Silicon Wafer Map Defect Classification Using Deep Convolutional Neural Network With Data Augmentation

Conference Paper · December 2019

DOI: 10.1109/ICCC47050.2019.9064029

Ashadullah Shawon
SELISE rockin' software
11 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Md. Omar Faruk
Daffodil International University
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Silicon Wafer Map Defect Classification Using Deep Convolutional Neural Network With Data Augmentation

Ashadullah Shawon*, Md Omar Faruk, Masrur Bin Habib, Abdullah Mohammad Khan
Software Engineer, Frontier Semiconductor,
Dhaka, Bangladesh
e-mail: shawona@frontiersemi.com, omarf@frontiersemi.com, masrurh@frontiersemi.com, hbkamk@gmail.com

***Abstract*—Wafer map defect classification is one of the most important process for semiconductor manufacturing. Wafer map defects are also cause of die failures. So, we propose a better method for classifying defects of wafer map. We propose deep convolutional neural network architecture and our proposed architecture is the improved version of previous researcher's architecture. We also solve the data imbalance problem by applying data augmentation technique before training of data. The test accuracy of our research is 99.29% which is the best performance so far.**

***Keywords*—wafer map defects, classification, deep learning, data augmentation, deep convolutional neural network**

I. Introduction

Wafer defect analysis from wafer map is a critical task in semiconductor manufacturing industry. The defects are analyzed from the wafer map visualization. The patterns from wafer map visualization are important for the classification. As efficient and effective wafer tools are in high demand [1], the wafer map defect classification is one of the most demandable tool for the semiconductor industry. Because the defects are analyzed by computer vision based software only. So it also reduces the hardware cost.

Wafer map pattern recognition technology has been developing gradually and previous approaches [2-4] were not large scale dataset based for lower accuracy. Before 2018, some previous researchers applied machine learning based algorithms for wafer defect classification [5]. But from 2018, the researchers [6-7] started to apply deep learning based algorithm like convolutional neural network (CNN) for wafer map defect classification. Researchers increased the accuracy by applying convolutional neural network (CNN). But the CNN architecture can be improved more for better accuracy. Besides there were some problems with data balancing that was not solved by the previous researchers.

So, our purpose of this research is to achieve better accuracy by presenting a better CNN architecture and solving the data imbalance problem. Our contributions are mainly in improvement of CNN architecture and data augmentation technique.

Our paper is organized in the following way. Section II describes the related work of our research. Dataset and preprocessing are analyzed in section III. In section IV, we have presented our proposed approach. Our main contribution that is data augmentation and CNN architecture are described in section V and section VI. Experiment results are showed in section VII and finally we have concluded our research paper in section VIII.

II. Related Work

There are significant related works in wafer map defect classification. Before 2018, the approaches were wafer based clustering [8], region based modeling [9], and spatial signature analysis [10] etc. In wafer based clustering, new failure pattern can be introduced. As clustering is unsupervised learning, so it can detect the failure pattern in unsupervised way. In region based modeling, multi pattern failures can be modeled in a single wafer map. In spatial signature analysis, K-nearest neighbor classifier has been applied for classification. In 2012, some researchers applied support vector machine for wafer map defect classification [5]. In 2015, similarity ranking was used for large scale wafer map defect dataset [11]. But recently which is from 2018, the researchers have applied convolutional neural network for the classification [6-7]. As deep convolutional neural network have recently gained the state of the art accuracy for image classification [12], so deep convolutional neural network has become a trending classification system. So, we also have applied deep convolutional neural network for the classification. Additionally we have improved the CNN architecture, parameters, and data augmentation.

III. Dataset and Preprocessing

We use the most popular WM-811K Wafer Map [13] dataset which was also used by previous researchers. This dataset has 811,457 wafer maps which was collected from 46,393 lots in real world fabrication. There are total 9 types of defect failure pattern. The defects type in this dataset are: Center, Donut, Edge-Loc, Edge-Ring, Loc, Random, Scratch, Near-full, none. Fig.1 shows the visualization of the defects type.

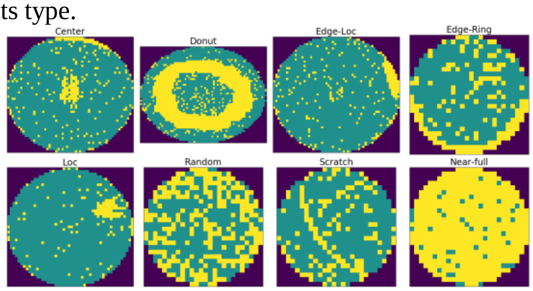


Figure 1. Defects type of WM-811K dataset.

Fig.2 shows the failure type frequency of the WM-811K dataset.

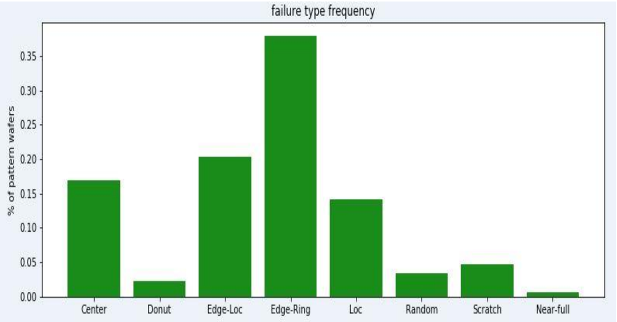


Figure 2. Failure type frequency of WM-811K dataset

In this dataset, 78.7% wafers are with no-label, 3.1% wafers have real failure patterns, and 18.2% wafers are labeled with none category. Figure 3 shows the visualization

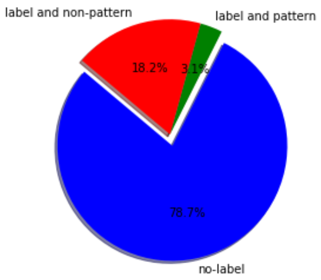
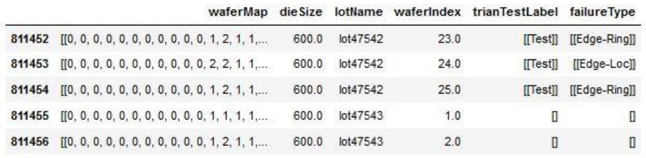


Figure 3. Visualization of dataset label

So, here we focus on only wafers that are labeled, patterned, and non-patterned. As a result 25,519 wafers are only eligible for the classification process. Besides wafer map, there are also some information in this dataset. Figure 4 shows the columns of the dataset.



	waferMap	dieSize	lotName	waferIndex	trianTestLabel	failureType
811452	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1,...	600.0	lot47542	23.0	[[Test]]	[[Edge-Ring]]
811453	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 1, 1,...	600.0	lot47542	24.0	[[Test]]	[[Edge-Loc]]
811454	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1,...	600.0	lot47542	25.0	[[Test]]	[[Edge-Ring]]
811455	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,...	600.0	lot47543	1.0	[]	[]
811456	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1,...	600.0	lot47543	2.0	[]	[]

Figure 4. Columns of dataset

All types of informations are not necessary here, so we delete unnecessary columns from the dataset like wafer index.

IV. Proposed Approach

Recently deep convolutional neural network is providing outstanding performance in the field of computer vision applications. Deep learning mainly extracts the features automatically. But the performance of deep CNN depends on the architecture. So, we have proposed a deeper architecture that gives us better performance than previous. Only deeper architecture cannot ensures better performance. Removing data imbalance problem, enhance the training of CNN, improvement of generalization, and increasing the robustness of the model are also important for the best performance and data augmentation is used for this criteria [14]. So, we have applied data augmentation before training. Figure 5 shows our proposed approach.

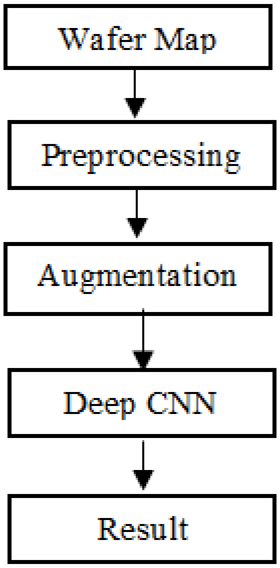


Figure 5. Our proposed method

Data augmentation technique is described in section V and deep learning architecture, parameters are described in section VI.

V. Data Augmentation

Performance of deep learning is often improved by the amount of more data [14]. So it needs to augment the existed training data artificially by some techniques. These techniques are called data augmentation. By applying domain specific techniques to the samples from the training data can create new training samples. Besides, to solve class imbalanced problem we have used data augmentation. As the wafer data is image data, so we have used convolutional autoencoder for data augmentation. Table I shows our convolutional autoencoder architecture summary.

TABLE I. Convolutional Autoencoder

Layer	Output Shape	Parameters
Input_1	(None,26,26,3)	0
Conv2d_1	(None,26,26,64)	1792
Maxpooling2d_1	(None,13,13,64)	0
Conv2dTranspose_1	(None,13,13,64)	36928
Upsampling2d_1	(None,26,26,64)	0
Conv2dTranspose_2	(None,26,26,3)	1731

We have made 2000 samples of new wafer for each case. Noise has been also removed during augmentation. After generating new samples for dataset, the augmented data is concatenated with original data. Now the data has 30707 images. As none class is not important here, so we can exclude none images. So, the new data has 19707 images. The 19707 images are divided into three dataset. These are: training, validation, and testing. The training dataset has 12730 images, validation dataset has 6270 images, and testing dataset has 705 images.

VI. Deep Convolutional Neural Network

A. Architecture of Model

We have proposed an architecture of model that contains 3 convolutional layers with 2 fully connected dense layer and one dense output layer. The first layer has 16 filters and the filter size is 3×3. The second layer has 64 filters with 3×3 filter size. The third convolutional layer has 128 filters and the filter size is also 3×3. Maxpooling layer with 2×2 pool size is used after every layer. The maxpooling layer mainly extracts maximum value feature. Rectified linear unit (Relu) [15] is used as an activation function for non-linearity. The first fully connected dense layer has 512 filters and second fully connected dense layer has 128 filters. The last output dense layer has 9 filters for the 9 types of defect pattern. Softmax activation function is used as a last activation function for classification. Adam [16] optimizer is a stochastic optimizer that updates weights efficiently. We have used categorical cross entropy for error or loss calculation. Fig.6 shows our proposed CNN architecture.

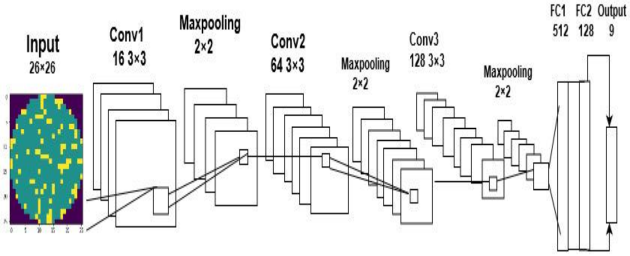


Figure 6.Our proposed CNN architecture

Our proposed CNN architecture can be understood clearly by the model summary. Table II shows our model summary.

TABLE II. Model Summary

Layer	Output Shape	Parameters
Input_1	(None,26,26,3)	0
Conv2d_1	(None,26,26,16)	448
Maxpooling2d_1	(None,13,13,16)	0
Conv2d_2	(None,13,13,64)	9280
Maxpooling2d_2	(None,6,6,64)	0
Conv2d_3	(None,6,6,128)	73856
Maxpooling2d_3	(None,3,3,128)	0
Flatten_1	(None,1152)	0
Dense_1	(None,512)	590366
Dense_2	(None,128)	65664
Dense_3	(None,9)	1161

B. Training Model Parameters

There are some parameters for this CNN training model. These parameters are also important for the performance. Table III shows the parameters.

TABLE III. Training Model Parameters

Parameters Name	***Value***
Learning rate	10^{-3}
Batch Size	1024
Epoch	30
Shuffle	True

Low learning rate helps to avoid local minima and so we keep the learning rate closer to 10^{-3}

VII. Experiment and Result Analysis

A. Experimental Environment

The experimental environment of our research is configured with Intel core i9 processor, Tesla k80 GPU, and 12 GB of RAM. Thus it has become possible to reduce the training time and increase the performance.

B. Training, Validation, and Testing

After data augmentation, the 19707 images are divided into training, validation, and testing dataset. The training dataset contains 12630 images. The validation dataset contains 6270 images, and the testing dataset contains 705 images.

C. Result Analysis

As our dataset is divided into training, validation, and testing, so there are three types of result of our research. These are: training accuracy, validation accuracy, and testing accuracy. We have found our result after 30 epoch. Table IV shows our result.

TABLE IV. Experimental Result

Training	**Validation**	**Testing**
99.81%	99.30%	99.29%

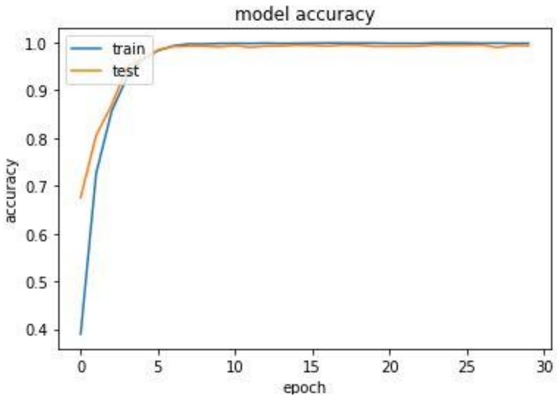


Figure 7. CNN model accuracy curve

Figure 7 and Figure 8 shows the model accuracy curve and model loss curve. Blue curve is for training and orange curve is for testing. In the figures, epoch means number of iteration for training our CNN model. We have used 30 epochs that means we have trained our model approximately 30 times.

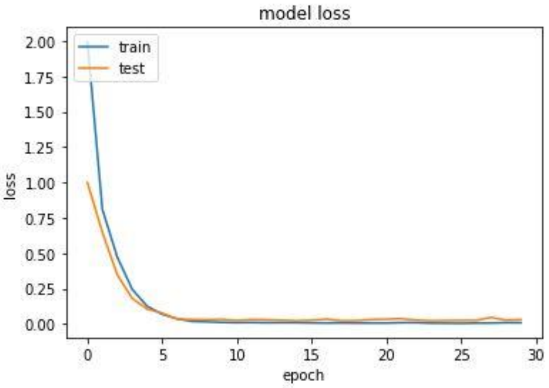


Figure 8. CNN model loss curve

We have also checked the over fitting of our model. Generally, when training accuracy increases but validation accuracy decreases then then it is called over fitting. But when the training accuracy curve and validation accuracy curve remains approximately same then it is called accurate model. Figure 9 shows our model training accuracy vs validation accuracy curve.

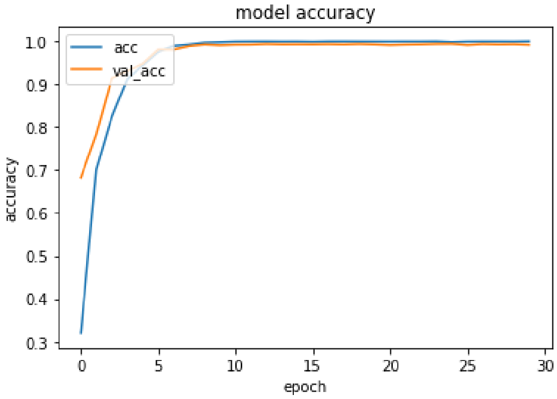


Figure 9. Training accuracy vs validation accuracy

From this figure, we can see that our model's training accuracy curve and validation accuracy curve are approximately in same line after 5 epochs. So our model is not over fitting.

Finally, we have compared our testing accuracy with most recent research papers [6-7]. Table V shows the comparison of our results with most recent research paper's result. The results are based on final testing accuracy.

TABLE V. RESULT COMPARISON WITH MOST RECENT RESEARCH

Deep CNN [6]	**Deep CNN [7]**	**Proposed Deep CNN+ Data Augmentation**
98.20%	97.40%	**99.29%**

Data augmentation has been using in image classification based research for achieving a promising test accuracy [17]. So we have also used in our research and the data augmentation technique actually helps the model to perform better. By comparing our result with previous result, we can conclude that our result performs better than previous researchers result.

VIII. CONCLUSION

In this paper, we have showed a better approach for wafer map defect classification. We have presented a deep CNN architecture with data augmentation that removes the data imbalance problem and performs better than previous approaches of researchers. Data augmentation techniques has boosted our proposed CNN model performance. That's the reason behind 99.29% state of the art testing accuracy. But there are also some limitations. We have not tested our model by external dataset. Besides the autoencoder model can be improved more in future. We recommend to continue this research according to future work direction.

REFERENCES

[1] Q. P. He and J. Wang, "Large-scale semiconductor process fault detection using a fast pattern recognition-based method," *IEEE Trans. Semicond. Manuf.*, vol. 23, no. 2, pp. 194–200, May 2010.

[2] T. Yuan, W. Kuo, and S. J. Bae, "Detection of spatial defect patterns generated in semiconductor fabrication processes," *IEEE Trans. Semicond.Manuf.*, vol. 24, no. 3, pp. 392–403, Aug. 2011.

[3] Y.-S. Jeong, S.-J. Kim, and M. K. Jeong, "Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 4, pp. 625–637, Nov. 2008.

[4] J. Y. Hwang and W. Kuo, "Model-based clustering for integrated circuit yield enhancement," *Eur. J. Oper. Res.*, vol. 178, no. 1, pp. 143–153, Apr. 2007

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

[6] T. Nakazawa and D. V. Kulkarni, "Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network," in *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 309-314, May 2018.

[7] K. Kyeong and H. Kim, "Classification of Mixed-Type Defect Patterns in Wafer Bin Maps Using Convolutional Neural Networks," in *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 3, pp. 395-402,Aug.2018.

[8] F. L. Chen and S. F. Liu, "A neural-network approach to recognize defect spatial pattern in semiconductor fabrication," *IEEE Trans. Semicond.Manuf.*, vol. 13, no. 3, pp. 366–373, Aug. 2000.

[9] J. Y. Hwang and W. Kuo, "Model-based clustering for integrated circuits yield enhancement," *Eur. J. Oper. Res.*, vol. 178, no. 1, pp. 143–153,Apr.2007

[10] K. W. Tobin, S. S. Gleason, T. P. Karnowski, S. L. Cohen, and F. Lakhani, "Automatic classification of spatial signatures on semiconductor wafer maps," in *Proc. Metrol. Insp. Process Control Microlith.*, Santa Clara, CA, USA, 1997, pp. 434–444

[11] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015.

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

[13] MIRLAB, “WM-811K wafer Map”. [Online]. Available: http://mirlab.org/dataSet/public/

[14] H.Garc'ia, Alex and P. Konig. “Further advantages of data augmentation on convolutional neural networks.” (2019).

[15] V.Nair, and Hinton, G. E. “Rectified linear units improve restricted boltzmann machines.” Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010.

[16] D. Kingma, and J. Ba, “Adam: A method of stochastic optimization”, arXiv:1412.6980[cs.LG], 2015.

[17] L. Perez, and J.Wang, “The Effectiveness of Data Augmentation in Image Classification using Deep Learning”, arXiv:1712.04621v1 [cs.LG], 2017.