Research on Wafer Surface Defect Pattern Detection Method Based on Incremental Learning

Naigong Yu1,2, Xin Li1,2, Qiao Xu1,2, Kai Jiang3

1 Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
E-mail: yunaigong@bjut.edu.cn
2 Beijing Key Lab of the Computational Intelligence and Intelligent System, Beijing 100124, China
E-mail: yunaigong@bjut.edu.cn
3 Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China
E-mail: 10000114@chnenergy.com.cn

Abstract: Wafer manufacturing is an important step in quality control and analysis in the semiconductor industry. The defect pattern classification algorithm of wafer maps has received extensive attention from academia and industry. At present, most methods for detecting wafer surface defect patterns focus on static data model classification and analysis. However, in the production process, static data models cannot satisfy the dynamic analysis of wafer defect patterns in the form of streaming data. In this regard, this paper proposes a wafer surface defect pattern detection method based on incremental learning. Our experiment uses Resnet as the backbone network, and the data set uses the WM811K wafer data set. Experiments have proved that our method can achieve better classification accuracy in the field of wafer defect detection, which provides the possibility for continuous learning of wafer defects in the future.

1. INTRODUCTION

Wafers are an important carrier for integrated circuit manufacturing. Many process problems in integrated circuit manufacturing will result in the formation of specific defect patterns on the wafer map. In recent years, due to technological progress, wafer testing [1] has been used to analyze and evaluate the electrical performance of the chips in the wafer. According to the results of this step, the defective chips are filtered out, and the wafer space data obtained becomes the wafer Warehouse map (WBM). Common wafer defect cluster patterns in the manufacturing process include Center, Edge-Loc, Edge-Ring, Loc, Scratch Random and None patterns. Figure 1 shows examples of these patterns. These patterns are included in the published WM811K wafer map data set.

The traditional wafer defect pattern detection and analysis is performed manually using an electron microscope. This method is based on the experience of the operator, and has disadvantages such as low accuracy, slow operation speed, and high labor cost. In recent years, the algorithm for wafer defect pattern detection has been manually extended to machine learning [2-4], especially with the rise of deep learning [5-8], a large number of deep convolutional neural networks are used to perform wafer maps.
The detection algorithm has made unprecedented progress. Although the deep learning method improves the overall accuracy in the detection of wafer defect patterns, the model cannot perform incremental learning when facing new data, which in turn affects the judgment of defect categories. This is very limited in actual production.

This article proposes an increase based on depth. Detection method of wafer defect pattern based on quantitative learning. This method can perform dynamic analysis of wafer defect patterns in the form of streaming data, and can realize incremental updates of neural network models using only new class samples, and avoid catastrophic forgetting.

2. Materials and Methods

2.1 Image Processing
We use The constrained mean filtering (C-mean filtering) algorithm [9] to reduce the noise of the original samples. CMF is an improved mean filtering algorithm, which only filters the defective grains and protects the edges and normal The crystal grains are not destroyed.

The filtering result of CMF will not introduce new elements, only background, defective grains and normal grains are present. The samples are processed by a 3×3 filter window and a CMF with a mean threshold of 1.25 for our subsequent training and testing. The processed wafer map defect sample is shown in Figure 2.

2.2 Backbone
We choose Resnet as the backbone network for wafer map classification. The core content of Resnet is residual learning. Because the wafer map is relatively simple, we choose Resnet-18 with a shallower level as the backbone network. The input wafer map size is 224×224, the first layer of convolution is composed of a 112-channel 7×7 convolution kernel, followed by a maximum pooling layer for next use. The residual learning unit is composed of two 3×3 convolutional layers, each convolution module contains two residual units. Finally, the global average pooling layer (GAP) is used for downsampling, and a fully connected layer of 9 neurons is connected, and the softmax operation is performed on the output layer.

2.3 Algorithm
There are many types of incremental learning algorithm update fast methods, and a suitable learning algorithm needs to be selected for incremental training on the wafer data set. iCaRL is a learning algorithm that increases recognition types. Its idea is based on playback, also known as rehearsal-based. When training new tasks, part of the representative old data will be retained and used for the model to
review the old knowledge learned. It is characterized by adopting the nearest average algorithm rule of samples in the classification stage, selecting the old class examples using the herding method, and using cross-entropy loss and distillation loss as the total loss function.

In the initial stage of training, use the feature extractor to extract the feature vectors of all new data and part of the old data, and calculate their average feature vectors:

\[
\mu_y = \frac{1}{P_y} \sum_{p \in P_y} \vartheta(p)
\]

where \(\mu_y\) is the average feature vector, \(P\) is the stored set of old and new classes.

In the incremental learning stage, the predicted value of the new and old data is calculated through NME:

\[
y^* = \arg \min \| \vartheta(x) - \mu_y \|
\]

After getting \(y^*\), calculate the loss function \(\ell(\theta)\) for parameter update during characterization learning:

\[
\ell(\theta) = - \sum_{(x,y) \in D} \left[ \sum_{j=y} \delta_{y=j} \log g_j(x) + \delta_{y\neq j} \log(1-g_j(x)) + \sum_{j=1} \left( q_j^y \log g_j(x) + (1-q_j^y) \log(1-g_j(x)) \right) \right]
\]

Where the upper part of the loss is the cross-entropy loss of the new class, and the lower part is the distillation loss of the stored old class. \(D\) is the set of the new category and the old category, \(P_1 \sim P_s\) is the old category set, and \(P_s \sim P_t\) is the new category set. \(q_j^y\) is the output of the softmax layer on the old category.

3. Experiments and results

3.1 WM811K dataset

Our experiment is based on the WM811K wafer data set training and testing model, which is by far the largest public data set. The data set is derived from the actual production process and contains a total of 811,457 samples and 9 defect patterns. We divide it into 5 old sample categories and 4 new sample categories. All samples are divided into 60% training set, 15% validation set and 25% test set. The specific distribution diagrams of the old and new samples are shown in 3 below. Since there are many samples in the None mode, only 3000 samples are selected.

![Fig.3 Data division of new and old defect samples](image)

3.2 Framework

The incremental learning process has requirements on the algorithm and the deep CNN model. In the
incremental training, we use the iCaRL incremental algorithm to require the participation of old samples, so the incremental learning framework can be simplified as shown in Figure 4:

![Incremental learning framework based on ResNet](image)

3.3 Experimental setup
The initial learning rate of the model is set to 2.0, the maximum number of old sample samples that can be stored is set to 2000, the number of epochs for each training iteration is 100, and the images loaded for each training are set to 64. At the beginning of the increment, the learning rate is reduced by 5 times from the initial 2.0, after 48, 62, and 80 epochs. During the experiment, set the weight attenuation to 0.00001 to optimize the Stochastic Gradient Descent (SGD). Since there are only 4 new classes, we load a new defect every time for training.

3.4 Result analysis
Before the increment starts, we define the classification and recognition phase of the five defects of the model as phase 0, then phase 1 represents a new type of defect, and so on, phase 4 represents the recognition of all defect categories. Due to the imbalance of category samples, the average accuracy evaluation model is not reliable enough, so Precision, Recall, and F1-Score are used to evaluate the model. F1-Score is the harmonic average of Precision and Recall, which can comprehensively evaluate the performance of the model. The recognition accuracy of defects in each stage of incremental training is shown in Table 1.

| Phase | Precision | Recall | F1-score | F1 average |
|-------|-----------|--------|----------|------------|
| Phase 0 | 0.912 | 0.912 | 0.912 | |
| Phase 1 | 0.902 | 0.917 | 0.909 | |
| Phase 2 | 0.869 | 0.889 | 0.878 | 0.887 |
| Phase 3 | 0.865 | 0.865 | 0.865 | |
| Phase 4 | 0.862 | 0.881 | 0.872 | |

As the number of incremental training of the model increases, the comprehensive index of the model shows a downward trend, which shows that the model's ability to recognize defects decreases.

There is no uniform definition for the evaluation index of incremental learning. We use the average classification accuracy (ACC) of the model, the memory ability of the model, and the migration ability of the model as the evaluation index. We have verified Table 2 on the validation set.

| Comparison table of model's memory ability and transfer ability |
|---------------------------------------------------------------|
| Phase 0 | Phase 1 | Phase 2 | Phase 3 | Phase 4 |
| Center | 0.986 | 0.980 | 0.980 | 0.977 | 0.973 |
| Donut | 0.892 | 0.928 | 0.935 | 0.921 | 0.942 |
| Pattern   | 0.892  | 0.870  | 0.857  | 0.856  | 0.831  |
|-----------|--------|--------|--------|--------|--------|
| Edge-Loc  | 0.958  | 0.950  | 0.943  | 0.945  | 0.942  |
| Edge-Ring | 0.850  | 0.851  | 0.808  | 0.818  | 0.791  |
| Loc       | 0.920  | 0.779  | 0.688  | 0.752  |
| Random    | 0.789  | 0.816  |
| Scratch   |        |        |        |
| Near-Full |        |        |
| None      |        |        |

We can see that as the increment progresses, the classification accuracy of the Center, Edge-Loc, Edge-Ring and Loc defect patterns is getting lower and lower, indicating that the model increases with the increment. For the new defect category, the recognition accuracy of Random and None patterns is higher, which means that these two defect patterns are similar to the defect patterns of the previous old samples. Edge-Ring has a high recognition rate. The None mode is highly similar to these two patterns. The model can easily identify similar features in the stored old sample features. Special attention is paid to the fact that the recognition accuracy of Donut pattern does not decrease significantly as the increment increases, indicating that this type of defect is very different from other defects, so the model has a strong memory capacity for it. Due to the contingency of the training results, we use the Top-5 method to get the average classification accuracy of the model as shown in Figure 5.

![Fig.5 Average classification accuracy of the model](image)

The deep neural network model has the problem of forgetting when performing incremental learning. When we perform incremental learning of wafer defects, the classification accuracy of the model does not decrease when the Near-Full mode is added, indicating that the model has not extracted similar features, the model has a good recognizability for the Near-Full mode.

In order to understand the incremental training process more clearly, we visualized the results based on the Grad-weighted Class Activation Mapping (Grad-CAM) [10] algorithm. Figure 6 is a heat map generated by Grad-CAM for the last convolutional layer of each incremental stage. It can be found that as the incremental training phase continues to increase, the model forgets the key defect cluster features of the old category of defects more seriously. For example, the Edge-Loc and Loc patterns, the forgetting is very obvious in the final stage of the increment. For newly-added defect categories such as Random and None patterns, the model also has high classification and recognition accuracy.
Fig. 6: Different incremental phases heat map generated by Grad-CAM

4. Conclusion
Wafer defect pattern inspection is a key step in the semiconductor manufacturing process. Deep convolutional neural networks have greatly promoted the development of wafer inspection. Our work mainly verifies that incremental learning can dynamically analyze wafer defect patterns in the form of streaming data, can achieve incremental updates of neural network models using only new class samples, and avoid catastrophic forgetting. However, although the model performance is good, it has not reached an excellent level. We hope that future incremental research on wafer surface defect pattern detection can focus on improving the model’s recognition accuracy of new defect categories and the ability to maintain old defect categories, and further promote the incremental research on wafer surface defect detection.

Acknowledgments
This work was supported by the Ministry of Industry and Information Technology of China (Grant No. Z135060009002) and the project jointly funded by Beijing Municipal Education Commission and Municipal Natural Science Foundation (Grant No. KZ202010005004).

References
[1] Shankar N G., Zhong Z W. (2005) Defect detection on semiconductor wafer surfaces[J]. Microelectronic Engineering, 77(3/4): 337-346.
[2] Ming J W., Jang J R., Chen J L. (2015) Wafer map failure pattern recognition and similarity ranking for large-scale data sets[J]. IEEE Transactions on Semiconductor Manufacturing, 28(1): 1–12.
[3] Saqlain M., Jargalsaikhan B., LEE, Lee J Y . (2019) A Voting Ensemble Classifier for Wafer Map Defect Patterns Identification in Semiconductor Manufacturing[J]. IEEE Transactions on Semiconductor Manufacturing, 32(2): 171-182.
[4] Baly R., Hajj H. (2012) Wafer Classification Using Support Vector Machines[J]. IEEE Transactions on semiconductor manufacturing, 25(3): 373-383.
[5] Nakazawa T., Kulkarni D V. (2018) Wafer Map Defect Pattern Classification and Image Retrieval using Convolutional Neural Network[J]. IEEE Transactions on Semiconductor Manufacturing, 31(2): 309-314.
[6] Nakazawa T., Kulkarni D V. (2019) A anomaly Detection and Segmentation for Wafer Defect Patterns using Deep Convolutional Encoder-Decoder Neural Network Architectures in
Semiconductor Manufacturing[J]. IEEE Transactions on Semiconductor Manufacturing, 32(2):250-256.

[7] Xin F.(2018).Research on Defect Detection and Classification of Wafer SEM Image Based on CNN(Master's degree thesis, Zhejiang University).

[8] Jiatong L., Jianbo Y. (2020) Two-Dimensional Principal Component Analysis Convolutional Autoencoder for Pattern Recognition of Wafer Surface Defects[J/OL].Journal of Computer Aided Design and Graphics, 32(03):425-436.

[9] Naigong Y., Qiao X., Honglu Wang. (2019) Wafer Defect Pattern Recognition and Analysis Based on Convolutional Neural Network[J]. IEEE Transactions on Semiconductor Manufacturing.

[10] Islam, A., Jia, S., & Bruce, N. (2020). How much Position Information Do Convolutional Neural Networks Encode.