A Deep Learning Model for Identification of Defect Patterns in Semiconductor Wafer Map

Yang, Yuan-Fu
Dept. of Manufacturing Excellence
Taiwan Semiconductor Manufacturing Company, Limited
Republic of China, Taiwan
yfyangd@tsmc.com

Abstract

The semiconductors are used as various precision components in many electronic products. Each layer must be inspected of defect after drawing and baking the mask pattern in wafer fabrication. Unfortunately, the defects come from various variations during the semiconductor manufacturing and cause massive losses to the companies' yield. If the defects could be identified and classified correctly, then the root of the fabrication problem can be recognized and eventually resolved.

Automatic optical inspection (AOI) is used to visualize defect patterns and identify root causes of die failures. AOI can be replaced a large number of human inspections with high-speed and accurate inspection technology, to achieve consistency in the detection and shorten the inspection time, then improve product quality and competitiveness. The defect is judged from the feature in AOI, but the final goal is to determine if the defect is a true or a pseudo defect of the wafer. Then, we need to determine what defect type is. But the current AOI needs a subsequent final verification by the human to judge the type of defect.

Machine learning (ML) techniques have been widely accepted and are well suited for such classification and identification problems. In this paper, we employ convolutional neural networks (CNN) and extreme gradient boosting (XGBoost) for wafer map retrieval tasks and the defect pattern classification. CNN is the most famous deep learning architecture. The recent surge of interest in CNN is due to the immense popularity and effectiveness of convnets. XGBoost is the most popular machine learning framework among data science practitioners, especially on Kaggle, which is a platform for data prediction competitions where researchers post their data and statisticians and data miners compete to produce the best models. CNN and XGBoost are compared with a random decision forests (RF), support vector machine (SVM), adaptive boosting (Adaboost), and the final results indicate a superior classification performance of the proposed method.

Our experimental result demonstrates the success of CNN and extreme gradient boosting techniques for the identification of defect patterns in semiconductor wafers. The overall classification accuracy for the test dataset of CNN and extreme gradient boosting is 99.2%/98.1%. We demonstrate the success of this technique for the identification of defect patterns in semiconductor wafers. We believe this is the first time accurate computational classification in such task has been reported achieving accuracy above 99%.

I. INTRODUCTION

A wafer is an elementary unit in semiconductor manufacturing. Several hundred or thousand integrated circuits (ICs) are simultaneously fabricated on a single wafer [1]. There are hundreds of steps in the fabrication process before finalizing the design and approving its functionality [6]. For each steps, wafer need to be classified as either functional or defective. Wafer maps are used to visualize the locations of defective ICs chips on the wafer and identify potential process issues [2]. Inline metrology tools perform inspection after a certain process step and monitor abnormalities on dies. Then a wafer map is created based on the detected abnormal locations [3]. Defective chips commonly occur in clusters or display some systematic patterns. Such defect patterns contain useful information about manufacturing process conditions [4].

These defects can be a result of imperfections in the machines used or due to the chemical dyes used, physical damages, human mistakes, etc. For example, uneven temperatures or chemical aging lead to spatial cluster on the wafer map. Clusters also can be the result of crystalline non-uniformity, photo-mask misalignment, or particles caused by mechanical vibration [2]. Material shipping and handling also can leave a scratch on the wafer map [5]. It is very challenging to produce a defect-free wafer lot. If we were able to identify such defect patterns correctly, the probability of fixing the root cause of the problem would be considerably high [6].

In the majority of semiconductor manufacturing processes, visual inspection of defects depends on manual review by human experts [12]. The inspectors randomly select samples from the total wafer population and try to find the root causes of defects by examining their locations, sizes, colors, and shapes using a high-resolution microscope. This human inspection procedure, however, is time consuming and highly subjective [7]. A previous study has shown agreement between observers to be as low as 45%, with long-term repeatability values of less than 95% [13]. Conversely, Automatic optical inspection can quickly examine a wafer using laser light. It can identify the locations of the defects and their relative sizes based on the scattering of the light. Automatic methods, capable of quickly assessing the root causes of the defects by analyzing the data obtained from AOI, are of great importance to semiconductor manufacturing because these automatic methods can lead to a considerable reduction in operator workload and improvements in accuracy and consistency [6].

There have been numerous studies on the automatic defect detection and feature extraction in semiconductor manufacturing. Chen and Liu [7] utilized neural networks for spatial defect pattern recognition. Gleason et al. [8] employed an automated clustering algorithm via artificial intelligence. Cunningham and Mackinnon [9] used an empirical clustering algorithm. Hsieh and Chen [10] developed an analytical structure made up of a fuzzy rule-based inference system to help identify defect spatial patterns. Liu [11] developed intelligent systems that use wafer maps and wafer bin maps, respectively, to recognize defect spatial patterns and aid in the diagnosis of causes of failures. They adapted a neural network called an adaptive resonance theory network 1 (ART1) for this purpose.

Principal component analysis (PCA) could be one of effective feature extraction techniques, which is often used for pattern recognition [16]. PCA has the ability to discriminate directions with the largest variance in a dataset, and to extract several representative features (i.e., principal components). As a kind of linear supervised feature extraction technique, linear discriminant data analysis (LDA) has been used to isolate faults from normal operation and to optimize their separability [17]. LDA is achieved by maximizing the between-class scatter while minimizing the within-class scatter. LDA can obtain better feature extraction performance than that of PCA due to the utilization of class label information. These feature extraction methods have been applied in manufacturing process monitoring, fault diagnosis, etc. [18]. Singular value decomposition (SVD) is often used for image feature extraction, especially in failure pattern in wafer [14][15]. It has been successfully used to feature extraction in various area, such as physiological signals, motor faults classification. SVD is a nonlinear filtering with a small phase shift and no-time delay, which is widely applied in the de-noising of the vibration signal. In this study, our datasets provides a massive amount of information, but much of the insight might be redundant or useless (noise). Thus, we used SVD to recognize the most informative features of lower dimensional data.

Once the features are extracted, the common pattern classification algorithms such as support vector machines (SVM), random decision forests (RF), adaptive boosting (Adaboost) etc. are applied for the classification task. SVM seemed to be the most widely used algorithms to encounter the defect classification problem due to their powerful performance and versatility [16][17][18]. SVM models can be easily manipulated, and they have the ability to deal with multiclass data classification problems, multimodal data and with datasets consisting of inseparable points [16]. The goal behind it is to design a hyper-plane that classifies all training vectors into different classes. Many planes can be suitable for the same problem. In such cases the one that leaves maximum margin between the classes is chosen [17]. Kwon and Kang [19] proposed a defect detection method that can find the surface irregularity of the variety surface with Random Forest using.

Kim, Lee, and Cho [20] used Adaboost to propose an automatic defect classification algorithm. Boosting is to make a strong classifier by combining multiple weak classifiers. The weak classifier just has a bit better performance than random decision. An important task in the boosting scheme is how to combine the weak classifiers into a strong one. Adaboost generalizes the process of many boosting algorithms [21], [22]. The Adaboost consists of two elements, one is how to combine the weak classifiers, and the other is how to update the weights of the training data.

Extreme gradient boosting also called XGBoost is a kind of boosting algorithm that has received rave from the machine learning practitioners. XGBoost is the short name for “Extreme Gradient Boosting” proposed by Friedman in the Greedy Function Approximation: gradient boosting machine journal of the term “Gradient Boosting”. XGBoost is based on this original model [23]. Boosting classifier is a common integrated learning model. Many tree models with low accuracy are combined according to its classification principle. Finally, it becomes a classifier with high accuracy. A new tree can be formed in each of iteration. However, long calculation time is required in the method aiming at complex large dataset. Chen Tianji [24] improved gradient boosting machine aiming at the problem. The XGBoost version based on CPU parallelism is realized. Speed and accuracy are greatly improved. Currently, CPU acceleration version has been realized [25].

Convolutional neural networks (CNN) [26] have advanced new classification performance recently and became the standard approach for image classification tasks. CNN is the end-to-end model and does not require any feature engineering [3]. This end-to-end model approach is beneficial since we don’t need to develop the task specific feature extractors and the domain specific export knowledge is not required. A lot of approach for defect detection is to apply CNN. Borisov and Scheible [27] proposed a fast and accurate solution based on novel artificial neural network (ANN) architecture for precise lithography hotspot detection using a convolution neural network (CNN) adopting a state-of-the-art technique. Marco Maggipinto, Matteo Terzi, and Chiara Masiero [28] exploited modern deep learning (DL)-based technologies that are able to automatically extract highly informative features from the data, providing more accurate and scalable virtual metrology (VM) solutions. The proposed methodology is tested on a real industrial dataset related to etching, one of the most important semiconductor manufacturing processes. The dataset at hand contains optical emission spectroscopy data and it is paradigmatic of the feature extraction problem in VM under defect examination. Max, Ronay, and Lee [29] developed CNN model in the automatic localization of defects in metal castings. In this study, CNN had shown outstanding performance in both image classification and localization tasks. Nakazawa and V. Kulkarni [3] employed CNN for the defect pattern classification and wafer map retrieval tasks. They demonstrated that by using only synthetic data for network training, real wafer maps can be classified with high accuracy.

In this paper, we employ convolutional neural networks and extreme gradient boosting (XGBoost) to achieve an accurate and efficient classification of wafer defect patterns. As a dataset, we use wafer maps from the real wafers of TSMC 300mm fab. The paper is organized as follows. Section II describes our proposed method including data preprocess, feature extraction, data augmentation and model turning. Section III discusses the model validation method and testing results. Section IV concludes this paper.

II. METHOD

This section introduces the method of this research. First, we explored the data of defect image. Second, we reduced the noise of image by data processing. In the initial model, we used random forest for defect classification performance benchmark. Then, feature extraction was performed. Traditionally, the image classification requires feature extraction using object color and shapes. In CNN, we don’t need to develop the task specific feature extractors since the deep CNN can learn rich features at each layer, these intermediate features are used as good descriptors for image retrieval.

However, in the non-CNN model, feature extraction is necessary. It can increase the recognition of the image. In this study, we used singular value decomposition (SVD) to extract feature for non-CNN model (SVM, AdaBoost, XGBoost). Then, we used GridSearchCV to find the optimization parameters in each non-CNN model. Finally, data augmentation is performed to improve the over fitting problem.

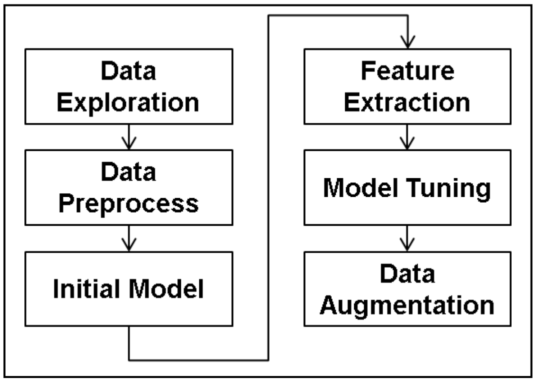


Fig. 1. Work flow of defect learning

A. Data Exploration

Data exploration is used to understand, summarize and analysis the contents of a dataset, usually to investigate a specific question or to prepare for more advanced modeling. In predictive modeling, "signal" is the true underlying pattern that we wish to learn from data. On the other hand, "noise" refers to the irrelevant information or randomness in a dataset [30].

In this study, there are a total of 6 defect types, include of particle, residual, over mapping, discolor, rubbing, poor coating. Nosie exists in each defect pattern. Figure II illustrate that the typical defect signature of wafer in high resolution image. The data look like pretty messier and it appears that the background is not really random noise but rather has some spatial correlations. Some of these blobs in the picture are not that high above noise, so it may be advantageous to first transform the images in some way to enhance the contrast between the signals and the background.

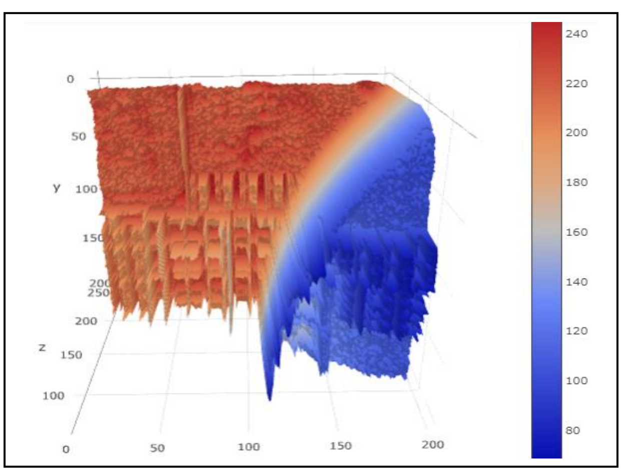


Fig. 2. Data exploration of wafer map

B. Data Preprocess

Data preprocessing is the most important phase in machine learning projects. Data collection methods are usually loosely controlled, resulting in values that are out of range. Therefore, the representation and quality of data is first and foremost before analysis. Knowledge discovery during the training phase is more difficult if there are many uncorrelated and redundant information or noise and unreliable data. Data preparation and filtering steps can take a considerable amount of processing time. Data preprocess includes cleaning, instance selection, normalization, transformation, feature extraction and selection. The product of data preprocessing is the final training set. In this study, we use Min-Max scaling to normalize the data. A Min-Max scaling will scale the data between the 0 and 1. It is typically done via the following equation:

$$X_{min\text{-}max} = (X - X_{min})/(X_{max} - X_{min}) \tag{1}$$

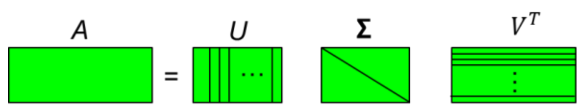
In this approach, the data is scaled to a fixed range (typically 0 to 1). We end up with a smaller standard deviation, which can suppress the effects of outliers.

C. Feature Extraction

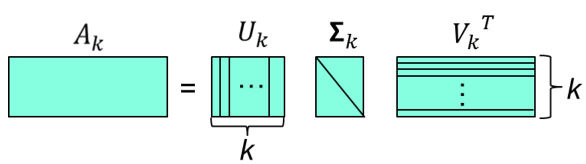
Feature extraction is the process of transforming the input data into a set of features which can very well represent the input data. It is a process of deriving new features from the original features in order to reduce the cost of feature measurement, increase classifier efficiency, and allow higher classification accuracy. Feature extraction presents raw image in a reduced form to facilitate decision making such as pattern detection, classification or recognition. If the features extracted are carefully chosen, it is expected that the features set will perform the desired task using the reduced representation instead of the full size input.

Singular Value Decomposition (SVD) has been successfully used to feature extraction in various area, such as physiological signals, motor faults classification. By SVD, we can extract orthogonal matrix (U), diagonal matrix (s), unitary matrix (V) from original image:

$$A = U\,\Sigma\,V^T \tag{2}$$



$$A_k = U_k\,\Sigma_k\,V_K^T \tag{3}$$



An annotation prediction is performed by computing a reduced rank approximation Ak of the annotation matrix (where $0 < k < r$, with r the number of non-zero singular values of A, i.e. the rank of A). The picture of the left one is the original image. We can see that there are many noises around the target. After feature extraction by SVD (the picture of the right one), the target (defect) is much clear and the noise waves has almost been removed.

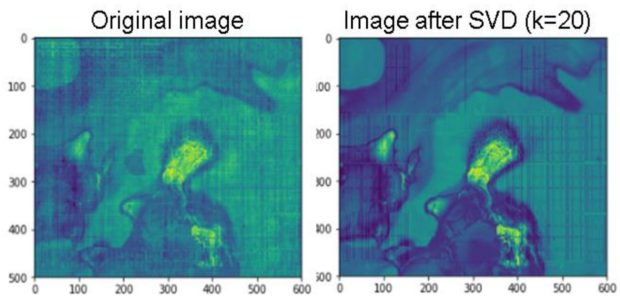


Fig. 3. Wafer map after feature extraction by SVD

D. Data Augmentation

In deep learning, we often need a lot of data to ensure that there is no over-fitting during training. However, in today's digital era, data is the new oil that drives the A.I. engine to work. If the company is thirsty for information, most of the valuable information will be in the well-funded companies or companies in related fields. It is difficult for individual developers or ordinary companies to own or collect the complete required materials. Therefore, we generally take data augmentation to make up for the over-fitting problem caused by insufficient data.

Data augmentation has proven to solve the problem of insufficient data and improve the accuracy of system training [31]. It is the process of generating samples by transforming training data, with the target of improving the accuracy and robustness of classifiers. In this paper, we added more image by data augmentation, including random cropping, , rotating, resizing the picture. After this work, our eyes can still recognize it as the same picture, but it is a completely different new image for the machine. Therefore, data augmentation is to modify the existing image in the dataset to create more images for machine learning and make up for insufficient data.

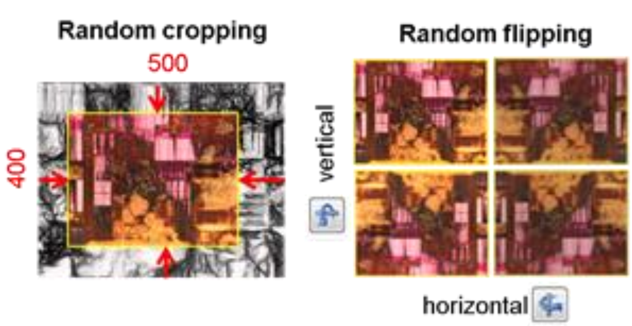


Fig. 4. The method of data augmentaion

E. Hyperparameter optimization

Hyperparameter optimization is one of the cornerstones of building successful machine learning models, and it's crucial for the success training of any model. Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined loss function on given independent data. The objective function takes a tuple of hyperparameters and returns the associated loss. Cross-validation is often used to estimate this generalization performance. The traditional way of performing hyperparameter optimization has been Grid Search, or a parameter sweep, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm.

A Grid Search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set or evaluation on a held-out validation set. In this study, a typical XGBoost classifier has at least three hyperparameters that need to be tuned for good performance on unseen data: learning rate, max_depth, and gamma. To perform grid search, one selects a finite set of reasonable values for each.

TABLE I. HYPERPARAMETER TUNING

Classifier	Parameter	Range
SVM	C_range	0.1, 1, 10, 50, 100
	Gamma	0, 5, 10, 20, 50
AdaBoost	n_estimators	10, 50, 100, 200, 1000
	learning_rate	0.2, 0.4, 0.6, 0.8, 1.0
XGBoost	learning_rate	0.2, 0.4, 0.6, 0.8, 1.0
	Max_depth	6, 10, 20, 50, 100
	gamma	0, 5, 10, 20, 50

As below picture is a heatmap of the XGBoost's cross-validation accuracy score as a function of learning rate and gamma. For this example we explore a relatively grid for illustration purposes. The behavior of the model is very sensitive to the parameter of max_depth (the maximum depth of a tree). If max_depth is too large, it will make the model more complex and likely to be overfitting. Limit of max_depth is required for depth wise grow policy.

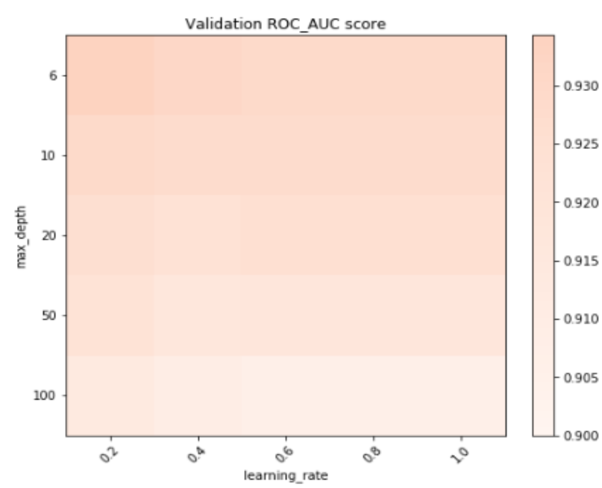


Fig. 5. Validation score of Grid Search in XGBoost

F. Convolutiona neural network configuration

Table II shows our CNN configuration. The input wafer map image size is 500 X 600. We have three convolutional layers. In the first convolutional layer, the receptive field size is 3 X 3 and stride is 1. In the second and third convolutional layer, the receptive field size is 2 X 2 and stride is 1. The first convolutional layer have 32 channels, the second and third convolutional layer has 64 channels. The rectified linear unit (ReLU) activation is used for each convolutional layer. The max pooling size is 3 X 3. One dense layer with 128 channels is added after convolution/pooling layers. The fully connected (FC) layer with channels 128 is added with sigmoid activation. After dropout, another fully connected layer with channels 6 (number of defect classes) is added. The last layer is the softmax layer for the class probability calculation.

TABLE II. CNN CONFIGURATION

Layer	Output Shape
32 3 X 3 2D Convolutional layer	498 X 598 X 32
Rectified Linear Unit activation (ReLU)	
64 2 X 2 2D Convolutional layer	497 X 597 X 64
Rectified Linear Unit activation (ReLU)	
64 3 X 3 Max pooling layer	165 X 199 X 64
Dropout (25%)	
64 2 X 2 2D Convolutional layer	164 X 198 X 64
Rectified Linear Unit activation (ReLU)	
64 3 X 3 Max pooling layer	54 X 66 X 64
Dropout (25%)	
Flatten layer	228096
Dense layer	128
Sigmoid activation	
Dropout (25%)	
Fully connected layer	6
Softmax	

III. RESULT

We train our models as follows. First, we split the images randomly into 9384 (80%) training data set and 2346 (20%) test data set. The training data set is used for training our models, and the testing data set is used for the validation. In CNN model, we split 25% from training data set for validation in each epoch. The training accuracy after the 20 epoch is 98.8% and the validation accuracy is 97.9%. The training and validation accuracy for each epoch is shown in Fig. 6. The average processing time for each epoch is 231.6 seconds.

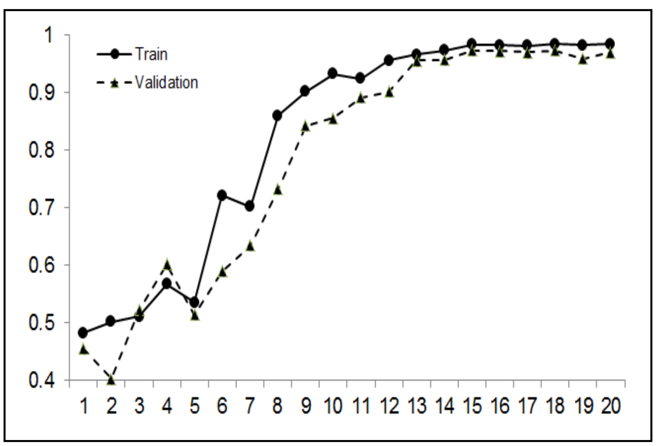


Fig. 6. The training and validation accuracy of CNN model.

Table III is the confusion matrix and it shows the per-class classification accuracy in percentage. Due to the confidentiality reason, we can only provide per-class accuracy, not the absolute number of wafers. Most of the class accuracy is greater than 98%. The overall accuracy is 99.2%.

TABLE III. THE TRAINING AND VALIDATION ACCURACY CONFUSION MATRIX IN PERCENTAGE FOR 6 DEFECT

Type	PAR	RE	OM	DC	Ru	PC
PAR	99.7	0.2	0.9	0.0	0.0	0.3
RE	0.0	99.5	0.0	0.2	0.5	0.0
OM	0.1	0.0	98.3	0.0	0.0	0.0
DC	0.0	0.0	0.4	98.3	1.0	0.3
Ru	0.1	0.3	0.0	1.5	98.5	0.0
PC	0.0	0.3	0.4	0.0	0.0	99.4

Compare with 4 methodologies, the CNN model is the best classifier in the defect classification problem, even that the accuracy of CNN before data augmentation (0.893) is good than SVM/AdaBoost after data augmentation (0.804/0.817). The final test show that the accuracy of CNN is 0.992, it is higher than what we expected (as table IV).

TABLE IV. FINAL RESULT

Model	Benchmark Model	Feature Extraction	Model Tuning	Data Augmentation
SVM	0.532	0.685	0.722	0.804
AdaBoost	0.532	0.706	0.738	0.817
XGBoost	0.532	0.819	0.937	0.981
CNN	0.532	0.893	-	0.992

IV. CONCLUTION

In this paper, we present a method for wafer map classification using different machine learning model. We successfully used data extraction & hyperparameter tuning to improve the 3 supervised learning (SVM/AdaBoost/XGBoost) performances. We also demonstrate efficiency and performance of CNN based image retrieval using the binary code generated by the FC layer of our CNN model. In this study, the CNN is still a more applicable method for image classification problems. The reason is that the CNN is able to learn relevant features from the input image, resulting in a better generalized model, which is better than other machine learning methods.

Although CNNs have achieved great success in experimental evaluations, there are still lots of issues that deserve further investigation. Firstly, since the recent CNNs are becoming deeper and deeper, they require large-scale dataset and massive computing power for training. Manually collecting labeled dataset requires huge amounts of human efforts. Thus, it is the next topic that how to explore unsupervised learning of CNNs. Furthermore, one major barrier for applying supervised learning on a new task is that it requires considerable skill and experience to select suitable hyper-parameters such as the learning rate, gamma, etc. These hyper-parameters have internal dependencies which make them particularly expensive for tuning. It is also a challenge of next study.

REFERENCES

[1] J. S. Fenner, M. K. Jeong, J. C. Lu, "Optimal automatic control of multistage production processes", IEEE Trans. Semicond. Manuf., vol. 18, no. 1, pp. 94-103, Feb. 2005.
[2] Y.-S. Jeong, S.-J. Kim, 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.

[3] Takeshi Nakazawa, Deepak V. Kulkarni, "Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network", IEEE Trans. Semicond. Manuf., vol. 31, no. 2, pp. 309-314, Jan. 2018.
[4] S. P. Cunningham, S. McKinnon, "Statistical methods for visual defect metrology", IEEE Trans. Semicond. Manuf., vol. 11, no. 1, pp. 48-53, Feb. 1998.
[5] C. K. Hansen, P. Thyregod, "Use of wafer maps in integrated circuit manufacturing", Microelectronics Reliability, vol. 38, pp. 1155-1164, 1998.
[6] 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.
[7] F. L. Chen, S. F. Liu, "A neural-network approach to recognize defect spatial pattern in semiconductor fabrication", IEEE Trans. Semiconduct. Manuf., vol. 13, no. 3, pp. 366-373, Aug. 2000.
[8] S. S. Gleason, K. W. Tobin, T. P. Karnowski, F. Lakhani, "Rapid yield learning through optical defect and electrical test analysis", Proc. SPIE-Int. Soc. Opt. Eng., vol. 3332, pp. 232-242, 1998.
[9] S. Cunningham, S. MacKinnon, "Statistical methods for visual defect metrology", IEEE Trans. Semiconduct. Manuf., vol. 11, no. 1, pp. 48-53, Feb. 1998.
[10] H. W. Hsieh, F. L. Chen, "Recognition of defect spatial patterns in semiconductor fabrication", lnt. J. Production Res., vol. 42, no. 19, pp. 4153-4172, 2004.
[11] S. F. Liu, F. L. Chen, W. B. Lu, "Wafer bin map recognition using a neural network approach", Int. J. Production Res., vol. 40, no. 10, pp. 2207-2223, 2002.
[12] N. G. Shankar, Z. W. Zhong, "Defect detection on semiconductor wafer surfaces", Microelect. Eng., vol. 77, no. 3, pp. 337-346, 2005.
[13] A. Freeman, M. McIntyre, M. Retersdorf, C. Wooten, X. Song, A. Hesse, "The application and use of an automated spatial pattern recognition (SPR) system in the identification and solving of yield issues in semiconductor manufacturing", Proc. IEEE/SEMI Adv. Semiconduct. Manuf. Conf., pp. 302-305, 2007-Jun.
[14] Mohamed Baker Alawieh, Fa Wang, Xin Li, "Identifying systematic spatial failure patterns through wafer clustering", IEEE International Symposium on Circuits and Systems, pp. 22-25, Aug. 2016.
[15] Mohamed Baker Alawieh, Fa Wang, Xin Li, "Identifying Wafer-Level Systematic Failure Patterns via Unsupervised Learning", IEEE Trans. Semicond. Manuf., vol. 37, no. 4, pp. 832–844, July. 2017.
[16] R. Baly and H. Hajj, “Wafer classification using support vector machines,” IEEE Trans. Semicond. Manuf., vol. 25, no. 3, pp. 373–383, Aug. 2012.
[17] W. Zeng, J. Jia, Z. Zheng, C. Xie, and L. Guo, “A comparison study: Support vector machines for binary classification in machine learning,” in Proc. 4th IEEE Int. Conf. Biomed. Eng. Informat. (BMEI), vol. 3. Shanghai, China, 2011, pp. 1621–1625.
[18] L. Xie, R. Huang, and Z. Cao, “Detection and classification of defect patterns in optical inspection using support vector machines,” in Intelligent Computing Theories. Berlin, Germany: Springer, 2013, pp. 376–384.
[19] Bae- guen Kwon, Dong-joong Kang, “Fast defect detection algorithm on the variety surface with random forest using GPUs”International Conference on Control, Automation and Systems, Oct. 2011, pp. 26–29.
[20] Hong Il Kim, Sang Hwa Lee, Nam Ik Cho “Automatic Defect Classification Using Frequency and Spatial Features in a Boosting Scheme”IEEE Signal Processing Letters, Mar. 2009, pp. 374–377.
[21] R. E. Schapire, "The boosting approach to machine learning: An overview", Proc. MSRI Workshop on Nonlinear Estimation and Classification, 2002.
[22] R. E. Schapire, Y. Singer, "Improved boosting algorithms using confidence-rated predictions", Proc. Computational Learning Theory, pp. 80-91, 1998.
[23] T. Chen, T. He, Xgboost: extreme gradient boosting. R package version, pp. 4-2, 2015.
[24] Tianqi Chen, Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016.
[25] Rory Mitchell, Eibe Frank, Accelerating the XGBoost algorithm using GPU computing. PeerJ Preprints, no. 1, 2017.
[26] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Proc. Adv. Nueral Inf. Process. Syst., pp. 1097-1105, 2012.
[27] Vadim Borisov, Jürgen Scheible, "Lithography Hotspots Detection Using Deep Learning", 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Aug. 2018.
[28] Marco Maggipinto, Matteo Terzi, Chiara Masiero, "A Computer Vision-Inspired Deep Learning Architecture for Virtual Metrology Modeling With 2-Dimensional Data", IEEE Trans. Semicond. Manuf., vol. 31, no. 3, pp. 376-384, Jun. 2018.
[29] Max Ferguson, Ronay Ak, and Yung-Tsun Tina Lee, "Automatic localization of casting defects with convolutional neural networks", 2017 IEEE International Conference on Big Data (Big Data), Jan. 2018.
[30] Nate Silver, "The Signal and the Noise", 2012.
[31] Marco Maggipinto, Matteo Terzi, Chiara Masiero, "Adaptive data augmentation for image classification", 2016 IEEE International Conference on Image Processing (ICIP), Aug. 2016.