Using GAN to Improve CNN Performance of Wafer Map Defect Type Classification

Yield Enhancement

YongSung Ji [a b]
[a] Sungkyunkwan University, Suwon, Korea,
adek0208@g.skku.edu
[b] Samsung Electronics, Hwasung, Korea

Jee-Hyong Lee [c †]
[c] Sungkyunkwan University, Suwon, Korea,
john@skku.edu

***Abstract* – Semiconductor wafer map data provides valuable information for semiconductor engineers. Correctly classified defect patterns in wafer maps can increase semiconductor productivity. Convolutional Neural Networks (CNN) achieved excellent performance on computer vision and were frequently used method in wafer map classification. The CNN-based classifier of the wafer map defect pattern requires a sufficiently large training set to ensure high performance. However, for the real semiconductor production environment, it is challenging to collect various defect patterns enough. In this paper, we propose a method to supplement the lack of training set using Generative Adversarial Networks (GAN) to improve the performance of the classifier. We measure our performance on the *'WM-811k'* dataset, which consists of 811K real-world wafer maps. We compare the performance of our classifiers with commonly used augmentation techniques. As a result, we achieved remarkable performance enhancement from 97.0% to 98.3%.**

Key Words – Generative Adversarial Networks, Wafer Defect Map Classification, Data Augmentation, Convolutional Neural Networks

I. INTRODUCTION

As the semiconductor industry develops, the semiconductor manufacturing process becomes more complex and very sophisticated. Minor differences in each process step can cause a greater impact on the process uniformity of each chip in the wafer and can lead to significant semiconductor chip defects. In particular, defects in mass-produced products lead to substantial losses for the manufacturing company.

The semiconductor production process consists of wafer fabrication, wafer-level test, assembly, and final test. Wafer fabrication goes through many steps, including doping, oxidation, etching, and deposition. At the wafer manufacturing stage, minute differences can cause defects in the semiconductor chip; however, we can usually identify these defects at the wafer-level test stage.

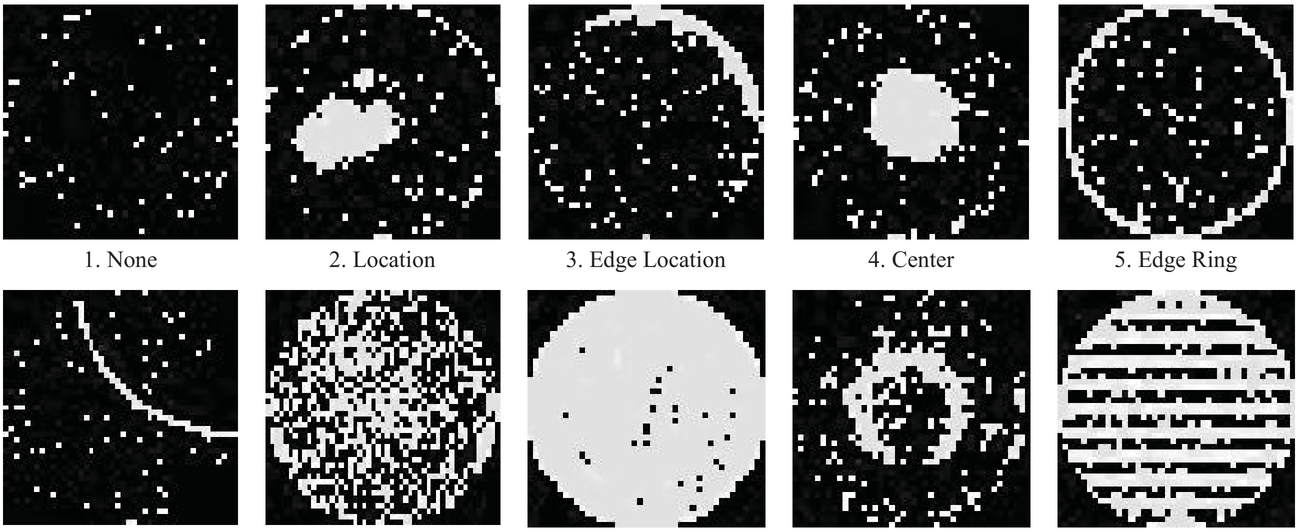
The wafer map generated after the wafer-level test gives the engineer lots of information. It records the electrical/structural characteristics of each chip and shows the pass-fail of each chip in the shape of the wafer. In particular, repeated patterns of fail chips can associate with steps of specific fabrication stages. Analyzing defect patterns in the wafer map to find and feedback defects in a particular step can help to improve productivity and can provide a variety of benefits throughout semiconductor wafer manufacturing.

Convolutional neural networks (CNN) [1] have improved the performance of image classification and currently used for many areas of computer vision. CNNs are also using in the semiconductor field for the classification of wafer map failure patterns based on deep learning [2-8]. In general, the more training data, the higher the performance of the CNN-based classifier model. However, it is difficult to collect various types of wafer defect maps; therefore, it leads to the poor performance of the CNN model.

Generative adversarial networks (GAN) [9] used for a generative model via two different networks: Generator and Discriminator. Generator generates a fake image, capturing the data distribution. Discriminator identifies real and fake images. By doing this repeatedly, the generator network can generate more realistic images. GANs have gained great popularity in the computer vision community and continues to introduce various models to generate high quality, realistic images. We intend to use GANs to add defect pattern wafer maps to supplement the lack of data among the wafer classes to be classified by the CNN model.

In this paper, we utilize CNNs to classify wafer map defect patterns. Wafer map data is composed of a total of 8,560 wafers for 10 classes in the *WM-811k* dataset. There is enough data in certain classes, but some classes have insufficient data. The data imbalance of this dataset reflects the problem that frequently occurs in the actual semiconductor manufacturing environment. We proposed a method to supplement data for unbalanced classes using GANs.

This paper is organized as follows. Section □ describes how to generate wafer maps, configure CNNs, and configure GANs. In Section □, we use wafer map images generated in various ways to verify the results of CNN training/testing. And compare the accuracy of the CNN model according to the use of wafer map images secured by GANs. The conclusion is in Section □.



1. None 2. Location 3. Edge Location 4. Center 5. Edge Ring

6. Scratch 7. Random 8. Near Full 9. Donut 10. Horizontal Stripe

Fig. 1. Wafer map for each class from *WM-811k* dataset.

Table 1. List of the wafer map defect types.
(Bold refers that insufficient data)

Class	Defect Type	The # of data samples	
		Training set	Test set
T1	None	2,340	260
T2	Location	**440**	**60**
T3	Edge Location	**540**	**60**
T4	Center	**440**	**60**
T5	Edge Ring	1,800	200
T6	Scratch	**300**	**60**
T7	Random	**300**	**60**
T8	Near Full	**140**	**60**
T9	Donut	**180**	**60**
T10	Horizontal Stripe	1,080	120
Total		7,560	1,000

II. METHODOLOGY

A. *WM-811k Dataset Preprocessing*

We use the *WM-811k* dataset to train and test the CNN model. The original *WM-811k* dataset contains more than 810,000 wafer maps divided into nine classes. However, this dataset has many unlabeled data, and since the size of wafer maps varies, using this dataset directly may cause low performance on CNN performance.

Therefore, we preprocess the wafer maps contained in the *WM-811k* dataset to implement an efficient CNN model. In the first step, we fix the various data size into 64x64 images. Next, we mark the position of fail chips in white and mark all the rest, including pass chips in black. We also remove unclear or unlabeled data. Furthermore, we exclude wafer maps with too high/low-resolution from the dataset. This preprocessing process is a significant process to ensure stability for model training.

In the preprocessing step, the dataset consists of 10 classes by adding one characteristic fail pattern class called horizontal stripe. Since there is less wafer map data in a specific classification class, limit the number of data on the data-rich side to an appropriate level. Through this process, we obtain a total of 8,560 data samples consisting of 10 classes. There are enough wafer map data images in some classes, but there are classes that do not have sufficient wafer map data images. The wafer maps in the actual semiconductor industry show a similar composition to this dataset, where the class distribution is imbalanced. Fig. 1 shows 10 defect type classes, and Table 1 is the list and the number of wafer map defect types.

B. *CNN architecture*

Fig. 2 shows the structure of our proposed wafer map classification system. Each of the three convolutional layers consists of a max-pooling layer and a Rectified Linear Unit (Relu). All convolutional layers use a 3x3 kernel with a 2x2 max-pooling layer. After that, a fully connected (FC) layer with 256 nodes is connected. Finally, this FC layer is connected to another FC layer, and the class probability calculation is completed through the last softmax layer.

To evaluate the performance of the CNN model, we separate the data into a training set and a test set. We first divide the 8,560 data samples randomly by 9:1, and then set the ratio of the test set more for classes with fewer data samples. To train the CNN model, we proceed with 100 epochs with a learning rate of 0.001 and a batch size of 64. Table 2 shows the accuracy of each class in the CNN model using the *DS0*. The total accuracy is 97.0%. It shows high accuracies for the three classes with a large number of data in the training set. Contrarily, the classes with a small amount of data, especially the scratch class, have very low accuracies. However, “Random”, “Near-full”, and “Donut” classes show high accuracy even with a small amount of data due to the characteristics of the wafer map image.

C. Deep Convolutional GAN (DCGAN)

GANs consist of two neural networks that are trained simultaneously. The first network is called “Generator” and denoted by *G*. The role of *G* is to learn real samples and synthesize fake samples. The second network is “Discriminator” and denoted by *D*. The role of *D* is to distinguish between real and fake samples. Repeated learning can train *G* to produce fake samples that are hard to distinguish from real samples. Fig. 3 shows the structure of the GANs. Since the introduction of GANs, various researches have been conducted to improve performance, and DCGAN [10] is one of them. DCGAN is a type of GANs which consists of multiple layers of CNNs, used to generate realistic images. DCGAN uses CNNs' superior image interpretation capabilities to enhance GANs' image generation capabilities. Therefore, we follow the structure of DCGAN to obtain additional wafer map image data.

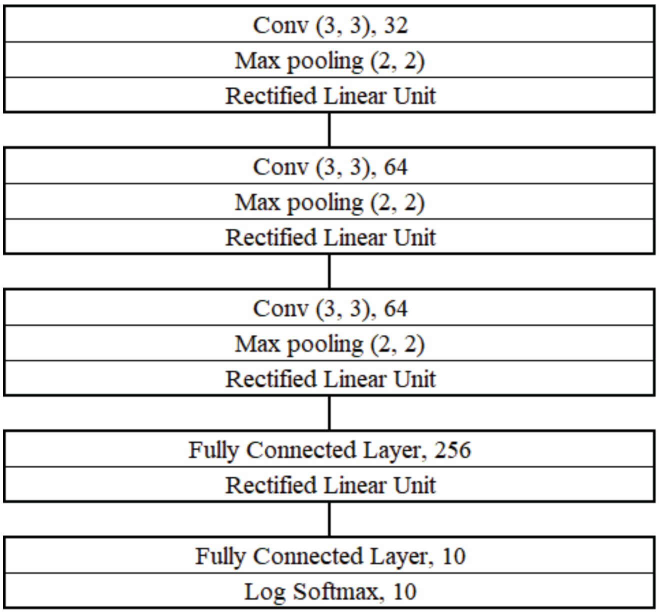


Fig. 2. Convolutional neural network architecture.

Table 2. Accuracy of confusion matrix for CNN model with original dataset (*DS0*).

Predicted Actual	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Accuracy (%)
T1	**258**	0	0	0	0	2	0	0	0	0	99.2
T2	1	**56**	1	1	0	0	0	0	1	0	93.3
T3	1	2	**56**	0	1	0	0	0	0	0	93.3
T4	0	1	0	**59**	0	0	0	0	0	0	98.3
T5	0	0	1	0	**199**	0	0	0	0	0	99.5
T6	5	11	1	0	0	**43**	0	0	0	0	**71.7**
T7	0	0	0	0	0	0	**60**	0	0	0	100.0
T8	0	0	0	0	0	0	0	**60**	0	0	100.0
T9	0	0	0	1	0	0	0	0	**59**	0	98.3
T10	0	0	0	0	0	0	0	0	0	**120**	100.0
											97.0

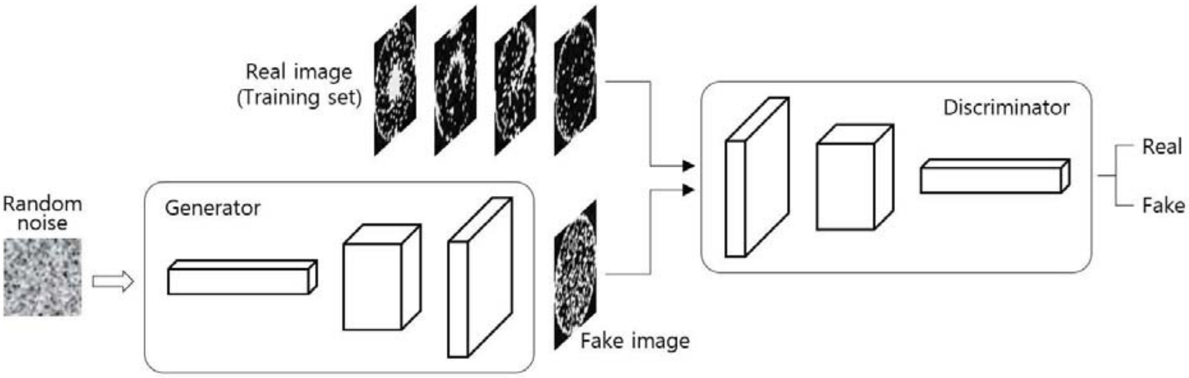


Fig. 3. Generative Adversarial Network architecture.

III. EXPERIMENTS

A. *Original Dataset (DS0)*

The dataset preprocessed after "Section □-A" is called the "original dataset" and is denoted by *DS0*. There are 7,560 training samples, 1,000 test samples, and the performance of the CNN model for *DS0* was confirmed to be 97.0% accuracy for 100 epochs. The accuracy of the scratch class is the lowest, at 71.7%.

B. *DCGAN Dataset (DS1)*

We use DCGAN to generate wafer map images to supplement classes with few data. DCGAN learns only the map data contained in the training set of the *DS0*. The classes that need additional data are the seven classes with less than 600 data samples in the training set prepared through preprocessing: Location, Edge Location, Center, Scratch, Random, Near Full, Donut. Fig. 4. shows the wafer map images generated by DCGAN. We add 1,400 wafer map images generated with DCGAN to the training set of *DS0*. The dataset now consists of 8,960 training samples and 1,000 test samples, and we denoted this data as *DS1*. The performance of the CNN model for *DS1* is 98.0% for 100 epochs. Fig. 5. shows the accuracy per epoch for each dataset.

C. *Classic Augmentation Dataset (DS2)*

We use the classic augmentation technique to compare the effect of DCGAN generated wafer map images on CNN performance. We use three classic techniques (left/right inversion, top/bottom inversion, and rotation) considering our dataset characteristics. The augmentation is also limited to the map data included in the training set of the *DS0*. We add 1,400 map data generated by augmentation to the training set of *DS0*. This dataset is referred to as *DS2*, and the number of data samples is the same as *DS1*. The performance of the CNN model on the *DS2* is 97.5% for 100 Epoch.

D. *DCGAN with Classic Augmentation Dataset (DS3, DS4)*

DCGAN generates fake images through training, and in general, the richer the amount of data to learn, the better fake images can be generated. Therefore, we do an additional experiment using classic augmentation and DCGAN at the same time to improve CNN performance. There are 8,960 training samples in *DS2*, 1,400 of which are images created by augmentation. In this experiment, we generate 1,400 fake images based on the *DS2* training set. And, we put these fake images on *DS0* and denote this dataset as DS3. The performance of the CNN model for *DS3* is 98.2% for 100 epochs.

DS4 is a dataset only for some classes with low accuracy. The fake image generation method follows *DS3* and add 600 map data samples into three classes (T2, T3, T6) with low accuracy in table 1. The dataset consists of 8,160 training samples and 1,000 test samples, and we denoted this dataset as *DS4*. Table 3 shows the number of data samples of each dataset. The performance of the CNN model for *DS4* is 98.3% for 100 epochs. Table 4 shows the accuracy of the CNN model with *DS4*. Table 5 shows the accuracy at the 100th epoch of each dataset.

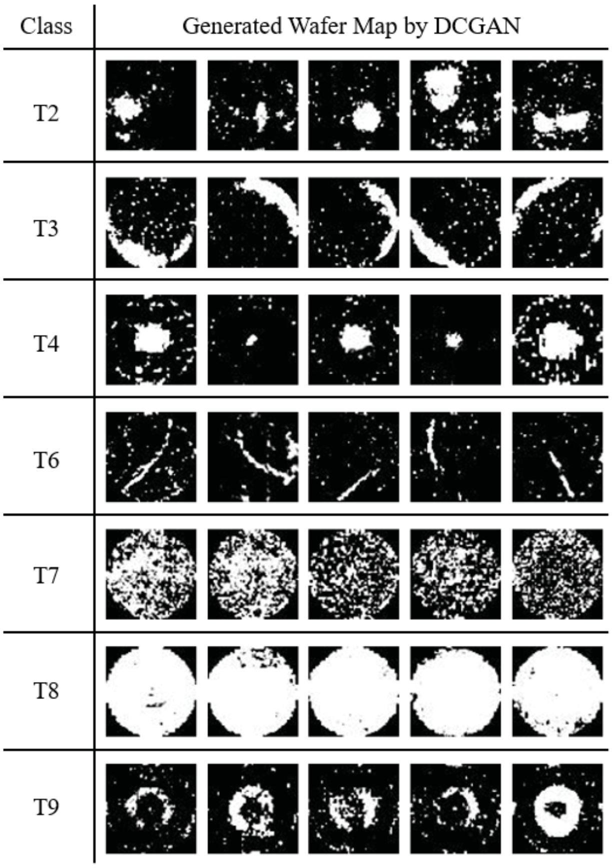


Fig. 4. Generated wafer defect map by DCGAN.

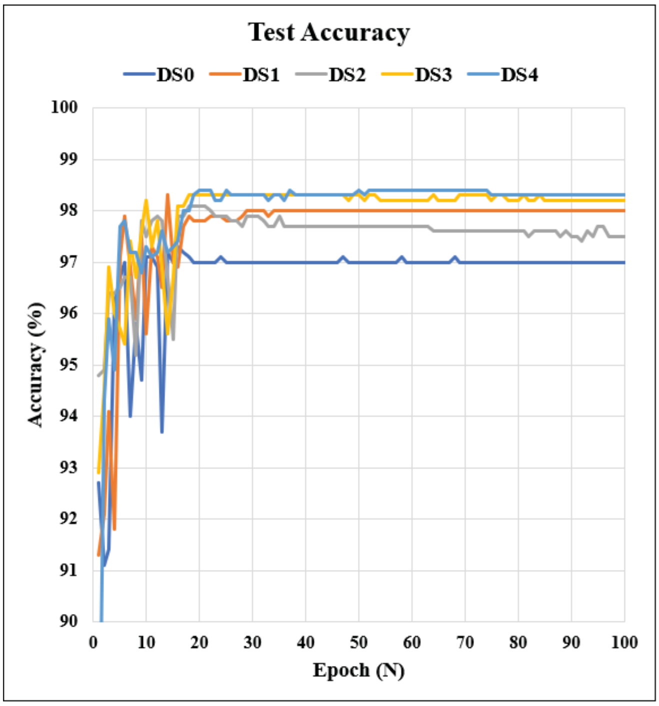


Fig. 5. Test accuracy for each dataset.

Table 3. The number of data samples in the training set.

Dataset	The # of data samples										
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Total
DS0	2,340	440	540	440	1,800	300	300	140	180	1,080	7,560
DS1	2,340	**640**	**740**	**640**	1,800	**500**	**500**	**340**	**380**	1,080	8,960
DS2	2,340	**640**	**740**	**640**	1,800	**500**	**500**	**340**	**380**	1,080	8,960
DS3	2,340	**640**	**740**	**640**	1,800	**500**	**500**	**340**	**380**	1,080	8,960
DS4	2,340	**640**	**740**	440	1,800	**500**	300	140	180	1,080	8,160

Table 4. Accuracy of confusion matrix for CNN model on DS4.

Predicted Actual	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Accuracy (%)
T1	260	0	0	0	0	0	0	0	0	0	100.0
T2	0	58	1	1	0	0	0	0	0	0	96.7
T3	1	2	56	0	0	1	0	0	0	0	93.3
T4	0	2	0	58	0	0	0	0	0	0	96.7
T5	0	0	1	0	199	0	0	0	0	0	99.5
T6	5	2	0	0	0	53	0	0	0	0	88.3
T7	0	0	0	0	0	0	60	0	0	0	100.0
T8	0	0	0	0	0	0	0	60	0	0	100.0
T9	0	0	0	1	0	0	0	0	59	0	98.3
T10	0	0	0	0	0	0	0	0	0	120	100.0
											98.3

Table 5. Accuracy of each dataset.

Dataset	The # of data samples		Accuracy
	Training set	Test set	
DS0	7,560	1,000	97.0%
DS1	8,960	1,000	98.0%
DS2	8,960	1,000	97.5%
DS3	8,960	1,000	98.2%
DS4	8,160	1,000	98.3%

IV. CONCLUSION

There are many steps to the fabrication process in the semiconductor industry, and the small differences in each step are closely related to the yield and quality of semiconductor manufacturing. An effective wafer map classifier can play an important role in maintaining high yields. Recently, many semiconductor industries are trying to utilize CNN architectures for high-performance wafer map classification. However, it is difficult to obtain data samples for some defect patterns; therefore, CNNs cannot be properly trained.

This paper proposed a new method to supplement the amount of dataset by generating wafer map images using GANs. We showed that the performance of a CNN-based wafer map classifier is significantly improved compared to the classic augmentation technique. Our proposed method can figure out the data imbalance/shortage problem in the CNN model. In conclusion, we can expect to increase the semiconductor yield using our method.

REFERENCE

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[2] Nakazawa, T., & Kulkarni, D. V. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network. IEEE Transactions on Semiconductor Manufacturing, 31(2), 309-314.

[3] Kyeong, K., & Kim, H. (2018). Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. IEEE Transactions on Semiconductor Manufacturing, 31(3), 395-402.

[4] Liu, C. W., & Chien, C. F. (2013). An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing. Engineering Applications of Artificial Intelligence, 26(5-6), 1479-1486.

[5] Baly, R., & Hajj, H. (2012). Wafer classification using support vector machines. IEEE Transactions on Semiconductor Manufacturing, 25(3), 373-383.

[6] Di Palma, F., De Nicolao, G., Miraglia, G., Pasquinetti, E., & Piccinini, F. (2005). Unsupervised spatial pattern classification of electrical-wafer-sorting maps in semiconductor manufacturing. Pattern Recognition Letters, 26(12), 1857-1865.

[7] Chang, C. Y., Li, C., Chang, J. W., & Jeng, M. (2009). An unsupervised neural network approach for automatic semiconductor wafer defect inspection. Expert Systems with Applications, 36(1), 950-958.

[8] Wang, J., Yang, Z., Zhang, J., Zhang, Q., & Chien, W. T. K. (2019). AdaBalGAN: An Improved Generative Adversarial Network with Imbalanced Learning for Wafer Defective Pattern Recognition. IEEE Transactions on Semiconductor Manufacturing, 32(3), 310-319.

[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[10] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.