oper. Res. 190 (1) (2008) 228–240.

[8] M.J. Wu, J.S.R. Jang, J.L. Chen, Wafer map failure pattern recognition and similarity ranking for large-scale data sets, IEEE Trans. Semicond. Manuf. 28 (1) (2015) 1–12.

[9] F.L. Chen, S.F. Liu, A neural-network approach to recognize defect spatial pattern in semiconductor fabrication, IEEE Trans. Semicond. Manuf. 13 (3) (2000) 366–373.

[10] J.Y. Hwang, W. Kuo, Model-based clustering for integrated circuit yield enhancement, Eur. J. Oper. Res. 178 (1) (2007) 143–153.

[11] C.S. Liao, T.J. Hsieh, Y.S. Huang, C.F. Chien, Similarity searching for defective wafer bin maps in semiconductor manufacturing, IEEE Trans. Autom. Sci. Eng. 11 (3) (2014) 953–960.

[12] H. Wu, J. Zhao, An intelligent vision-based approach for helmet identification for work safety, Comput. Ind. 100 (2018) 267–277.

[13] G. Choi, S.H. Kim, C. Ha, S.J. Bae, Multi-step ART1 algorithm for recognition of defect patterns on semiconductor wafers, Int. J. Prod. Res. 50 (12) (2012) 3274–3287.

[14] C.Y. Chang, C.H. Li, Y.C. Chang, M. Jeng, Wafer defect inspection by neural analysis of region features, J. Intell. Manuf. 22 (6) (2011) 953–964.

[15] S.P. Cunningham, S. MacKinnon, Statistical methods for visual defect metrology, IEEE Trans. Semicond. Manuf. 11 (1) (1998) 48–53.

[16] J. Chen, C.J. Hsu, C.C. Chen, A self-growing hidden markov tree for wafer map inspection, J. Process Control 19 (2) (2009) 261–271.

[17] D.M. Tsai, J.Y. Luo, Mean shift-based defect detection in multi-crystalline solar wafer surfaces, IEEE Trans. Ind. Inf. 7 (1) (2011) 41–49.

[18] C.H. Wang, Recognition of semiconductor defect patterns using spatial filtering and spectral clustering, Expert Syst. Appl. 34 (3) (2008) 1914–1923.

[19] C.H. Wang, Separation of composite defect patterns on wafer bin map using support vector clustering, Expert Syst. Appl. 36 (2-part-P1) (2009) 2554–2561.

[20] J. Kim, Y. Lee, H. Kim, Detection and clustering of mixed-type defect patterns in wafer bin maps, IISE Trans. 50 (2) (2018) 99–111.

[21] C.J. Huang, Clustered defect detection of high quality chips using self-supervised multilayer perceptron, Expert Syst. Appl. 33 (4) (2007) 996–1003.

[22] R. Baly, H. Hajj, Wafer classification using support vector machines, IEEE Trans. Semicond. Manuf. 25 (3) (2012) 373–383.

[23] L.C. Chao, L.I. Tong, Wafer defect pattern recognition by multi-class support vector machines by using a novel defect cluster index, Expert Syst. Appl. 36 (6) (2009) 10158–10167.

[24] B. Kim, Y.S. Jeong, S.H. Tong, I.K. Chang, A regularized singular value decomposition-based approach for failure pattern classification on fail bit map in a DRAM wafer, IEEE Trans. Semicond. Manuf. 28 (1) (2015) 41–49.

[25] P.L. Ooi, P.L. Ooi, K.S. Hong, C.K. Ye, S. Demidenko, C. Chan, Defect cluster recognition system for fabricated semiconductor wafers, Eng. Appl. Artif. Intell. 26 (3) (2013) 1029–1043.

[26] Y. Park, I.S. Kweon, Ambiguous surface defect image classification of AMOLED displays in smartphones, IEEE Trans. Ind. Inf. 12 (2) (2016) 597–607.

[27] Y. Wang, X. Wang, W. Liu, Unsupervised local deep feature for image recognition, Inf. Sci. 351 (2016) 67–75.

[28] S. Zhan, Q.Q. Tao, X.H. Li, Face detection using representation learning, Neurocomputing 187 (2016) 19–26.

[29] S.M. Siniscalchi, D. Yu, L. Deng, C.H. Lee, Exploiting deep neural networks for detection-based speech recognition, Neurocomputing 106 (2013) 148–157.

[30] D. Weimer, B. Scholz-Reiter, M. Shpitalni, Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection, CIRP Ann. Manuf. Technol. 65 (1) (2016) 417–420.

[31] T. Nakazawa, D.V. Kulkarni, Wafer map defect pattern classification and image retrieval using convolutional neural network, IEEE Trans. Semicond. Manuf. 31 (2) (2018) 309–314.

[32] K. Kyeong, H. Kim, Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks, IEEE Trans. Semicond. Manuf. (2018) In Press..

[33] K.B. Lee, S. Cheon, C.O. Kim, A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes, IEEE Trans. Semicond. Manuf. 30 (2) (2017) 135–142.

[34] H. Lee, Y. Kim, C.O. Kim, A deep learning model for robust wafer fault monitoring with sensor measurement noise, IEEE Trans. Semicond. Manuf. 30 (1) (2017) 23–31.

[35] J. Masci, U. Meier, D. Cire¸san, J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extrac- tion,”, Proceedings of the International Coriference on Articial Neural Networks (ICANN) (2011) 52–59.

[36] W. Luo, J. Li, J. Yang, W. Xu, J. Zhang, Convolutional sparse autoencoders for image classification, IEEE Trans. Neural Networks Learn. Syst. (2017) in press.

[37] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, Proceedings of the 26th Annual International Conference on Machine Learning (ICML), (2009) , pp. 609–616 June 2019.

[38] H. Lee, Y. Kim, C.O. Kim, A deep learning model for robust wafer fault monitoring with sensor measurement noise, IEEE Trans. Semicond. Manuf. 30 (1) (2017) 23–31.

[39] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, Proceedings of the International Conference on Artificial Neural Networks (ICANN) (2011) 52–59.

[40] J. Deng, W. Dong, R. Socher, L.J. Li, L.F. Fei, Imagenet: a large-scale hierarchical image database. Computer vision and pattern recognition, CVPR, 2009. IEEE Conference on, (2009) , pp. 248–255 June 2009.

[41] H.C. Shin, M.R. Orton, D.J. Collins, S.J. Doran, M.O. Leach, Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1930–1943.

[42] A. Ng, Sparse autoencoder, CS294A Lecture notes 72 (2011) 1–19.

[43] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (6583) (1996) 607–609.

[44] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors Available from: <arXiv:1207.0580>, (2012) .

[45] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[46] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, Proceedings of the 27th International Conference on Machine Learning (ICML-10), (2010) , pp. 807–814.

[47] X. Zhang, R. Wu, Fast Depth Image Denoising and Enhancement Using A Deep Convolutional Network. Acoustics, Speech and Signal Processing (ICASSP), (2016) , pp. 2499–2503.

[48] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.

[49] Y. Boureau, J. Ponce, Y. Lecun, A theoretical analysis of feature pooling in visual recognition, Proceedings of the 27th International Conference on Machine Learning (2010) 111–118.

[50] G. DeNicolao, E. Pasquinetti, G. Miraglia, F. Piccinini, Unsupervised spatial pattern classification of electrical failures in semiconductor manufacturing, Proc. Artificial Neural Networks Pattern Recognition Workshop, (2003) , pp. 125–131.

[51] M. Abadi, A. Agarwal, P. Barham, Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, in arXiv preprint arXiv, 2016, pp. 1603 04467v2.

[52] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems(NIPS), (2012) , pp. 1097–1105 2012.

[53] A.F.M. Agarap, A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data, Proceedings of the 2018 10th International Conference on Machine Learning and Computing (2018) 26–30.

[54] L. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. (2008) 2579–2605.

[55] A. Syaripudin, M.L. Khodra, A Comparison for Handling Imbalanced Datasets, Advanced Informatics: Concept, Theory & Application, IEEE, 2015.