Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network

Takeshi Nakazawa and Deepak V. Kulkarni

***Abstract*—Wafer maps provide important information for engineers in identifying root causes of die failures during semiconductor manufacturing processes. We present a method for wafer map defect pattern classification and image retrieval using convolutional neural networks (CNNs). Twenty eight thousand six hundred synthetic wafer maps for 22 defect classes are generated theoretically and used for CNN training, validation, and testing. The overall classification accuracy for the 6600 test dataset is 98.2%. One thousand one hundred and ninety one real wafer maps are used for CNN performance evaluation for the same model trained by synthetic wafer maps. We demonstrate that by using only synthetic data for network training, real wafer maps can be classified with high accuracy. For image retrieval, a binary code for each wafer map is generated from an output of a fully connected layer with sigmoid activation. A retrieval error rate is 0.36% for the test dataset and 3.7% for the real wafers. Image retrieval takes 0.13 s per wafer map from the 18 000 wafer map library.**

***Index Terms*—Deep learning, convolutional neural network, information retrieval, semiconductor defects.**

I. Introduction

IN THE semiconductor manufacturing, wafer maps are used to visualize defect patterns and identify potential process issues. 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. One of the main purposes for wafer map visualization is to monitor any abnormal defect signatures and respond to process problems quickly. Once wafer map libraries are created with corresponding root causes, defect pattern similarities between wafers could be a good indication of the common root causes and this knowledge base can be used to solve problems. In order to have an effective knowledge base, two components are required: 1) wafer map defect pattern classification and 2) wafer map image retrieval from historical wafer map libraries. The wafer map defect pattern classification can provide information about a defect occurrence rate for each defect class and engineers focus on the most important issue using this data. The wafer map image retrieval is helpful to identify a root cause by querying historical wafer maps with the known root cause.

Manuscript received November 2, 2017; revised December 31, 2017; accepted January 15, 2018. Date of publication January 18, 2018; date of current version May 8, 2018. *(Corresponding author: Takeshi Nakazawa.)*

The authors are with Intel Corporation, Chandler, AZ 85226 USA (e-mail: takeshi.nakazawa@intel.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSM.2018.2795466

There are a number of studies for wafer map pattern recognitions [1]–[4]. Their classification approaches can be divided into two main groups: 1) model-based pattern recognition, 2) feature extraction based pattern recognition. The model-based pattern recognition uses a predefined probability distribution function for each defect pattern and selects the best matching model using information criterion such as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). The feature extraction based pattern recognition extracts pattern features using techniques such as correlogram and Radon transform. Once the pattern features are extracted, the common pattern classification algorithms such as support vector machines, neural networks, nearest neighbors etc. are applied for the classification task.

Deep convolutional neural networks (CNN) [5] have recently advanced the state-of-the-art image classification performance and became the standard approach for any image classification tasks. CNN is the end-to-end model and does not require any task-specific feature engineering. 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. Another aspect of image classification is the problem of image retrieval [6], [7]. The image retrieval is a task of finding images containing similar objects or scene, given a query image, and has been used in security and surveillance, medical imaging, and many other areas. Traditionally, the image retrieval requires feature extraction using object color and shapes. Since the deep CNN can learn rich features at each layer, these intermediate features are used as good descriptors for image retrieval [8], [9].

In this paper, we employ CNN for the defect pattern classification and wafer map retrieval tasks. As a dataset, we use wafer maps from simulation and the real wafers. For CNN training and validation, we only use the simulated wafer maps because real data available for each class from the manufacturing process is highly imbalanced. In this case, it is beneficial to train CNN by using theoretically generated data so that we can also include rare defect patterns to the model and yet achieve reasonable classification accuracy. To verify the performance of the proposed method, we generated 28,600 dataset by simulation. Data from 1,191 real wafers are also used to evaluate the performance of the trained CNN.

Our paper is organized as follows. In Section II, methods for wafer map pattern generation, the CNN configuration and CNN based image retrieval are descried. In Section III, we present the results of defective wafer map pattern generations, the CNN training/validation/test results using theoretically

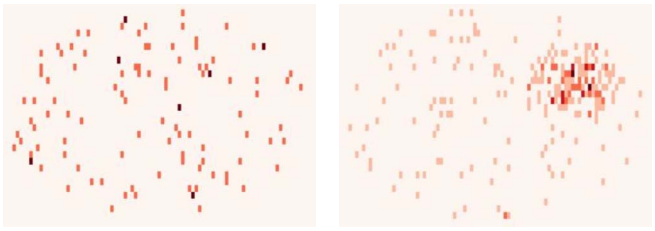

Fig. 1. Defect density wafer map with the random defects (left) and with the random and the non-random defects (right).

generated wafer maps. The performance of trained CNN is also validated using data from the real wafers. Then the image retrieval result is shown by comparing a query image and the top three retrieved wafer maps from the 18,000 wafer library. The conclusion is given in Section IV.

II. Method

A. Wafer Map Pattern Generation

Defect patterns can be categorized into three types: 1) random pattern, 2) non-random pattern and 3) the superposition of random and non-random patterns. Typical wafer map shows either pure random defect pattern, or random and non-random mixed pattern. Fig. 1 illustrates these two examples.

From the yield analysis and the process improvement perspective, wafers showing non-random patterns are more important since it clearly indicates the process related issue. In general, random defects with the controlled number of defects are acceptable and it provides less information from the process improvement perspective.

There are different ways to visualize wafer map based on the purpose of analysis. For example, if engineers are interested in good versus bad die locations, a binary (pass/fail) wafer map is used. Sometimes engineers need to know the frequency of defects at each die, and in this case defect frequency density map is the right method. The benefit for the density map is that it provides additional information about spatial defect occurrence rates, which could be helpful for engineers to understand a problem more deeply, as opposed to the binary wafer map that only provides pass/reject unit information. In our study, we use the density wafer map.

The wafer map defect pattern is modeled using Poisson point process. The Poisson distribution is given by

$$P(k, \Lambda) = \frac{\Lambda^n}{k!} e^{-\Lambda}, \tag{1}$$

where Λ is often called the rate parameter that defines the average number of events in an interval. The number of events is defined by k. By following the algorithm described in [10], we generate random points in the polar coordinates. In addition, non-random points are superimposed by controlling the interval of the uniform distribution used in the algorithm. Once all points are generated for a single wafer map, a density map is created by summing up the number of points within each die boundary and normalized by the maximum number among all dies.

TABLE I
CNN Configuration

32 3 x 3 2D convolutional layer
Rectified linear activation
Max pooling layer
32 3 x 3 2D convolutional layer
Rectified linear activation
Max pooling layer
64 3 x 3 2D convolutional layer
Rectified linear activation
Max pooling layer
Fully connected layer 256
Sigmoid activation
Dropout
Fully connected layer 22
Softmax

B. Convolutional Neural Network Configuration

Table I shows our CNN configuration. The input wafer map image size is 286 x 400. We have three convolutional layers with the receptive field size of 3 x 3 and stride 1. The first and second convolutional layers have the 32 channels and the third convolutional layer has the 64 channels. The rectified linear activation is used for each convolutional layer. The max pooling size is 2 x 2. The fully connected (FC) layer with the size of 256 is added after the convolutional layers with sigmoid activation. After dropout, another fully connected layer with the size of the defect class is added. The last layer is the softmax layer for the class probability calculation.

C. Wafer Map Image Retrieval Using Convolutional Neural Network

Since images are high-dimensional data, dimensionality reduction is essential to achieve rapid search in a large database. To achieve this goal, we follow the similar approach described in [9]. In our case, no latent layer is required due to the node size of the FC layer. For the CNN configuration used in their study, the layer 7 has the 4096 nodes and it is still too large for efficient database search. Since our FC layer has the 256 nodes, we simply use the features extracted at this layer after the sigmoid activation. To get a binary code for each wafer map, we applied the threshold value of 0.5 to the output of the sigmoid activation, i.e., if a value is greater or equal to 0.5, the value is 1 and 0 otherwise. Once the binary code library is built for the entire wafer map, the Hamming distance measure is used to retrieve similar wafer maps, given a query wafer map.

III. Result

A. Wafer Map Pattern Generation

We defined the 22 defect classes for our classification task. Table II is the list of the defect patterns. The simulated wafer maps are with 1) pure random defects, 2) random defects and typical non-random defects and 3) random defects and multiple different non-random defect types. In order to evaluate the classification performance for wafer maps showing multiple defect classes, we added the class with line scratch

Fig. 2. The example of the generated wafer map for each class.

defect and non-random cluster defect at each quadrant. The simulated defect classes contain the similar defect patterns but its location is at the different area of the wafer map. For example, the non-random cluster defects for each quadrant are considered as the different class. The reason is that sometimes the defect location provides locational commonality

TABLE II
LIST OF WAFER MAP DEFECT CLASS

Class label	Wafer map defect class name
C1	Random defect
C2	Wafer edge ring defect
C3	Wafer right side edge defect
C4	Wafer left side edge defect
C5	Line scratch defect
C6	Curved scratch defect
C7	Non-random cluster defect at 1st quadrant
C8	Non-random cluster defect at 2nd quadrant
C9	Non-random cluster defect at 3rd quadrant
C10	Non-random cluster defect at 4th quadrant
C11	Non-random cluster defect at center
C12	Non-random cluster defect at top
C13	Non-random cluster defect at bottom
C14	Non-random cluster defect at right
C15	Non-random cluster defect at left
C16	Gross defect at entire wafer
C17	Gross defect at left half of wafer
C18	Gross defect at right half of wafer
C19	Line scratch with non-random cluster defect at 1st quadrant
C20	Line scratch with non-random cluster defect at 2nd quadrant
C21	Line scratch with non-random cluster defect at 3rd quadrant
C22	Line scratch with non-random cluster defect at 4th quadrant

information for a specific process tool and helps to identify the specific issue.

Fig. 2 illustrates the example wafer maps for each defect patterns. We use the defect density map for this study.

B. Wafer Map Classification Accuracy

We train our CNN as follows. First, the 1,300 wafer maps are generated for each class using the method described in the previous section. Then, we split these images randomly into 1) 700 training data set, 2) 300 validation data set and 3) 300 test data set. The 700 wafer maps for each class is used for training our CNN and the 300 wafer maps are used for the validation. Once the desired training/validation accuracy is achieved using the 15,400 training and 6,600 validation images, the 6,600 test images are used to evaluate our CNN performance. The training accuracy after the 10 epoch is 99.8% and the validation accuracy is 97.8% for the simulated wafer maps. Fig. 3 is the confusion matrix for the test dataset. Most of the class accuracy is greater than 95% except 89.0% for the line scratch defects (C5) and 87.7% for the curved scratch defects (C6). The line scratch is misclassified as curved scratch and vice versa. The overall accuracy is 98.2%.

Table III shows the relationship between the batch size and the average memory usage in percentage during the network training phase.

The training and validation accuracy for each epoch is shown in Fig. 4. The average processing time for each epoch is 110.6 seconds.

Fig. 5 illustrates the misclassified wafer map examples with the top 5 class probabilities. The true class is the curved scratch defect (C6) but it was misclassified with the line scratch defect (C5). The class probability is 55.9% for C5 and 42.8% for C6.

In addition to the simulated wafer map, we test the CNN inference results using the 1,191 real wafers. Fig. 6 is the confusion matrix and it shows the per-class classification

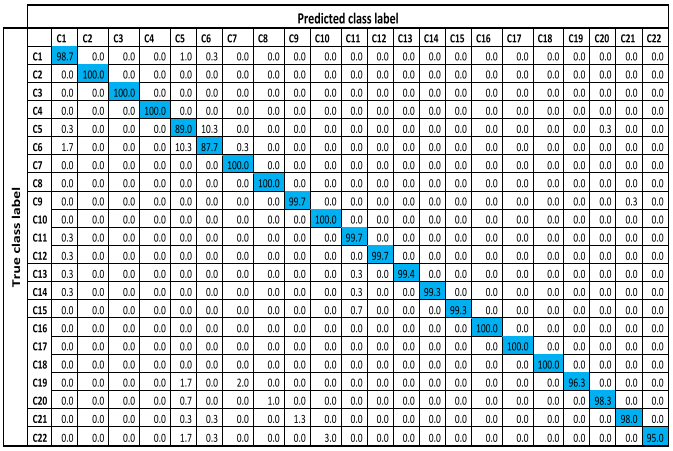

Fig. 3. Accuracy confusion matrix in percentage for the simulated test wafer maps.

TABLE III
BATCH SIZE AND MEMORY USAGE

Batch size	Average memory usage (%)
385	43.4
770	63.3
1540	68.2
3080	74.9

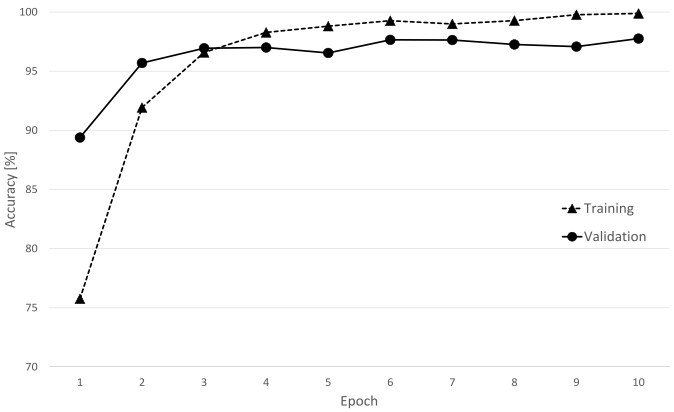

Fig. 4. The training and validation accuracy.

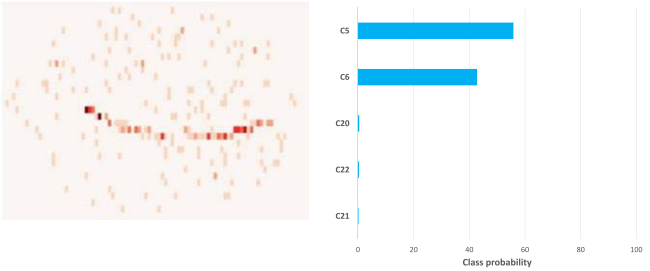

Fig. 5. The misclassified wafer map (left) and the top 5 class probability (right).

True class label \ Predicted class label	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22
C1	99.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0
C2	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	0	×	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C4	0	0	0	×	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C5	0	0	0	0	×	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C6	33.3	0	0	0	0	66.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C7	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C8	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C9	0	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0	0	0	0	0	0
C10	0	0	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0	0	0	0	0
C11	0	0	0	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0	0	0	0
C12	0	0	0	0	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0	0	0
C13	0	0	0	0	0	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0	0
C14	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
C15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83.3	0	16.7	0	0	0	0	0
C16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
C17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
C18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
C19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0	0	0
C20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0	0
C21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0
C22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Fig. 6. Accuracy confusion matrix in percentage for the real wafer maps.

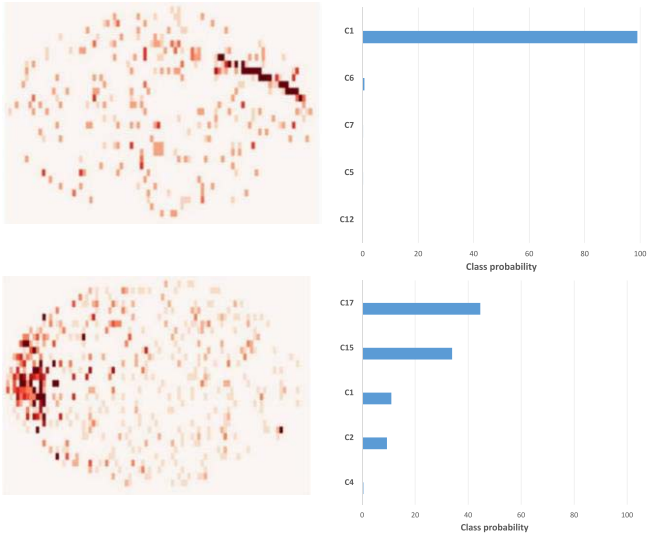

Fig. 7. The misclassified wafer map from the real wafer (left) and the top 5 class probability (right).

accuracy in percentage. This real dataset contains not all 22 classes but 9 classes and the crosses along the diagonal of the table indicate no real wafer map data for that particular class. The dataset is imbalanced and the dominant class is the random defects (C1). Due to the confidentiality reason, we can only provide per-class accuracy, not the absolute number of wafers. The curved scratch defect shows the 66.7% classification accuracy based on the data size of 3 wafers and it is the rare defect class within this dataset.

Fig. 7 illustrates the misclassified real wafer map examples with the top 5 class probabilities. The top figure is the misclassification example of the curved scratch defect (C6) and the prediction is the random defect (C1). The class probability for C1, C6 is 98.9% and 0.006% respectively. For the bottom example, the true class is the non-random cluster defect at left (C15) but it was misclassified with the gross defect at left half of wafer (C17). The class probability is 44.6% for C17 and 33.9% for C15.

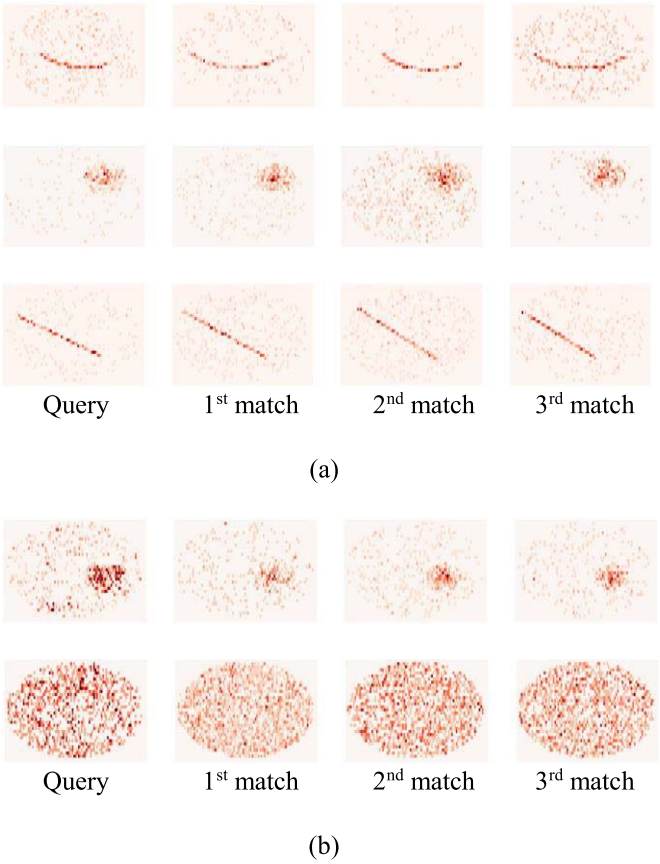

Fig. 8. Query wafer map (1st column) and the corresponding top 3 retrieved wafer map images for the selected defect patterns. (a) Query wafer map is from simulation. (b) Query wafer map is from the real wafer.

TABLE IV
IMAGE RETRIEVAL ERROR RATE

Dataset	Error rate (%)
6,600 simulated wafer map	0.36
1,191 real wafer map	3.7

C. Wafer Map Image Retrieval

Fig. 8 shows the wafer map image retrieval results for the selected defect class. The first image at each row is the query image and the rest of images are the top 3 retrieved wafer map images. As we can see from these examples, the algorithm successfully retrieved similar wafer map from the library.

For image retrieval performance evaluation, we check an error rate based on a top 1 retrieved image class and a true class. Table IV summarizes the result. The image retrieval takes 0.13 seconds per image from the 18,000 wafer map library with the basic Python code implementation.

IV. CONCLUSION

In this paper, we present a method for wafer map pattern classification and wafer map image retrieval using CNN. In the semiconductor manufacturing, rare event detection is critical to maintain high yield. We demonstrate the benefit of using theoretically generated wafer maps for CNN training to enable classification tasks for the imbalanced dataset from

the real wafers. Without having enough number of dataset, CNN cannot be trained well and it is difficult to have enough data size in some cases if defect patterns happen rarely. Our model enables rare event detection capability without having real data and it is particularly beneficial during technology development phase.

We also demonstrate efficiency and performance of CNN based image retrieval using the binary code generated by the FC layer of our CNN model. Once the root causes and solutions of a particular defect mode are associated with its wafer map pattern(s), wafer map image retrieval can be used to trigger the actions for problematic processes.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *J. Vis. Commun. Image Represent.*, vol. 10, no. 1, pp. 39–62, Mar. 1999.

[7] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[8] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland, 2014, pp. 584–599.

[9] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Boston, MA, USA, 2015, pp. 27–35.

[10] R. Pasupathy, "Generating homogeneous Poisson processes," in *Wiley Encyclopedia of Operations Research and Management Science*. Hoboken, NJ, USA: Wiley, Jan. 2011.

Takeshi Nakazawa received the Ph.D. degree in optical sciences from the College of Optical Sciences, University of Arizona in 2011.

He is currently working with Intel Corporation, Chandler, AZ, USA, as a Yield Engineer/Data Scientist for developing image and data analysis systems and yield prediction models using machine learning. He was the recipient of several Intel divisional and department awards, the Best Paper Award for *Intel Technology Journal*, and several distinguished invention awards.

Deepak V. Kulkarni received the Ph.D. degree in mechanical engineering from the University of Illinois at Urbana-Champaign in 2005. He currently serves as an Engineering Technology Development Manager with the Assembly and Test Technology Development Group, Intel Corporation, Chandler, AZ, USA. His interests are in applying big data analysis techniques to improve manufacturing yield.