Bayesian Optimized Deep Neural Network for Nuisance Event Filter

(Computer Vision)

Yuping He

yphe8@stanford.edu

Abstract

As semiconductor process shrinks, and chips become denser, wafer inspection becomes more and more challenging, and the conventional inspection methods may not be good enough to separate the defect of interest from nuisance. In this report, we demonstrate that deep learning-based defect inspection could be particularly effective. In addition, the combination of deep learning and Bayesian optimization techniques can further improve the efficiency of wafer inspection system.

1 Introduction

Wafer inspection is a science of finding defects on a wafer. In the inspection process, a wafer inspection tool takes photos of dies in a series of time, and then compares them. If there's a change among them, that's generally a defect. The goal is to find a defect of interest (DOI) on wafer, but the inspection system may also detect what is commonly called a nuisance (NUI), which is a noise or false defect. As semiconductor process shrinks, and chips become denser, wafer inspection becomes more challenging due to weaker defect signal and higher nuisance rates. Deep learning technique has recently deployed to improve the efficiency of wafer inspection system. However, Deep learning requires a lot of experience to tune the large number of hyper-parameters. Such manual tuning process is likely to be biased. Bayesian optimization [1] is one of powerful techniques for the search of machine learning hyperparameters. It attempts to find the optimal set of model parameters in a minimum number of steps comparing to grid search and random search [2]. Bayesian optimization incorporates a prior belief about f, and updates the prior with samples drawn from f to get a posterior that better approximates f, i.e. Gaussian processes (GP)[3]. In addition, it employs an acquisition function, i.e. expected improvement (EI), to propose a better next sampling point in the search space.

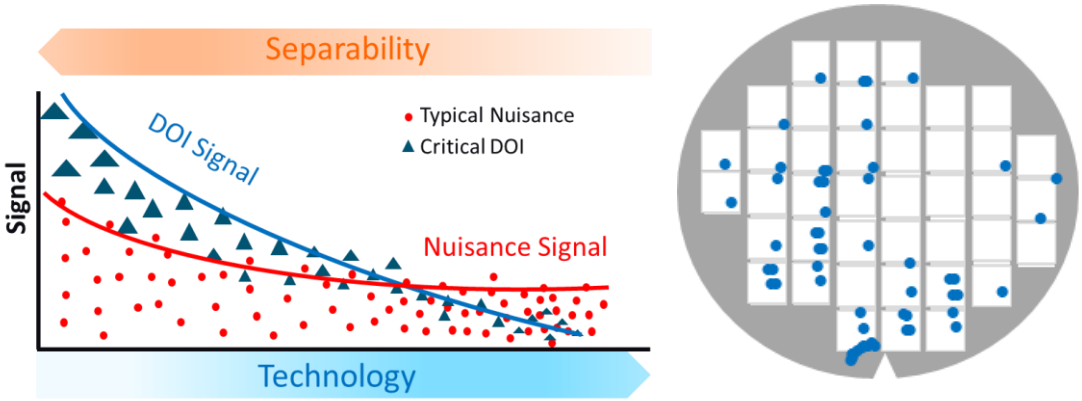


Figure 1. Separability of DOI and Nuisance with respect to semiconductor process technology

2 Dataset

The human-labeled defect images are used in this project. There is a total of 3409 defect images. which were collected from two wafers. In order to avoid any data mismatch problem, the images were taken by using the same inspection tool and same inspection receipt, and two wafers are the same layer under the same semiconductor process. All collected defect images were then studied and labeled by human experts into two groups: defect of interested (DOI) and nuisance (NUI). There are 271 DOI and 3138 NUI among 3409 defect images, and each is a 32x32 grayscale image. The total 3409 images were then randomly shuffled and split into 50% and 50% for training and validation sets using stratified sampling method. The stratified sampling method is used to result the same fraction of DOI and NUI for both training and validation sets. The reason of using 50% splitting is because the model measure metric is the DOI and NUI count ROC due to the specific goal of defect inspection in our case. Since we are interested in the comparison of model performance between with and without Bayesian optimization, we don't need to create the test set. The criterion for being a best model is to capture DOI as many as possible with less NUI. In the metric of defect count ROC, the X-axis and Y-axis represent the number count of true NUI and the number count of true DOI in the total number of predicted DOI.

3 Approach

a. Neural network architecture

Since the defect image is a 32x32x1 grayscale image, a new neural network architecture is built based on LeNet-5 [4], which is a widely known CNN architecture for predicting written digits recognition (MNIST) with 32x32x1 images in grayscale. A modified

LeNet-5 architecture is shown in Figure 2. Different from the original LeNet-5, we use the filter of 3x3 for both CONV1 and CONV2 and use "same" padding (p=1) instead of "valid" padding (p=0). We also use maximum pooling to replace average pooling. To solve the overfitting problem, we adopt several new techniques, such as L2 norm, dropout for fully connected layers and batch normalization for convolutional layers. In addition, we find that it is critical to add a new layer of spatial pyramid pooling (SPP) [5] between the convolutional layers and the fully connected layers due to the shifted defect position in the defect images. A softmax function is used at the last layer, and it results in the probabilities of predicted DOI and NUI. Therefore, the network architecture in Figure 2 performs a binary classification.

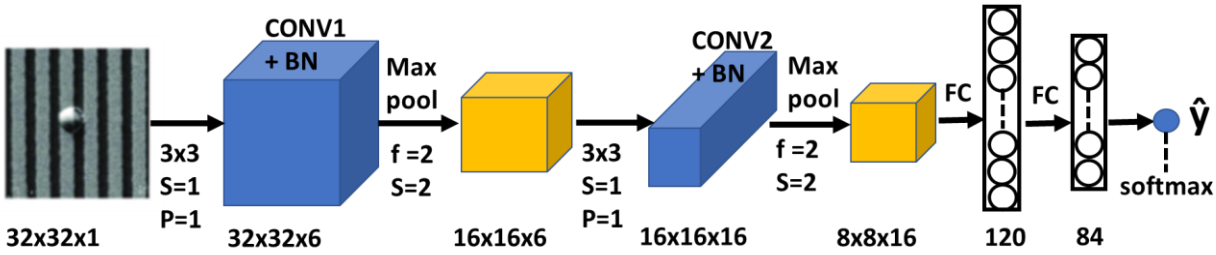


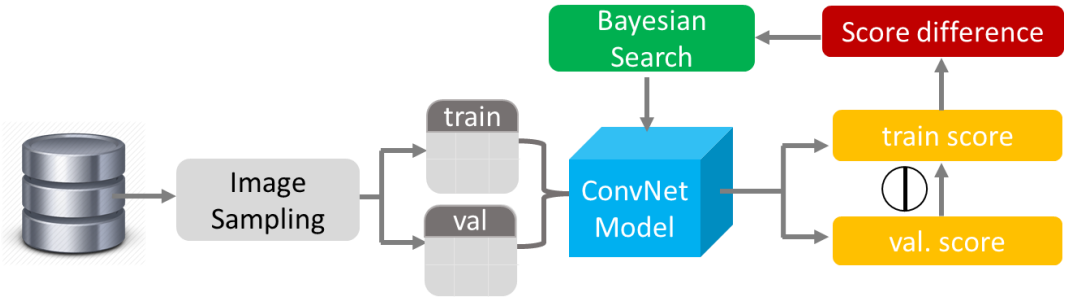
Figure 2. CNN architecture

b. Bayesian hyper-parameter search

The Bayesian optimization technique is integrated with the network architecture in Figure 2. Figure 3a shows the workflow of the integrated prototype. Figure 3b shows the algorithm process of Bayesian optimization for hyperparameter search. We use the difference of F1 score between train and validation as an objective function, Gaussian Process (GP) as surrogate model, and Expected Improvement (EI) as acquisition function. For deep neural network, there are two types of hyperparameters which need to be tuned: architecture parameters and learning parameters. In this work, we focus on searching learning parameters, such as, learning rate, weight decay and keep probability, as shown in Table 1.

Table 1. Chosen learning hyperparameters for Bayesian search.

name	type	domain
Learning_rate	continues	(0.00001,0.001)
Weight_decay	continues	(0.00001,0.01)
keepProb	continues	(0,1)



(a)

Bayesian optimization algorithm

Input: Data set D
Objective function : F1 score $f(\theta, D)$
Tuning parameters and their domains : $\{\theta\}$
A surrogate function : GP
An acquisition function : $EI(\theta|f)$

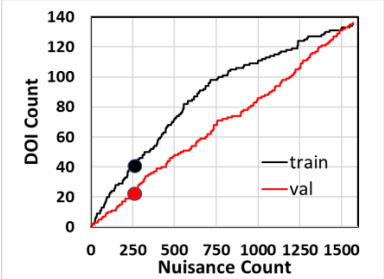
1. For t =1 to T:
   (a) $\hat{\theta}_t = argmax_\theta\{EI(\theta|\hat{f}_{1:t-1})\}$ over GP
   (b) evaluate $\hat{f}_t(\hat{\theta}_t, D)$
   (c) update GP using Bayes' theorem
   $\mathrm{p}(\hat{\theta}_t|\hat{f}_t) = \mathrm{p}(\hat{f}_t|\hat{\theta}_t)\mathrm{p}(\hat{\theta}_t)$
2. Output θ which yields the largest value of f

(b)

Figure 3. (a) Schematic diagram of integrated ConvNet and Bayesian optimization prototype; (b) Bayesian optimization algorithm for hyperparameter search.

4 Results

Figure 4 and 5 show the obtained ROC of defect counts, and confusion matrix of training and validation sets for the models without (M1) and with (M2) Bayesian optimization, respectively. The confusion matrixes (CM) were calculated at same classification threshold (C) for both models. Although the ROCs of both models are similar, the receiver operating points are very different at the same of C (see Figure 4a and 5a). Bayesian optimization can significantly improve the operating points. Therefore, the capture rate of DOI is dramatically increased with Bayesian optimization. This observation appears for both training and validation (see Figure 4b, 4c and Figure 5b, 5c).



Training Confusion Matrix:

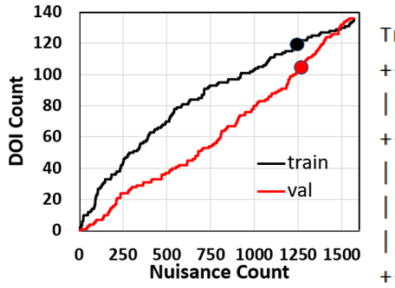
Actual	0.0	1.0	All
0.0	1308	95	1403
1.0	261	40	301
All	1569	135	1704

Validation Confusion Matrix:

Actual	0.0	1.0	All
0.0	1308	114	1422
1.0	261	22	283
All	1569	136	1705

(a) (b) (c)

Figure 4. (a) Defect count ROC, and dots representing the operating points for confusion matrix (CM); (b) training CM; (c)validation CM, obtained from the model without Bayesian optimization (M1). "0" and "1" represent Nuisance and DOI. The column is actual truth, and the row is prediction.



Training Confusion Matrix:

Actual	0.0	1.0	All
0.0	314	18	332
1.0	1255	117	1372
All	1569	135	1704

Validation Confusion Matrix:

Actual	0.0	1.0	All
0.0	308	33	341
1.0	1261	103	1364
All	1569	136	1705

(a) (b) (c)

Figure 5. (a) Defect count ROC, and dots representing the operating points for confusion matrix (CM); (b) training CM; (c)validation CM, obtained from the model with Bayesian hyperparameter search (M2). "0" and "1" represent Nuisance and DOI. The column is actual truth, and the row is prediction.

5 Conclusion/Future work

We demonstrate the use of CNN deep learning method to separate DOI from Nuisance for wafer inspection system. We also show that using the combination of Bayesian optimization and deep learning, we can further improve the model performance by enhancing the receiver operating point. In this work, we only optimized three learning hyperparameters. We expect that in the future, a better model could be obtained by optimizing more hyperparameters including architecture parameters with Bayesian method.

Source code: https://github.com/yuphe/CS230Project

References

[1] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. “Algorithms for Hyper-Parameter Optimization”, In: Advances in neural information processing systems. pp. 2546-2554 (2011)
[2] James Bergstra and Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”, In: Journal of Machine Learning Research 13(Feb), pp: 281-305 (2012)
[3] C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Machine Learning”, the MIT Press, (2006)
[4] Yann LeCun, Léon Bottou, Yoshua Bengio and Patrick Haffner. “Gradient-based learning applied to document recognition” In: Proceedings of the IEEE. 86(11) pp: 2278-2324 (2016)
[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”, In: D. Fleet, T. Pajdla and T. Tuytelaars (eds) Computer Vision -ECCV (2014)