Deep-Learning Based Classification Models for Wafer Defective Pattern Recognition

Jeong Cheol Seo
Department of Computer Science
Stanford University
jeongseo@stanford.edu

Abstract

Predictive multi-class models are trained for wafer defect map classification of 9 classes (including 'none' meaning no defect). Wafer map dataset (9 classes) is imbalanced so convolutional autoencoder and map rotating are used for data augmentation. The test results show that the rotating is the better way of data augmentation for wafer map data. Especially there are around 5% of noisy ground truth labeled data so 10K data for each class was generated for training by each augmentation method. VGG-16, ResNet50 and two types of Simplified VGG-16(with/without GAP layer) model architectures are trained on NVIDIA RTX 2080(8G) with Keras. For hyperparameter tuning, Grid search is used and the best parameters are picked for each model. VGG-16 and simplified VGG-16 model(with GAP) show the best result(f1-score macro average: 0.91, training accuracy: 0.99, test accuracy: 0.98). However, 'Scratch' class's f1 score is the lowest one among all classes in all test cases. Class Activation Map(CAM) data shows the reason and also indicates the similar shape with each classes' pattern.

1 Introduction

During semiconductor fabrication, integrated circuits (ICs) are made by creating circuit structures on many layers of a single wafer and interconnecting the structures using wires. To produce a high-density IC, the wafer surface must be extremely clean, and the circuit layers fabricated on the previous wafer should be aligned. The high-density structure may collapse if these conditions are not satisfied [1]. One important method of preventing collapse is to take scanning electron microscope (SEM) images of the wafer surface after the completion of each circuit layer and inspect whether there are any particles (e.g., rock and ring shapes), flaws (e.g., spots and scratches), or irregular connections caused by misaligned electronic circuits stacked on layers. By classifying wafer surface defects, manufacturers can determine the defect-introducing process steps and whether the wafer can be repaired and proceed to the next fabrication step[1]. Therefore, defect pattern recognition (DPR) and automatic pattern classification (ADC) is critical to improve yield and productivity. As the integrated circuits process becomes more and more complex with the technology scaling, improving the ability to recognize the defect patterns of the wafer maps is required [2]. Currently analytical method is mainly used but this is using predetermined pattern so this pattern need to keep updated after it is detected by human operator and every result should be reviewed by human operator again.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

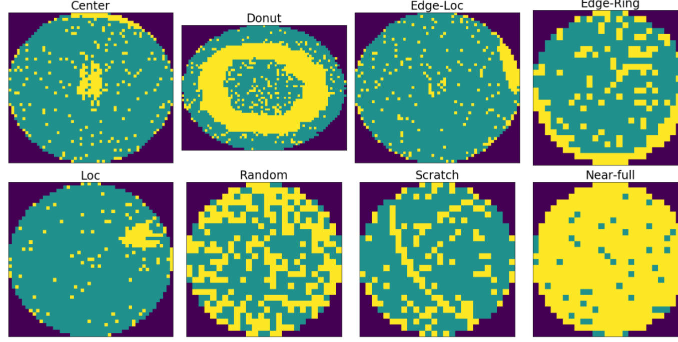

Figure 1: Wafer Defect map

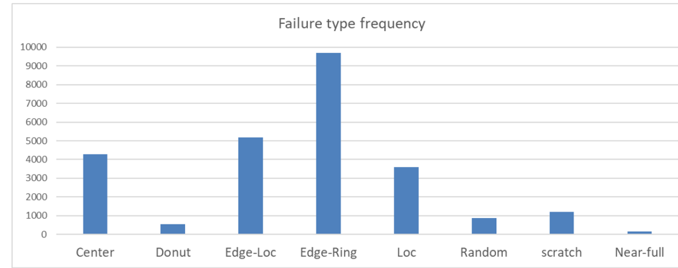

Figure 2: Failure Type Frequency

2 Related work

Recently there are some researches using machine learning based defects detection and classification methodologies [4]-[7]. However, Supervised learning based ML method demands a heavy computation burden because defect data are continually accumulated. ADC should use a proper data augmentation method that creates a realistic and unexpected defective patterns whenever a new class image is collected. Furthermore, Current defect hitting rate is around 80% so this should be improved to the level of around 95% for the practical applications [1]. To make this deep learning based classification model practically get applied to the production, this project will propose the better data augmentation method and best deep learning architecture which also can answer the following questions by showing class activation maps[8]. What part of the input is responsible for the output? And Can we check what the network focuses on given an input image? If class activation maps for each defect map has the same shape as its defect pattern then, this classification could be considered reasonable and give confidence.

3 Dataset and Features

Kaggle wafer map dataset[9] is used for learning and testing of the model. There are 811457 images in the data but only 172950 images with manual label (totally 9 labels: 0:Center, 1:Donut, 2:Edge-Loc, 3:Edge-Ring, 4:Loc, 5:Random, 6:Near-full, 7:Scratch, 8:none). From 172950 images, label 8 (none pattern, no defect) occupied 85.2%. There are 8 classes in defect maps and 'none' pattern class is no defect map. There are 9 classes in the dataset like the figure 1. However this data is heavily imbalanced by classes(Figure 2). The pattern (14.8% of labeled defect data) are 1)Center : 4294(2.5%), 2)Donut : 555(0.3%), 3)Edge-Loc : 5189(3.0%), 4)Edge-Ring : 9680(5.6%), 5)Loc : 3593(2.1%), 6)Random : 866(0.5%), 7)Scratch : 1193(0.7%), 8)Near-full : 149(0.1%). The issue on this dataset is that data is not balanced and there is noisy ground truth labeled data. So minimum 10K data is required for each class for training[10]. Convolutional autoencoder[11][12] and image rotation are used for data augmentation and the result will be compared to determine which is better way for the wafer map dataset. Training and Test data split is done by 80% : 20% ratio. Training is done using the two types of augmented dataset(10K per class) and Test is done using the originally distributed 20% imbalanced dataset for all test cases.

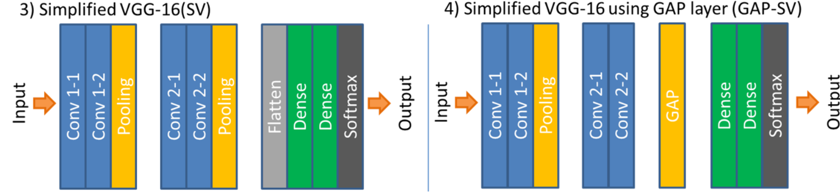

Figure 3: Simplified VGG-16 models (with/without Gloabl Average Pooling layer)

Initializer	Xavier_normal	Xavier_uniform	he_normal	he_uniform
F1 macro avg.	0.82	0.89	0.83	0.84

Table 1: Test Result for Initializer(optimizer:Adam)

4 Methods

The following four deep learning models are used for wafer map classification. 1)VGG-16, 2)ResNet-50, 3) Simplified VGG-16(SV) : initial 2 Conv. module and the final FC module, 4) Simplified VGG-16 using Global Average Pooling layer after the last Conv. layer (GAP-SV). The last two customized models are shown in Figure 3. Hyperparameter tuning is conducted by using grid search and the best parameters are picked for the two customized models(SV and GAP-SV). Xavier uniform is picked for initializer as shown in table 1 and learning rate(Lr), Regularizer L2 and dropout parameter are picked as shown in the Figure 4(learning rate: 0.001, dropout: 0.4, L2: 0.001). The value is F1 score macro average for all test results. Training accuracy is 0.99 and test accuracy is 0.97 for all cases.

5 Experiments/Results/Discussion

For training the four models, two types of training dataset are generated by two data augmentation method and after training is done by each data, test is conducted using the 20% original dataset for all cases. So Precision, Recall and F1 Score are reviewed for all test case, four models and two type of training dataset so totally 8 cases as shown in Figure 5. F1 Score macro average is used for final metric. And Training accuracy is same for all, 0.99. The detail results are shown in the figure 5. As shown in the result, the best F1 score is 0.91 when VGG16 and GAP-SV are used and two model have been trained using the rotating dataset. In the case of training is done by using the rotating dataset is showing the better F1 score than Autoencore's data. Precision, Recall and F1 Score for the best VGG16 model using rotating dataset is depicted in Figure 6. Furthermore, Scratch and Loc looks hard to predict as shown in Figure 5 and 6. Especially, Scratch's F1 score is lower than other class's for all cases. The CAM image in the Figure 7 shows clues for this[13][14], Scratch looks similar with 'Near-full' and 'none' and Loc looks similar with 'Edge-Ring'. The other CAM images looks similar with each classes defective map pattern and this means that the other classes classification works well and validated enough.

Dropout	L2(λ)	F1 score(macro avg.)	
		Lr=0.001	Lr=0.002
0.2	0.001	0.86	0.86
	0.01	0.84	0.77
	0.02	0.83	0.74
	0.03	0.81	0.79
0.3	0.001	0.86	0.84
	0.01	0.83	0.84
	0.02	0.67	0.82
	0.03	0.82	0.84
0.4	**0.001**	**0.89**	0.86
	0.01	0.81	0.85
	0.02	0.84	0.88
	0.03	0.64	0.79

Figure 4: Test Results for Dropout, Regualizer(L2) and Learning_rate (F1 score macro avg.)

Class	Data Augment: Conv. Autoencoder				Data Augment: Rotating			
	SV	GAP-SV	**VGG16**	ResNet50	SV	**GAP-SV**	**VGG16**	ResNet50
Center	0.98	0.97	**0.99**	0.96	0.99	**0.98**	**0.98**	0.99
Donut	1	0.67	**1**	0.67	1	**1**	**1**	1
Edge-Loc	0.82	0.84	**0.85**	0.78	0.85	**0.87**	**0.86**	0.85
Edge-Ring	0.83	0.74	**0.92**	0.91	0.85	**0.92**	**1**	0.83
Local	0.72	0.76	**0.74**	0.61	0.78	**0.8**	**0.78**	0.78
Near-full	0.79	0.87	**0.86**	0.87	0.92	**0.96**	**1**	1
Random	0.78	0.91	**0.86**	0.85	0.93	**0.92**	**0.83**	0.93
Scratch	0.46	0.76	**0.51**	0.29	0.68	**0.78**	**0.71**	0.64
none	0.99	0.99	**0.99**	0.98	0.99	**0.99**	**0.99**	0.99
Macro avg	0.82	0.83	**0.86**	0.77	0.89	**0.91**	**0.91**	0.89
Train Acc.	0.99	0.99	**0.99**	0.99	0.99	**0.99**	**0.99**	0.99
Test Acc.	0.97	0.97	**0.97**	0.97	0.97	**0.98**	**0.97**	0.97

Figure 5: Test Result : F1 Score by four models and two types of augmented dataset

	precision	recall	**f1-score**	# Data
Center	0.99	0.98	**0.98**	442
Donut	1	1	**1**	1
Edge-Loc	0.86	0.85	**0.86**	158
Edge-Ring	1	1	**1**	6
Loc	0.77	0.78	**0.78**	109
Near-full	1	1	**1**	11
Random	1	0.71	**0.83**	35
Scratch	0.89	0.59	**0.71**	27
none	0.98	0.99	**0.99**	2977
macro avg	0.94	0.88	**0.91**	
weighted avg	0.97	0.97	**0.97**	

Figure 6: Precision, Recall and F1 Score for VGG16 model using rotating dataset

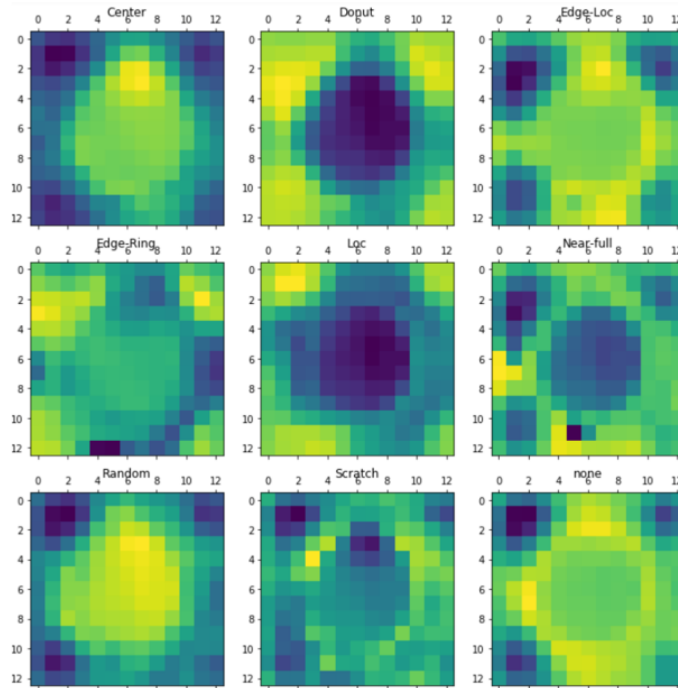

Figure 7: Class Activation Map images for all classes

6 Conclusion/Future Work

For the best deep-learning based classification model for wafer defective pattern recognition, VGG-16 is the best for both augmented dataset. However, Customized GAP-SV model is also showing the same best result in the case of using rotating dataset and there is one more benefit, the training time is half of VGG-16's. GAP-SV model also could be the useful classification model when this is applied to the production. For the training data augmentation method, Rotating is better than convolutional autoencoder for wafer map data for all test cases. It looks like rotating is generating more realistic, unexpected and regularized dataset. For future works, Improving 'Scratch' and 'Loc' pattern's prediction is required by generating an independent classification module for each class or by testing different network like Dense. To apply this to the production, a process flow and a certain model architecture for each process need to be defined more practically.

References

[1] Machine Learning Based Wafer Defect Detection, Yuansheng Mab, Feng Wanga, Qian Xiea, Le Hongb, Joerg Mellmannb, Yuyang Sunb, Shao Wen Gaoa, Sonal Singha, Panneerselvam Venkatachalama, James Wordb, *Proceeding of SPIE* on Oct. 7, 2019

[2] Classification of Mixed-Type Defect Patterns in Wafer Bin Maps Using Convolutional Neural Networks, Kiryong Kyeong and Heeyoung Kim, *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, VOL. 31, NO. 3, AUGUST 2018

[3] Convolutional Neural Network for Wafer Surface Defect Classiscation and the Detection of Unknown Defect Class, Sejune Cheon, Hankang Lee, Chang Ouk Kim, and Seok Hyung Lee, *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, VOL.32, NO.2, MAY 2019

[4] AdaBalGAN: An Improved Generative Adversarial Network With Imbalanced Learning for Wafer Defective Pattern Recognition, Junliang Wang , Zhengliang Yang , Jie Zhang, Qihua Zhang, and Wei-Ting Kary Chien, *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, VOL. 32, NO. 3, AUGUST 2019

[5] Deep Learning-Based Wafer-Map Failure Pattern Recognition Framework,Tsutomu Ishida1, Izumi Nitta1, Daisuke Fukuda1, Yuzi Kanazawa1, *20th Int'l Symposium on Quality Electronic Design*, 2019

[6] A Deep Learning Model for Identification of Defect Patterns in Semiconductor Wafer Map, Yang, Yuan-Fu, *ASMC* 2019

[7] Deep-Structured Machine Learning Model for the Recognition of Mixed-Defect Patterns in Semiconductor Fabrication Processes, Ghalia Tello, Omar Y. Al-Jarrah, Paul D. Yoo , Yousof Al-Hammadi, Sami Muhaidat, and Uihyoung Lee, *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, VOL. 31, NO. 2, MAY 2018

[8] Learning Deep Features for Discriminative Localization, Bolei Zhou et al. (2016)

[9] https://www.kaggle.com/qingyi/wm811k-wafer-map

[10] Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists, Rajpurkar Irvin et al., *PLOS Medicine*, 2018

[11] https://www.kaggle.com/kcs93023/keras-wafer-classification-cnn2d-with-augmentationData-augmentation

[12] https://towardsdatascience.com/aligning-hand-written-digits-with-convolutional-autoencoders-99128b83af8b

[13] Learning Deep Features for Discriminative Localization, Bolei Zhou et al., *In CVPR*, 2016

[14] https://github.com/jacobgil/keras-cam

[15] https://github.com/JeongSeo74/DeepLearning-CS230 (Source Code)