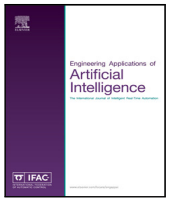
Engineering Applications of Artificial Intelligence 105 (2021) 104387

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Wafer map defect recognition based on deep transfer learning-based densely connected convolutional network and deep forest

Jianbo Yu *, Zongli Shen, Shijin Wang

School of Mechanical Engineering, Tongji University, Shanghai, 200084, China

ARTICLE INFO

Keywords:
Semiconductor manufacturing
Wafer map defect
Transfer learning
Convolution neural network
Deep forest

ABSTRACT

Due to the complexity and dynamics of the semiconductor manufacturing processes, wafer maps will present various defect patterns caused by various process faults. Identification of those defect patterns on wafer maps can help operators in finding out root-causes of abnormal processes, and then ensures that the manufacturing process is restored to the normal state as soon as possible. This paper proposes a wafer map defect recognition (WMDR) model based on integration of deep transfer learning and deep forest. Firstly, we transfer the network weight parameters of ImageNet to the convolutional neural network (CNN) (i.e., densely connected convolutional network (DenseNet)) and redesign the classification layer. This reduces the training time and then improves feature learning performance of DenseNet. Moreover, the transfer learning-based feature learning is able to solve class imbalance of wafer defect patterns. Finally, deep forest is utilized to identify the wafer defect pattern based on the abstract features from the wafer maps extracted by DenseNet. The experimental results on an industrial case show that the method can effectively improve WMDR performance and outperforms those well-known CNNs and other typical classifiers.

1. Introduction

Semiconductor manufacturing has become one of the most important industries in modern industry. Wafer manufacturing processes generally involve hundreds of integrated circuits (ICs). Due to the complexity and dynamics of semiconductor manufacturing process, wafer maps are prone to present various defect patterns caused by various process faults (Adly et al., 2015a; Kim et al., 2016; Yu and Lu, 2016; Wang and Chen, 2019; Liu and Chien, 2013). These typical defect patterns (e.g., ring, scratch, semicircle, repeat, cluster) on wafer maps are usually caused by specific equipment failures or process variations. For instance, edge ring defects generally arise from erroneous etching. Linear defects are caused by friction between the machine and the wafer surface, and central defects are usually produced by thin film deposition (Yu and Lu, 2016). Thus, quickly identifying the abnormal product and determining the defect causes in the production process can effectively reduce the rework rate and the scrap rate in the semiconductor manufacturing process, and then improve production efficiency and reduce costs (Wang and Chen, 2019; Liu and Chien, 2013).

Wafer map defect recognition (WMDR) is generally divided into two steps: defect detection and classification. The traditional defect detection method selects the specific regions as the image defect by comparing the image to be detected with the reference image (Shankar and Zhong, 2005). Wang and Chen (2019) applied weight masks to extract rotation-invariant features for improving classification performance of classifiers on the defect patterns of wafer maps. Liu and Chien (2013) integrated spatial statistics test, cellular neural network, adaptive resonance theory, and moment invariant to cluster different patterns of wafer maps effectively. However, those accurate reference images are not easy to obtain in real-world cases. The traditional defect classification method firstly extracts the features from wafer maps, and then uses them as inputs of the classifier to perform pattern recognition (Koo and Cho, 2010). A big challenge for these methods is whether the extracted features can effectively separate various patterns exhibiting on wafer maps.

With the rapid development of machine learning techniques, various typical recognizers have been widely used for wafer maps defect recognition and have achieved good results in recent years. The supervised classifiers consist of back-propagation network (BPN) (Hwang and Kuo, 2007), general regression neural network (GRNN) (Baly and Hajj, 2012), support vector machine (SVM) (Xie et al., 2014; Chao and Tong, 2009), randomized general regression network (Adly et al., 2015b), K-nearest-neighbor (KNN) (Kim et al., 2015), decision tree (Ooi et al., 2013), voting ensemble model (Saqlain et al., 2019) etc. Although these classic supervised recognizers have achieved some good results in wafer defect recognition, their performances dependent on effectiveness of those selected features manually. The high dimensions and noise of wafer images affect the performance of these classifiers significantly. At

* Corresponding author.
E-mail address: jbyu@tongji.edu.cn (J. Yu).

https://doi.org/10.1016/j.engappai.2021.104387
Received 26 April 2019; Received in revised form 23 March 2020; Accepted 10 July 2021
Available online 17 August 2021

the same time, selecting effective features manually based on specific wafer defect patterns is not an easy task in practice. Thus, it is very interesting for feature learning directly from high-dimensional images to automatically capture effective pattern features and then to improve recognition performance of these classifiers.

In recent years, feature learning techniques based on deep learning have attracted wide attentions from researchers in the field of machine learning (LeCun et al., 2015). Deep learning, also known as deep neural network (DNN), is a new feature learning method with multiple hidden layers. It uses a hierarchical structure with multiple neural layers and learns feature information of the input data layer by layer. This deep structure is capable of learning the high-level abstract features of complex raw data, which makes it easier to extract useful features when constructing classifiers or recognizers (Krizhevsky et al., 2012). Considering the ability of DNNs to handle large-scale data, various DNNs are widely applied in actual industrial productions, e.g., deep belief networks (DBN), auto-encoder (AE), denoising auto-encoder (DAE), stacked denoising auto-encoder (SDAE) have achieved good results in machine health monitoring and fault diagnosis (Thirukovalluru et al., 2016; Jiang et al., 2017; Dong, 2019). However, the above methods can only deal with one-dimensional signals. When two-dimensional data (e.g., wafer maps) are available, these recognizers do not work well. Convolutional neural networks (CNNs) have become the dominant machine learning approach for image visual recognition. Although CNNs were proposed more than 20 years ago (Szegedy et al., 2015), improvements in computer hardware has enabled wide applications of deep CNNs (Szegedy et al., 2015; He et al., 2015; Huang et al., 2016) only recently. Densely connected convolutional networks (DenseNet) (Huang et al., 2016), by densely connecting the feature information of all layers, mitigates the problem of gradient disappearance, enhances feature delivery, and makes more effective use of image features. In general, the extracted feature set extracted by CNN persists high dimension and much noise. Thus, it is necessary to further reduce feature dimension and redundant information, and then extract important representations from the learned features by CNNs.

In the field of wafer defect recognition, some researchers have employed CNNs to use wafer images directly as inputs for classification of wafer map defects (Weimer et al., 2016). CNN is also applied in mixed defect classification (Yu et al., 2019; Chen et al., 2020; Nakazawa and Kulkarni, 2018), automatic feature extraction (Kyeong and Kim, 2018), and wafer image retrieval (Lee et al., 2017). Yu et al. (2019) proposed a CNN-based feature learning method, and then performed principal component analysis-based dimensionality reduction and similarity ranking to infer the root causes of wafer map samples. Chen et al. (2020) extracted high-level features from wafer maps by using CNN and then fed them to SVM for WMDR. However, all of the above methods need to redesign and train a new deep CNN when encountering new tasks. The whole process requires a lot of time and computing resources. Transfer learning (Pan and Yang, 2010a,b) can be considered as a new strategy for machine learning at the minimum human supervision cost. The network model pre-trained in large-scale datasets can be used as a feature extractor in other tasks through model migration. Many experiments on different datasets prove that the features extracted by the pre-training network have better discriminability (Donahue et al., 2013). In addition, due to the small number of label images, the manual acquisition cost is high, and the transfer learning method is excellent in the recognition of small data. Transfer learning will provide a new way for solving some difficult problems (e.g., feature learning, high time cost of model construction) on WMDR. It is a very interesting issue to apply transfer learning to WMDR to improve the industrial applicability of those DNNs.

As a new deep learning technique, deep forest (Zhou and Feng, 2017; Hu et al., 2018), a decision tree model based on random forest algorithm, can effectively deal with high-dimensional features and exhibits good performance on pattern recognition by multi-dimensional scanning and cascade processing. Various experiment results illustrate that hyperparameter of deep forest is much less than DNNs, and its default hyperparameter setting is also suitable for dealing with different tasks in many applications (Mubarak et al., 2018). Deep forests have achieved good performance in housing grade classification, power forecasting, and forest fire identification (Hu et al., 2018; Mubarak et al., 2018; Zhao et al., 2018; Yang et al., 2018).

In summary, these are still some problems existing on these WMDR models: (1) The feature generation and selection is implemented generally on wafer maps before construction of the classifiers for WMDR; (2) Although many DNNs have been proposed to learn features from wafer maps, their training generally requires to redesign the whole deep network structure, and the parameter initialization of the deep network affects the convergence significantly in the training procedure; (3) The features extracted by DNNs (e.g., CNNs) are generally high dimensional and contain many redundant information, which brings some challenges for those regular classifiers on WMDR; (4) Most of the wafer map datasets have a class imbalance problem that affects the classification performance of these DNNs.

In order to solve these problems, this study proposes a novel model based on transfer learning-based DenseNet and GCForest (DenseNet-GCF) for WMDR. Transfer learning effectively reduces the training time cost of DenseNet, and then helps it effectively extract deep features of wafer maps. Based on the deep features extracted by DenseNet, a deep forest model is developed as a recognizer of defect pattern on wafer maps. The main contributions of this study are following: (1) Transfer learning setups the network structure and parameters of DenseNet effectively and the redesign of the classification layer improves converge performance, and then reduces the modeling difficulties and training time cost; (2) Transfer learning-based DenseNet is able to extract representative features from high-dimensional wafer maps directly, and then reduce the dependence on large amount data and finally improves performance of the classifier significantly; (3) GCForest is developed as the classification layer of DenseNet to reduce feature dimension and further extract effective features as inputs of the classifier for WMDR. Finally, the validity of DenseNet-GCF was verified in the industrial dataset (WM-811K). The experimental results show that the deep transfer learning model is capable of extracting effective features from wafer maps and improves the recognition performance of the classifier on WMDR.

The rest parts of the paper is organized as follows: Section 2 introduces CNN. The transfer learning-based DenseNet is proposed for WMDR in Section 3. Section 4 presents GCForest as a classification layer of DenseNet. Section presents the application procedure of the proposed model for WMDR. Section 6 presents the experiment and result analysis. The conclusion is given in Section 7.

2. Convolutional neural network

CNN is a feedforward neural network that includes convolution operations and has a deep structure. It has achieved great successes in the field of image and computer vision. As shown in Fig. 1, a typical CNN consists of an input layer, hidden layers, and an output layer. The hidden layers include convolution layers, pooling layers, and full-connection layers. CNN extracts features through convolutional layers and reduces the number of network parameters by sharing convolution weights and pooling operations. Finally, those tasks such as classification or regression can be performed by using traditional classifiers.

Convolution layer: The convolutional layer is to extract features from the input images. The convolutional layers contain a number of convolution kernels that convolute input images, generally called a feature map. Within each filter, those neurons are directly connected to the input data points and multiply them by the weights. The input will be transformed in the convolution kernel as follows:

$$Z^{l+1}(i,j) = [Z^l \otimes w^l](i,j) + b \tag{1}$$

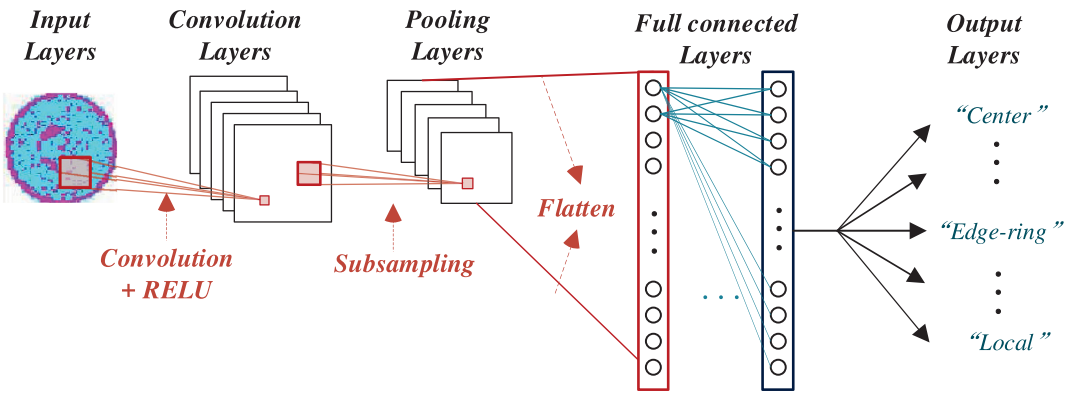


Fig. 1. Convolutional neural network.

where Z^l and Z^{l+1} denote the input and output of the $l+1$ convolutional layer, respectively, also known as feature maps, and b is the l-layer bias. The output of the convolution filter for the next layer is computed based on the activation function, i.e., rectified linear unit (ReLU) function:

$$f(x) = \begin{cases} 0 & , x < 0 \\ x & , x \geq 0 \end{cases} \tag{2}$$

where x represents the input that is the output result of each layer in the network.

Pooling layer: A pooling layer usually follows a convolution layer to generate a lower resolution representation through sub-sampling.

DenseNet generally consists of multiple dense blocks and transition 10 layers (Huang et al., 2016). The pooling layer contains a preset pooling function that is to replace the result of a single point in the feature map with the feature graph statistics of its neighboring area, thereby compressing the data and reducing the amount of parameters.

Fully connected layer: The neurons in a fully connected layer have the connections to all activations in the previous layer. The feature maps will lose the 3-dimensional structure in the fully connected layer, expanded into a 1-dimensional vector, and then be passed to the next layer through the excitation function.

Output layer: It specifies how training penalizes the deviation between the predicted and true class labels. Various loss functions have been used there for different tasks, the model-based transfer learning can refer to Yu et al. (2021).

3. Feature learning based on transfer learning and DenseNet

This section develops a deep CNN model, i.e., transfer learning-based DenseNet for feature learning from wafer maps. By transferring weight parameters of pre-trained DNN on a large dataset to the initial parameters of the DenseNet model and redesigning the fully connected classification layer, DenseNet is able to extract high-dimensional abstract features from wafer maps. Although there is a serious class imbalance in the industrial wafer map dataset, the transfer learning method can effectively extract high-dimensional features from the wafer maps with a small class size.

3.1. DenseNet

DenseNet (Huang et al., 2016) was proposed on the basis of GoogLeNet (Szegedy et al., 2015), ResNet (He et al., 2015) and HighwayNet (Srivastava et al., 2015). It exhibits efficient capability of feature extraction because of its their compact internal representations and reduced feature redundancy. In general, the recognition performance improvement of DNN is accompanied by the increase of the depth of the network. However, a problem emerges: as information about the input or gradient passes through many layers, it can vanish and "wash out" by the time it reaches the end (or beginning) of the network. DenseNet can alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. DenseNet introduces direct connections from any layer to all subsequent layers. Thus, DenseNet can comprehensively utilize the features of shallow layers, enhance the transmission and utilization of features, alleviate the problem of gradient disappearance, and obtain good generalization performance.

Fig. 2 shows a significant structure of DenseNet, i.e., the dense block. It can be seen that the h4 layer not only directly uses the original information x0 as an input, but also uses the information provided by the h1, h2 and h3 layer for x0 as the input. A very simple expression describes the transformation of each layer in the dense block as follows:

$$x_k = h_k([x_0, \ldots, x_{l-1}]) \tag{3}$$

where h_k represents the output function of the kth layer. In the back-propagation process, the gradient information of X_k is not only from the previous layer, but also contains the derivative of the loss function calculated on the output of all layers.

3.2. Transfer learning-based DenseNet169

DenseNet generally consists of multiple dense blocks and transition layers. Based on a densely connected neural network (DenseNet169) with a network structure depth of 169, the first four dense block layers in DenseNet169 are transferred from ImageNet. Since there are 9 classes of wafer maps in this study, a fully connected layer with 9 output neurons is used instead of the 1000-dimensional fully connected layer in the classification layer.

Fig. 3 shows the improved network structure of DenseNet169 in the wafer identification task. It can be seen that the red area is the transfer module, the dark blue area is the redesigned classification layer, and the parameter k in the structure table represents the number of each convolution layer filter, and the 1×1 and 3×3 convolution operations are alternately used in each dense block. The Softmax function is used in the classification layer to convert the neuron output to probability:

$$Softmax([x_1, \ldots, x_n]^T) = [\frac{\exp(x_1)}{\sum_{i=1}^{n} \exp(x_i)}, \ldots, \frac{\exp(x_n)}{\sum_{i=1}^{n} \exp(x_i)}]^T \tag{4}$$

where $x_1, \ldots, x_n$ correspond to the output value of the fully connected layer. The corresponding value is mapped into the interval between 0 and 1 by the Softmax function, and the predicted value is converted into the probability distribution of the possible events. Since there are 9 wafer map patterns to be considered, the 9 output neurons of the fully connected layer are constructed for the WMDR task.

DenseNet169 takes the cross entropy loss function as the objective function to measure the difference between the recognition result and the true result:

$$H(p, q) = -\sum_{x} p(x) \log q(x) \tag{5}$$

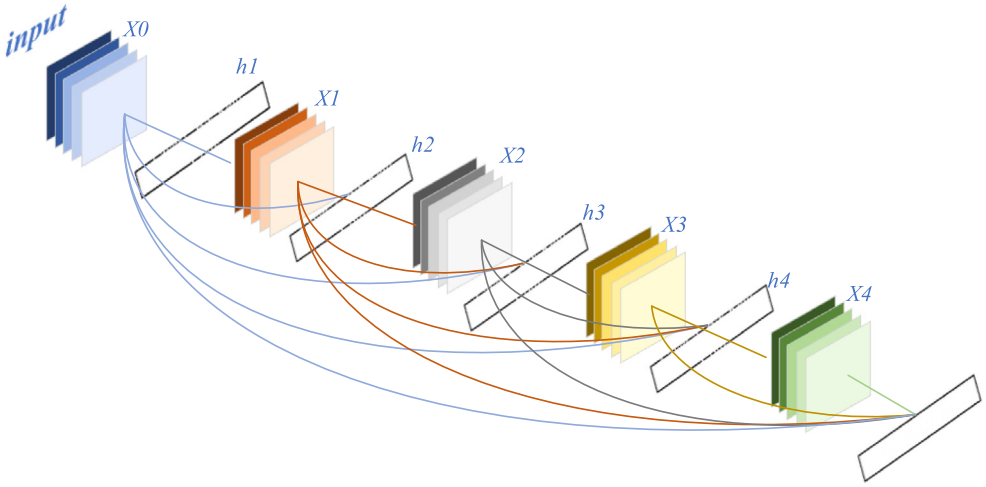
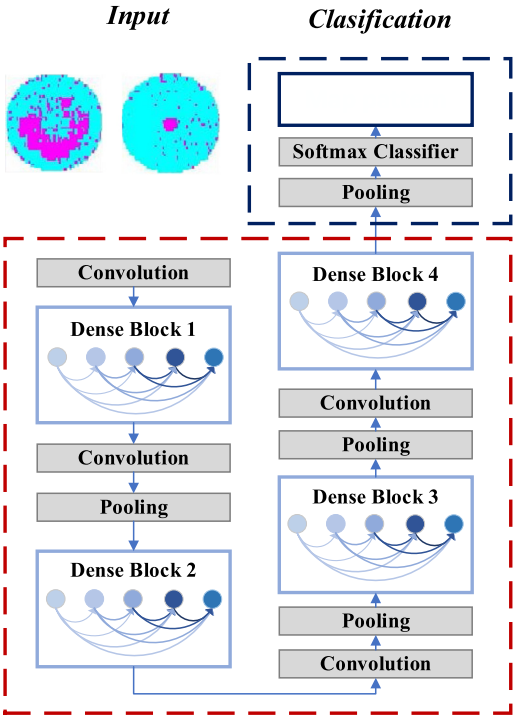


Fig. 2. Dense block.



Layers	DenseNet169(k=32)
Convolution	7x7 conv,stride 2
Pooling	3x3 conv,stride 2
Dense Block 1	[1x1 conv] *6 [3x3 conv] *6
Transition Layer	1x1 conv 2x2 average pool，stride 2
Dense Block 2	[1x1 conv] *12 [3x3 conv] *12
Transition Layers	1x1 conv 2x2 average pool，stride 2
Dense Block 3	[1x1 conv] *32 [3x3 conv] *32
Transition Layers	1x1 conv 2x2 average pool，stride 2
Dense Block 4	[1x1 conv] *32 [3x3 conv] *32
Classification Layers	7x7 global average pool 9D fully-connected,softmax

Fig. 3. Transfer learning-based DenseNet. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

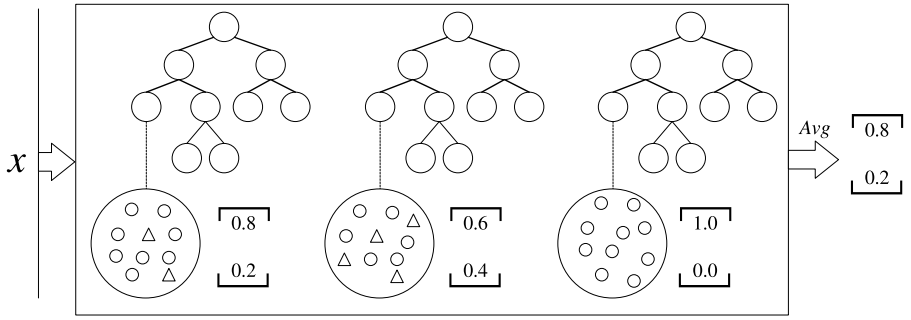


Fig. 4. Class vector generation in GCForest.

where p and q represent the true and predicted distribution, respectively, corresponding to the distribution of the real wafer map and the prediction distribution of DenseNet169.

DenseNet169 uses stochastic gradient descent (SGD) to train the network (Huang et al., 2016). We extract one sample from all training samples to update weight parameters each time.

$$\theta_i = \theta_i - \alpha(h_\theta(x_0^{(j)}, \ldots, x_n^{(j)}) - y_j)x_i^{(j)} \tag{6}$$

where α denotes the learning rate, θ represents the weight parameters to be updated and j is the chosen sample index.

The DenseNet network freezes the first 3 DenseBlocks in the training process and locally fine-tunes the subsequent network. After the model converges, the outputs from the global pooling operation in the final classification layer are used as the deep features of wafer maps.

4. GCForest

The features extracted by DenseNet169 is still high dimensional and consist of much noise. Deep forest can effectively deal with high-dimensional feature information through multi-dimensional scanning and cascade processing method, and has shown good performance in pattern recognition.

Multi-grained cascade forest (GCForest) is a new decision tree integration method (Zhou and Feng, 2017; Liu et al., 2008; Guo, 2009). This method uses a cascade structure based on random forest, which enhances the representation learning ability of decision tree. When the inputs are high dimensional data, the feature learning ability of the method can be further enhanced by multi-dimensional scanning. GCForest can adaptively determine the number of cascaded layers. Users can control the training cost based on available computing resources. Another advantage of GCForest is that it setups fewer hyperparameters to improve its applicability.

A GCForest model consists of two parts. The first part is a multi-granularity scanning structure. It scans the input data through a sliding window and generates feature vectors from a set of sliding windows with multiple sizes. The second part is the cascade forest structure to generate the corresponding class distribution according to each input instance. Representation learning in DNNs mostly depends on layer-by-layer processing of the data. GCForest introduces a cascade structure, where each level of cascade receives feature information processed by the preceding level, and outputs its features to the next level. Each level in cascade forests is an ensemble of decision tree forests. As shown in Fig. 4, each level of GCForest generates class distribution by calculating the percentage of different class instances at the leaf nodes of the decision tree, and then outputs the average of the predictions of all trees in the same forest to the next level layer by layer. Fig. 5 shows the structure of the deep forest. The multi-granularity scan stage generates multi-channel feature vectors with different sizes from the input data. Then, the cascade forest stage predicts the class distribution of the generated features through different cascade channels layer by layer. This procedure will be repeated till convergence of validation performance. Each layer in the cascade forests consists of two random forests and two completely random forests based on the consideration of integrated prediction diversity. Each random forest contains 500 decision trees. The completely random forest is composed of 1000 decision trees. Each tree randomly selects a feature as the split node of the split tree, and then grows until each leaf node is subdivided into only one category or no more than 10 samples. A common random forest consists of 1000 decision trees. Each tree is randomly selected $\sqrt{d}$ (d is the input feature dimension) candidate features, and then the split nodes are filtered by the Gini index. Thus, the main difference between the two forests lies in the candidate feature space. The completely random forest is randomly selected in the complete feature space to split, while the ordinary random forest selects the split node through the Gini index in a random feature subspace. The Gini index can be calculated as follows:

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2 \tag{7}$$

where p_k denotes the probability that the sample belongs to the k class. It is assumed that the possible values of the samples are {1, 2, …, K} in the classification task. The Gini index calculation can refer to Zhou and Feng (2017), Liu et al. (2008), Guo (2009)

5. Application procedure of DenseNet-GCF

This study proposes a WMDR model, i.e., DenseNet-GCF that integrates DenseNet and GCForest based on transfer learning to address the class imbalance and image high-dimension problem in the wafer map dataset. As shown in Fig. 6 the application of process of DenseNet-GCF for WMDR is divided into two parts: offline modeling and online identification.

Offline modeling:

Step 1: Collect the wafer images offline to generate a dataset;

Step 2: Generate a training and test dataset from the collected dataset for the proposed model;

Step 3: The network structure of DenseNet169 is setup;

Step 4: Transfer weight parameters from ImageNet to DenseNet169 to setup the weights of DenseNet169;

Step 5: Redesign the full connection and pooling layer of DenseNet169 and perform finetuning training based on the training dataset.

Step 6: Extract the global average pooling layer as the output features of the wafer image features;

Step 7: Train the deep forest model as the classifier based on the extracted features by DenseNet169

Online identification:

Step 1: Collect a wafer map from the manufacturing process online;

Step 2: Input this image to the trained DenseNet169 model for feature extraction;

Step 3: Extract the features of the wafer defect image as inputs of GCForest;

Step 4: Input the extracted features into the trained GCForest model for WMDR;

Step 5: Output the final classification result of GCForest, and implement the corresponding measurements to adjust the out-of-control process.

6. Experiment and result analysis

In this study, the WM-811K wafer map library (Wu et al., 2015) is used to test the effectiveness of DenseNet-GCF to recognize defects. The images in the sample library are all from the real semiconductor manufacturing system. The cyan, magenta and white pixels in the sample library represent the normal, defective and empty elements of each wafer map, respectively. The WM-811K dataset consists of normal and eight defect patterns. Fig. 7 shows the nine patterns, i.e., Center, Donut, Edge-local, Edge-ring, Local, Near-full, Random, Scratch and None-pattern. Fig. 8 shows the number of each pattern in the WM-811K dataset. It can be seen that these is a significant class imbalance, which could bring a great challenge for recognizers to correctly recognize these defect patterns. The size of each wafer image is preprocessed to $224 * 224$ pixels. The 7112 wafer images were randomly selected as the training dataset, and the remaining 2000 maps were used as the testing dataset.

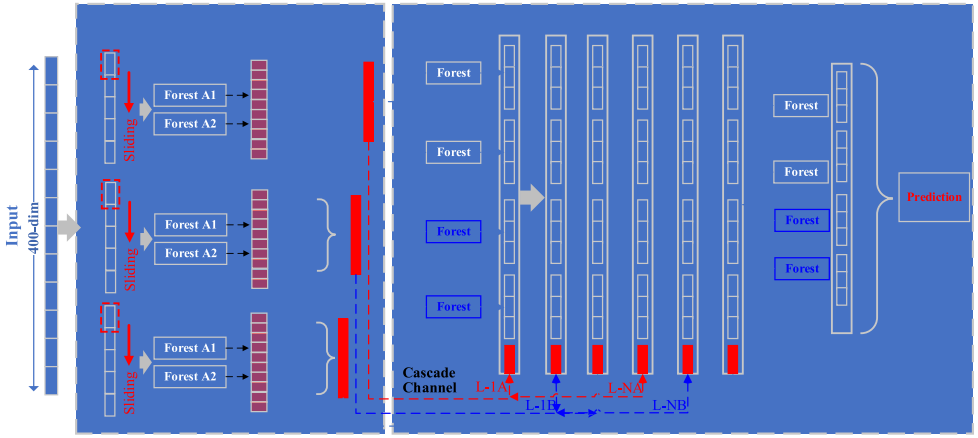


Fig. 5. Deep forest (taking 400-dimensional input as an example).

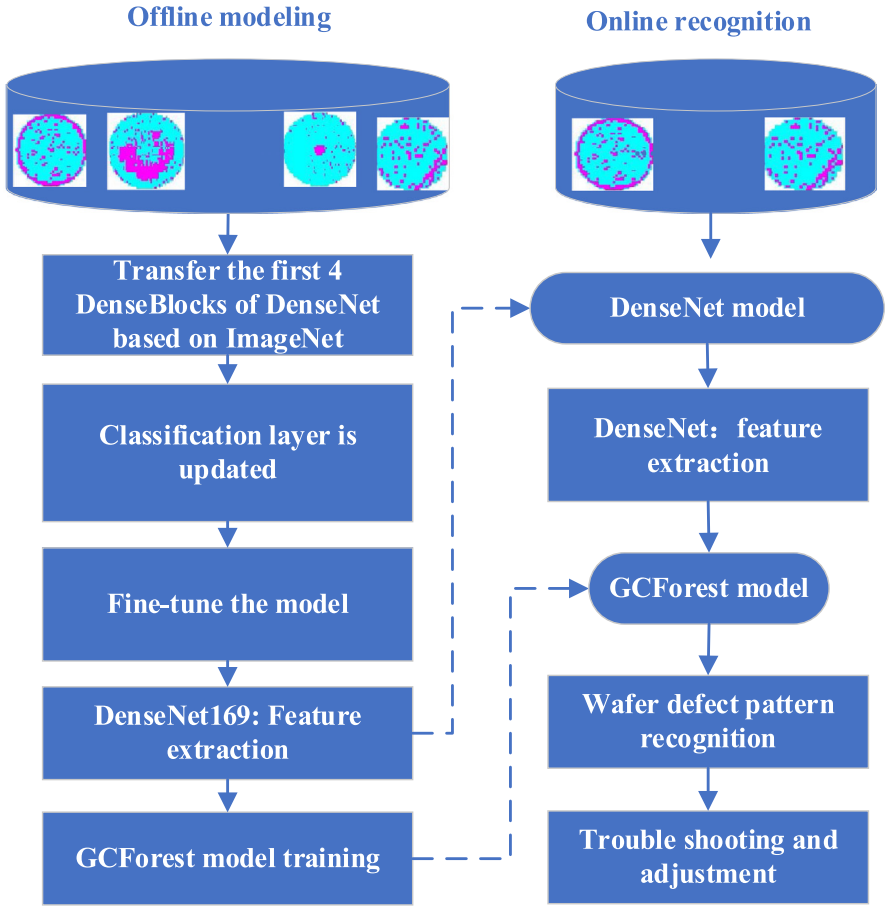


Fig. 6. Application procedure of DenseNet-GCF.

6.1. *Transfer learning-based DenseNet for feature learning*

The transfer learning-based DenseNet169 is firstly used to extract features from wafer maps. After fine-tuning the final fully connected layer and output layer, the loss function value on the training dataset is shown in Fig. 9(a). It can be seen from Fig. 9(a) that the transfer learning-based DenseNet169 achieves good convergence when the batch is 3000. However, DenseNet169 without using transfer learning is convergent when the batch is 5000. It is clear that the transfer learning-based DenseNet169 obtains better training result and faster convergence speed than that of DenseNet169 without using transfer learning. Fig. 9(b) presents the recognition rate changes of DenseNet169 on the training and testing dataset. Although transfer learning-based DenseNet169 achieved lower loss function and faster convergence, Fig. 9(b) exhibits that the final recognition accuracy of transfer learning-based DenseNet169 on the testing dataset is less than 85%. Thus, an effective classifier is very significant to classify the abstract features with high dimension extracted by DenseNet169.

Fig. 10 is an image feature output of the second, third and fourth convolution modules of DenseNet. As the number of the layers increases, and the extracted features by these convolutional layers are changed from wafer shape to abstract shape. From the output of the fourth convolution layer, the shapes of the wafer maps can no

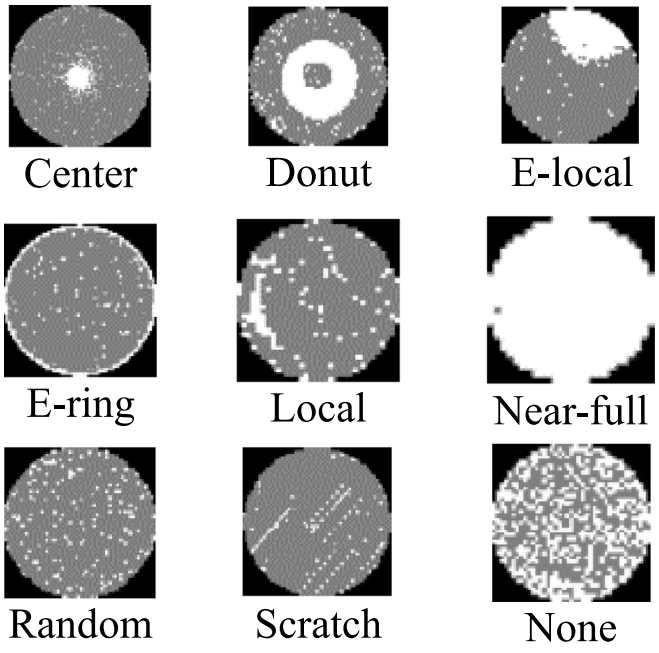


Fig. 7. Normal wafer pattern and 8 defect patterns.

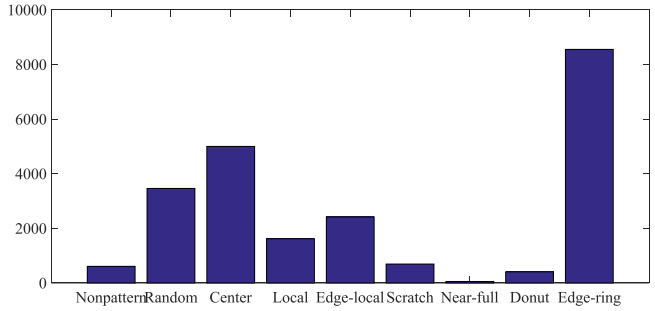


Fig. 8. Wafer map distribution of various patterns in the WM-811K dataset.

longer be distinguished, which indicates that the output features of the convolutional layer are more abstract than before.

In this study, the original image data and the output features of the pooling layer of DenseNet are further used for visualization analysis by using t-SNE (Van der Maaten and Hinton, 2008). Fig. 11 presents the two-dimension vectors extracted by t-SNE. As shown in Fig. 11(a), the features extracted by DenseNet can effectively separate the nine patterns. In contrast, Fig. 11(b) shows that the features from the raw wafer maps randomly spread in the dimension mapping with a large overlap under the different patterns. This feature visualization result demonstrates that the features of wafer maps extracted by the transfer learning-based DenseNet169 exhibit good class discriminant.

6.2. Recognition performance analysis

Table 1 presents the recognition rate of DenseNet-GCF in a combined confusion matrix. DenseNet-GCF achieves a correct recognition rate of 96.2% on the testing dataset. Except for Random, DenseNet-GCF shows very good performance on all other types of patterns. Although the wafer dataset has a significant class imbalance problem (as shown in Fig. 7), DenseNet-GCF effectively learns features from small-sample wafer maps (e.g., Near-full and Donut) and then performs effective classification based on transfer learning-based DenseNet and deep forest. As shown in Table 1, 5.88% and 0.75% of Random are misidentified as Near_full and Local, respectively. Fig. 12 presents four

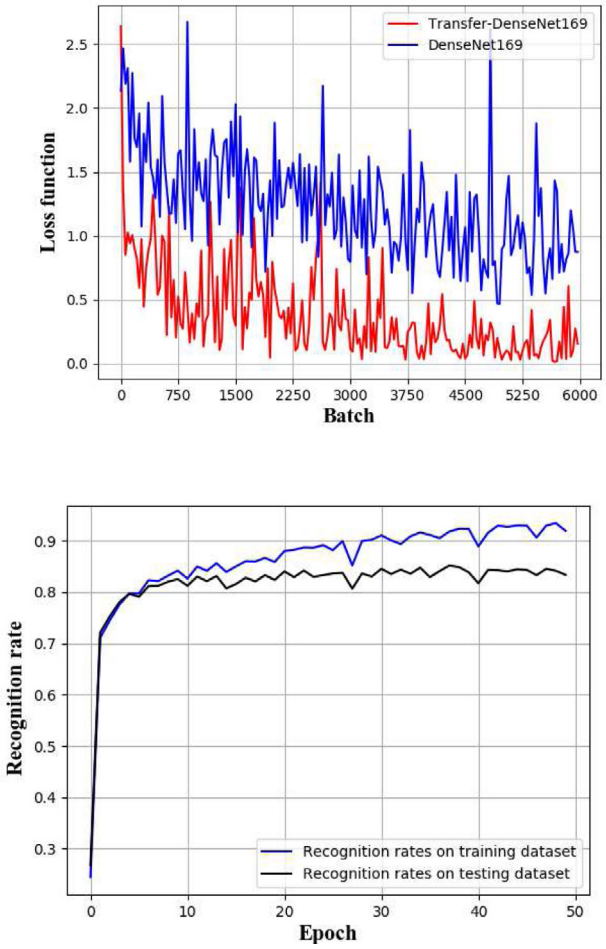


Fig. 9. Training procedure, (a) DenseNet169 and transfer learning-based DenseNet169, (b) Recognition rate (%) of transfer learning-based DenseNet169.

misidentified wafer maps with Random. The left two wafer maps have the features of both Random and Near_full defects, while the right two have Local and Edge_local defect features, respectively. As a result, misclassification of the model will occur due to the pattern similarity of these wafer maps. Overall, DenseNet-GCF exhibits good recognition performance in WMDR task.

6.3. Parameter sensitivity analysis

The main hyperparameters of GCForest consist of the number of forests, the number of decision trees per forest, the size of sliding windows, the cascade, the number of forests per layer and the decision tree stop growing rules. This study analyzes sensitivity of random forest generation rule parameters. It can be seen from Table 2 that these parameters have little affections on the testing results and GCForest is not sensitive to the parameter changes. Thus, deep forest makes the training of DenseNet-GCF easily and improves industrial applicability.

6.4. Performance comparison

In order to verify the effectiveness of DenseNet-GCF, several typical classifiers (i.e., SVM, RF, KNN and C4.5) and CNNs (GoogleNet (Szegedy et al., 2015), ResNet (He et al., 2015), and DenseNet (Huang et al., 2016)) are considered for comparison purpose. The maximum depth of decision tree C4.5 is set to 25 and the number of nodes is 100. The two SVMs (i.e., SVML and SVMG) use a linear kernel function and a Gaussian kernel function, respectively, and the penalty factor is set to

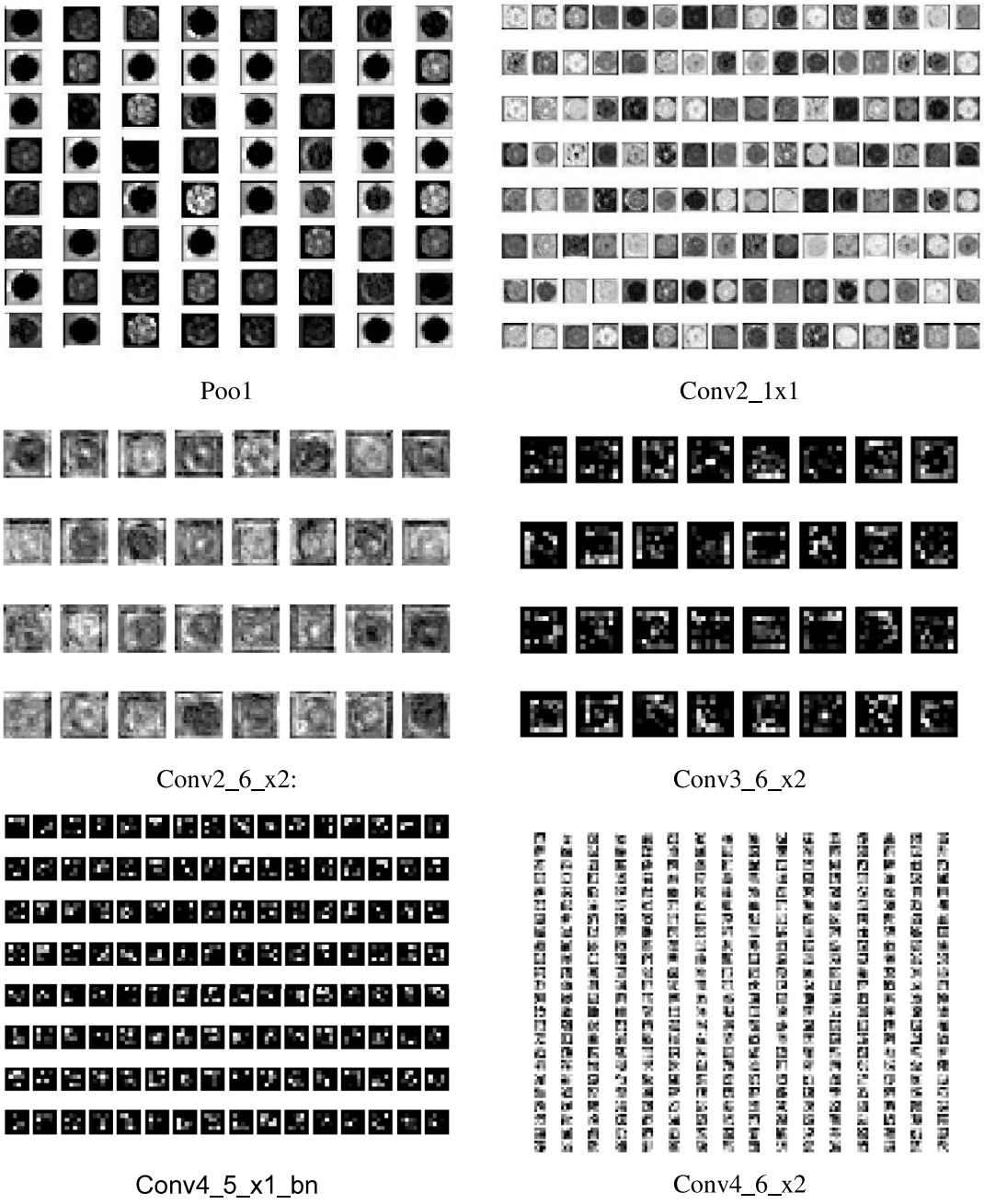


Fig. 10. Outputs of multi-level layers of DenseNet on wafer maps.

Table 1
Confusion matrix of DenseNet-GCF for WMDR (%).

	None	Center	Donut	Edge-local	Edge-ring	Local	Near-full	Random	Scratch
None	99.59	0.00	0.00	0.28	0.00	0.00	0.00	0.00	0.14
Center	0.83	97.52	0.00	0.83	0.00	0.00	0.00	0.00	0.83
Donut	0.00	0.00	94.74	0.00	0.00	5.26	0.00	0.00	0.00
Edge-local	0.86	0.29	0.00	96.83	0.29	1.44	0.00	0.29	0.00
Edge-ring	0.00	0.00	0.00	4.91	95.09	0.00	0.00	0.00	0.00
Local	0.75	0.38	0.00	1.51	0.00	96.23	0.00	0.75	0.38
Near-full	0.00	0.00	0.00	0.00	0.00	0.00	94.12	5.88	0.00
Random	0.00	2.56	0.00	0.00	2.56	5.13	2.56	87.18	0.00
Scratch	1.25	0.00	0.00	1.25	0.00	3.75	0.00	0.00	93.75

C = 1.0. The random forest has a maximum depth of 50 and contains 800 trees. The K value in KNN is set to 5.

In order to illustrate the computation complexity of these CNN models, the training time cost is further tested in this section. These testings will be performed on a laptop: CPU: inter core i7 9700K;

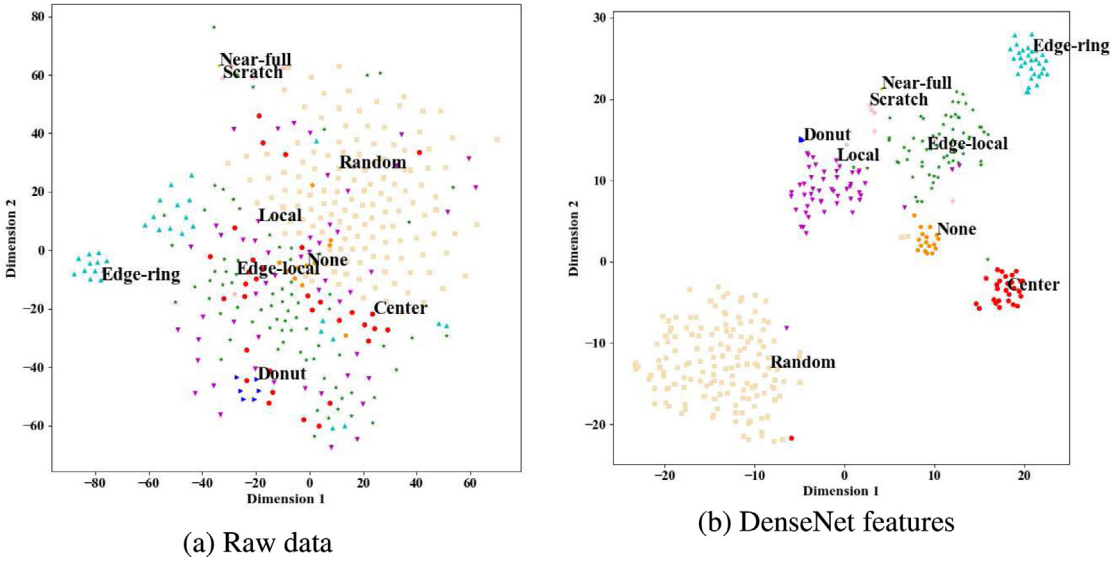


Fig. 11. Feature visualization via t-SNE for features learned by DenseNet169.

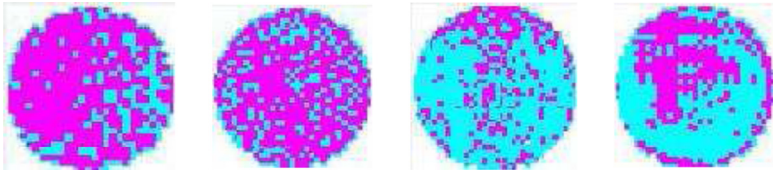


Fig. 12. Misclassification of the wafer maps with the Random pattern.

Table 2
Recognition accuracy of GCForest with different parameters (%).

Sliding window	ACC	Min samples	ACC	Tolerance	ACC	Num of trees	ACC
400	97.31	3	96.4	0.0	96.2	200	96.9
700	96.7	5	96.8	0.1	96.7	400	95.8
1000	96.2	7	95.8	0.2	96.2	600	97.0
1300	96.4	9	96.6	0.3	95.8	800	96.7
1664	96.2	11	96.7	0.4	95.4	1000	96.0

Table 3
Computation complexity of DenseNet, RestNet and GoogleNet.

CNN	Parameters	Flops
DenseNet169	$1.41 * 10^7$	$6.3 * 10^9$
ResNet50	$2.56 * 10^7$	$8.3 * 10^9$
GoogleNet	$4.55 * 10^6$	$2.28 * 10^9$

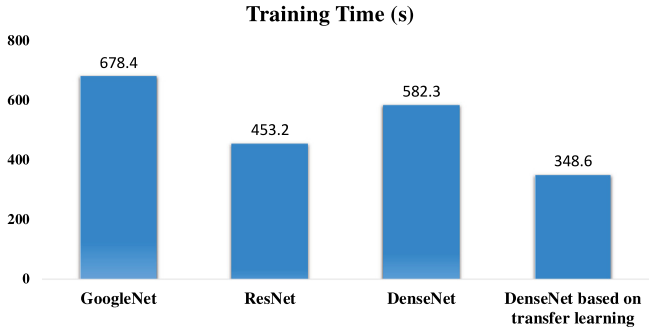


Fig. 13. Training time of GoogleNet, ResNet, DenseNet and DenseNet based on transfer learning.

GPU: NVIDIA GeForce GTX 1080; Memory: 16G. Fig. 13 exhibits the training time of three CNNs and DenseNet based on a transfer learning when they reach convergence. Obviously, DenseNet based on transfer learning can effectively reduce the training time, because it does not perform the complex hyperparameter tuning and network structure redesign. Finally, the computation time of the DenseNet-GCF model for an input wafer map in online WMDR is 3.5 ms. This real time cost of DenseNet-GCF is small and thus it can be applied in WMDR online.

The parameters and floating point operations (FLOPs representing the calculation scale of the algorithm) are generally utilized to verify the complexities of DNNs (He et al., 2015; Huang et al., 2016). Table 3 shows that although the parameter scale and Flops pf DenseNet are not dominant, transfer learning technique can effectively reduce training time cost of DenseNet.

Table 4 presents performance comparison between DenseNet-GCF and the three classic CNNs (i.e., GoogleNet, ResNet and DenseNet). It can be seen from Table 4 that DenseNet-GCF shows the obvious better result than that of these famous CNNs. At the same time, the training time of this method is far less than these CNNs. The ACU is the prediction accuracy that defines how many of the positive samples are true positive samples. The REC is the recall rate that defines how many positive examples are predicted correctly. F1($F1 = \frac{2*ACU*REC}{ACU+REC}$) is the harmonic mean of the two indications to represent the stability of the model recognition performance. As shown in Table 4, DenseNet-GCF shows the best performance on the seven wafer patterns, i.e., Center, Donut, Edge_ring, Edge_local, Local, None and Scratch. The F1-score of DenseNet-GCF on the Near_full and Random reaches 94.1% and 88.3%, respectively, which are close to the best results of other methods. In addition, DenseNet-GCF obtains recognition accuracy rates of 97.3% and 94.9% for Donut and Scratch with small samples, respectively. Thus, the comparison results illustrate that DenseNet-GCF based on transfer learning is capable of identifying small sample data and DenseNet has better feature learning performance than that of other CNNs.

Table 4
Performance comparison among DenseNet-GCF, GoogleNet, ResNet and DenseNet (%).

	DenseNet-GCF			GoogleNet			ResNet			DenseNet		
	ACU	REC	F1	ACU	REC	F1	ACU	REC	F1	ACU	REC	F1
Center	97.5	97.5	**97.5**	72.2	34.5	46.7	85.0	77.1	80.9	76.4	85.5	80.7
Donut	100	94.7	**97.3**	59.1	43.3	50.0	47.2	81.0	59.6	95.2	64.5	76.9
Edge_local	95.5	96.8	**96.1**	63.9	79.8	71.0	87.6	81.4	84.4	79.7	91.2	85.0
Edge_ring	98.7	95.1	**96.9**	93.4	87.6	90.4	94.3	86.6	90.3	96.1	81.5	88.2
Local	95.9	96.2	**96.0**	57.3	48.8	52.7	81.7	68.6	74.6	82.1	66.8	73.6
Near_full	94.1	94.1	94.1	93.3	93.3	93.3	75.0	75.0	75.0	91.7	100	**95.7**
None	99.0	99.6	**99.3**	90.6	92.9	91.7	89.9	99.3	94.4	95.2	99.3	97.2
Random	89.5	87.2	88.3	85.4	92.1	**88.6**	88.2	73.2	80.0	86.4	65.5	74.5
Scratch	96.2	93.8	**94.9**	81.7	81.7	81.7	59.5	71.0	64.4	81.5	72.6	76.8
Average	97.4	97.4	**97.4**	78.3	78.3	77.6	86.4	86.1	85.9	88.1	88.0	87.7

Table 5
Five-cross validation (%) of DenseNet-GCF and other recognizers.

DenseNet-GCF	GoogleNet	ResNet	GCForest	RF	SVML	SVMG	KNN	C4.5
96.8	74.3	86.5	**73.7**	68.9	72.5	40.2	30.1	62.4

Table 6
Comparison of five-fold cross recognition rate of various recognizers based on features extracted by DenseNet (%).

DenseNet-GCF	BPN	RF	SVML	SVMG	KNN	C4.5
96.8	85.12	95.1	95.6	95.5	92.6	87.22

The average recognition rate of these recognizers based on the five-fold cross-validation is shown in Table 5. This testing is to illustrate the feature learning and recognition performance of DenseNet-GCF based on a comparison with the typical CNNs (i.e., GoogleNet and ResNet), GCForest and other typical recognizers (i.e., RF, SVM, KNN and C4.5). It can be seen that DenseNet-GCF outperforms all other recognizers significantly. Meanwhile, these typical CNNs (e.g., GoogleNet, ResNet) generally perform better than traditional recognizers (e.g., SVM). This indicates that these CNNs are capable of learning effective features from wafer maps directly. However, these regular recognizers (e.g., SVM, RF) cannot show good recognition performance because the high-dimensional vectors transformed from the image are used as their inputs. Meanwhile, this further demonstrates that the integration of the transfer learning-based DenseNet and GCForest improves feature learning and recognition performance on WMDR.

In order to illustrate the feature learning of transfer learning-based DenseNet from wafer maps, the extracted features by DenseNet are used as input of other models. Table 6 shows recognition rates of these classifiers that use the extracted features by DenseNet. In comparison Table 5 with Table 6, it is clear that the features extracted by DenseNet improve the performance of all these recognizers significantly. This indicates that the features extracted by the transfer learning-based DenseNet have good class discrimination, and then improves recognition performance of these typical recognizers (e.g., SVM, RF). Thus, feature learning of the transfer learning-based DenseNet plays a key role in improving the recognition performance of GCForest. In particular, DenseNet-GCF still shows the best result among all models. This indicates that GCForest is effective to further improve the WMDR performance based on high-dimensional abstract features extracted by DenseNet. Thus, GCForest is very effective as a feature extractor and recognizer in the DenseNet-GCF model for WMDR.

7. Conclusions

This paper proposes a transfer learning-based CNN model, DenseNet-GCF for WMDR. The experimental results on the industrial dataset show that the recognition model has better performance than other typical classifiers. The proposed method has the following advantages: (1) The extracted features directly from the original wafer maps by the deep CNN are very effective to improve the classifier performance; (2) By transferring the pre-trained feature extraction model, the CNN model greatly reduces the training time and improves the image feature extraction performance; (3) The deep forest is very effective to classify high-dimensional abstract features. In comparison with those famous CNNs and traditional classifiers, DenseNet-GCF shows better classification performance on wafer maps. This study provides the guidance to transfer learning-based CNN for feature learning from wafer maps.

CRediT authorship contribution statement

Jianbo Yu: Project administration, Resources, Methodology, Conceptualization, Formal analysis, Funding acquisition, Supervision. **Zongli Shen:** Software, Investigation, Writing – original draft, Validation, Visualization, Data curation. **Shijin Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by Fundamental Research Funds for the Central Universities.

References

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

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

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

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

Chen, C.H., Kim, H., Piao, Y., Li, M., Piao, M., 2020. Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes. J. Intell. Manuf. 31, 1861–1875.

Donahue, J., Jia, Y., Vinyals, O., et al., 2013. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. 50(1), I-647.

Dong, Y., 2019. Implementing deep learning for comprehensive aircraft icing and actuator/sensor fault detection/identification. Eng. Appl. Artif. Intell. 83, 28–44.

Guo, Y.C., 2009. Knowledge-enabled short-term load forecasting based on pattern-base using classification and regression tree and support vector regression. In: Fifth International Conference on Natural Computation. IEEE, Tianjin, pp. 425–429.

He, K., Zhang, X., Ren, S., et al., 2015. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Hu, G.Z., Li, H.F., Xia, Y.Q., et al., 2018. A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis. Comput. Ind. 100, 287–296.

Huang, G., Liu, Z., Laurens, V.D.M., et al., 2016. Densely connected convolutional networks.

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

Jiang, G., He, H., Xie, P., Tang, Y., 2017. Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis. IEEE Trans. Instrum. Meas. 66, 2391–2402.

Kim, B., Jeong, Y.S., Tong, S.H., Chang, I.K., 2015. 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), 41–49.

Kim, B., Jeong, Y.S., Tong, S.H., Chang, I.K., Jeong, M., 2016. Step-down spatial randomness test for detecting abnormalities in DRAM wafers with multiple spatial maps. IEEE Trans. Semicond. Manuf. 29 (1), 57–65.

Koo, H.I., Cho, N.I., 2010. New automatic defect classification algorithm based on a classification-after-segmentation framework. J. Electron. Imaging 19 (2), 334–343.

Krizhevsky, A., Sutskever, I., Hinton, GE., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 1097–1105.

Kyeong, K., Kim, H., 2018. Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. IEEE Trans. Semicond. Manuf. 1.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

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

Liu, C., Chien, C., 2013. An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing. Eng. Appl. Artif. Intell. 26 (5), 1479–1486.

Liu, F.T., Ting, K.M., Yu, Y., et al., 2008. Spectrum of variable-random trees. J. Artificial Intelligence Res. 32 (1), 355–384.

Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 1 (2008), 1–48.

Mubarak, G., Abdu-Aguye, Walid, G., 2018. Novel approaches to activity recognition based on vector autoregression and wavelet transforms. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA).

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

Ooi, M.P.L., Sok, H.K., Kuang, Y.C., Demidenko, S., Chan, C., 2013. Defect cluster recognition system for fabricated semiconductor wafers. Eng. Appl. Artif. Intell. 26 (3), 1029–1043.

Pan, S.J., Yang, Q., 2010a. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Pan, S.J., Yang, Q., 2010b. A survey on transfer learning, status and development of transfer learning based category-level object recognition and detection. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Saqlain, M., Jargalsaikhan, B., Lee, J.Y., 2019. A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. IEEE Trans. Semicond. Manuf. 32 (2), 171–182.

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

Srivastava, R.K., Greff, K., Schmidhuber, Jürgen, 2015. Highway networks. Comput. Sci. arXiv:1505.00387.

Szegedy, C., Liu, W., Jia, Y., et al., 2015. Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:1409.4842 [cs.CV].

Thirukovalluru, R., Dixit, S., evakula, RK., Verma, NK., Salour, A., 2016. Generating feature sets for fault diagnosis using denoising stacked auto-encoder. In: Prognostics and Health Management (ICPHM), IEEE International Conference. pp. 1–7, 2016.

Wang, R., Chen, N., 2019. Wafer map defect pattern recognition using rotation-invariant features. IEEE Trans. Semicond. Manuf. 32 (4), 596–604.

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

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

Xie, L., Huang, R., Gu, N., Zhi, C., 2014. A novel defect detection and identification method in optical inspection. Neural Comput. Appl. 24 (7–8), 1953–1962.

Yang, F., Xu, Q., Li, B., et al., 2018. Ship detection from thermal remote sensing imagery through region-based deep forest. IEEE Geosci. Remote Sens. Lett. (2018), 1–5.

Yu, J., Lu, X., 2016. Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. IEEE Trans. Semicond. Manuf. 29 (1), 33–34.

Yu, J., Shen, Z., Zheng, X., 2021. Joint feature and label adversarial network for wafer map defect recognition. IEEE Trans. Automat. Sci. Eng. 18 (3), 1341–1353.

Yu, N., Xu, Q., Wang, H., 2019. Wafer defect pattern recognition and analysis based on convolutional neural network. IEEE Trans. Semicond. Manuf. 32 (4), 566–573.

Zhao, L., Wang, J., Nabil, MM., Zhang, J., 2018. Deep forest-based prediction of protein subcellular localization. Curr. Gene Ther. 18 (5), 268-274(7).

Zhou, Z.H., Feng, J., 2017. Deep forest: Towards an alternative to deep neural networks. https://arxiv.org/abs/1702.08835.