Anomaly Detection and Segmentation for Wafer Defect Patterns Using Deep Convolutional Encoder–Decoder Neural Network Architectures in Semiconductor Manufacturing

Takeshi Nakazawa and Deepak V. Kulkarni

***Abstract*—Abnormal defect pattern detection plays a key role in preventing yield loss excursion events for the semiconductor manufacturing. We present a method for detecting and segmenting abnormal wafer map defect patterns using deep convolutional encoder–decoder neural network architectures. Using a defect pattern generation model, we create synthetic wafer maps for 8 basis defect patterns, which are used as training, validation, and test datasets. One of the key capabilities for any anomaly detection system is to detect unseen patterns. We demonstrate that by using only synthetic wafer maps with the basis patterns for network training, the models can detect unseen defect patterns from real wafer maps.**

***Index Terms*—Deep learning, encoder, decoder, convolutional neural networks, anomaly detection, segmentation, semiconductor defects.**

I. Introduction

IN THE semiconductor manufacturing, engineers use wafer defect maps to visualize defect patterns and identify potential process and tool issues. There are different use cases for wafer maps and depending on the process maturity, the way wafer maps is used could be different. Two main use cases are: 1) finding out the existence of common defect patterns among different wafers for commonality analysis and root cause identification, 2) detecting unknown patterns that never exist before for excursion prevention. First, understanding common defect patterns helps to segment problems and quantify an occurrence rate, and eventually each unique pattern will be connected to root cause(s) of the problem so that a correct response flow is associated with the defect pattern and its solution. For example, line scratch pattern is associated with wafer handling at a certain process tool. To identify similar defect pattern groups, one can use unsupervised learning such as clustering methods. If the purpose is to classify patterns into predefined pattern categories or classes, one can use supervised learning. Secondly, detecting unknown pattern, or anomaly detection, is another important aspect because it could be an early signal of potential yield loss events. A system needs to detect anomaly signals first, which is the object detection problem, and then compares against the existing defect pattern library if the detected signal shows any known pattern or not.

The wafer map pattern recognition can be divided into two approaches: 1) model-based pattern recognition, 2) feature extraction-based pattern recognition [1]–[5]. For model-based pattern recognition, a predefined probability distribution function is used for each pattern and the best model is determined by comparing models using information criterion. For feature extraction-based pattern recognition, unique defective pattern features are extracted first and then different pattern classification algorithms are applied to classify these patterns. The feature extraction can be performed using different algorithms such as nearest neighbor method, correlogram and Radon transform etc.

Deep learning has been huge success lately in many different areas, particularly in computer vision. The main problem in computer vision is object recognition, which can be divided into three groups: object classification, object detection and semantic segmentation [6]. The object classification is a task of predicting presence or absence of a class in a test image. The object detection requires to find a minimum bounding box and assign a class label for each object. The semantic segmentation requires to assign pixel-wise class label, which is one of the active research areas. In autonomous driving applications, for example, semantic pixel-wise labelling is required to map video (image) captured by a camera to scene categories such as road, buildings, cars, pedestrians etc. Datasets used for training these models are input image and corresponding segmentation map pairs with pixel-wise labels. There are a number of different deep learning architectures for semantic segmentation such as Fully Convolutional Networks (FCN) [7], Deconvolution Network [8], SegNet [9], U-Net [10] and DeepLab [11].

In our previous literature, we demonstrated that deep convolutional neural networks (DCNN) can be used to classify defect patterns and to retrieve similar defect patterns from the library, given a query defect pattern [12]. In this paper, we focus on the defect pattern detection and segmentation problem and choose FCN, SegNet and U-Net as the base architectures for comparing anomaly defect detection performances. The difference from our previous study is as

Manuscript received October 1, 2018; accepted February 1, 2019. Date of publication February 5, 2019; date of current version May 3, 2019. *(Corresponding author: Takeshi Nakazawa.)*

The authors are with the Assembly and Test Technology Development, 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.2019.2897690

follows. Since DCNNs are the end-to-end training model, it does not require any intermediate pattern extraction steps. Inputs are wafer map images and outputs are corresponding defect classes. This type of architecture is appropriate if the primary interest is to know the defect classes. For some cases, the engineers need to understand more detail information from the defect cluster such as its location, size, major and minor axis length, orientation etc. These detail information help for further analysis in addition to the defect cluster classes, especially during initial technology development phase when the engineers do not have full understanding about the defective patterns. To extract these information, the defect cluster(s) should be identified and segmented first to perform subsequent data extraction steps and thus different neural network architectures are required. This paper discusses about defect cluster pattern detection/segmentation using the different type of CNN models.

FCN is the first work to train FCN end-to-end for pixel-wise prediction. The key idea of FCN is to replace the fully connected layers of typical classification neural networks with convolutional layers so that a network output can be a two dimensional heat map, rather than class probability prediction. FCN implements "skip architecture", meaning that shallow layer's outputs are merged to deeper layers so that the network can maintain both local and coarse information. SegNet consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The encoder network has 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 network [13]. Each encoder layer has a corresponding decoder layer, namely 13 corresponding convolutional layers in the decoder network. The decoder uses pooling indices computed in a max pooling step of the corresponding encoder to perform non-linear up-sampling. Since positional or boundary information are lost during the max pooling operations in the encoder network, maintaining positional information for each up-sampling operation in the decoder network is critical for accurate pixel-wise segmentation. U-net is originally proposed for a biomedical image segmentation application and has encoder-decoder architecture. Instead of using the pooling indices in the max pooling step, U-net copies and crops a feature map in each encoder layer to the corresponding decoder layer to maintain local positional information. For the segmentation problem, maintaining local information is the key so that each architecture has its own way to transfer local details from shallower layers to deeper layers.

As datasets, we use synthetic wafer maps using a pattern generation model and real wafer maps. For the network training, validation and testing, we only use the synthetic wafer maps. The real wafer map data is used only for model testing because one of the main objectives is to train the models without using the real wafer maps and yet achieve the practical performance. As input datasets, we generate abnormal defect patterns with random defects whereas target output datasets contain only the abnormal defect patterns without random defects. Using this pair dataset, the goal is to train the neural network models such that only abnormal defect patterns can be extracted. To verify the performance of the proposed method, we generated two different synthetic datasets, 1) basis defect patterns used for training, validation and testing, and 2) unseen defect patterns used only for testing. In addition, data from 1,191 real wafers are also used to evaluate qualitative performance of the trained neural network model.

The remainder of the paper is organized as follows. In Section II, methods for wafer map pattern generation and several convolutional encoder-decoder neural network architectures are descried. In Section III, we evaluate the performance on synthetic test and real wafer datasets. We also discuss the pattern detection capability for unseen defect patterns, which is an important element in real production scenario since preventing excursion is the key for any defect pattern detection tool. The conclusion is given in Section IV.

II. Method

A. Wafer Map Pattern Generation

For wafer map pattern generation, we follow the same method described in our previous literature [12], [14]. The only difference is that we use binary wafer maps instead of density wafer maps since our primary interest is to isolate defect cluster(s). The wafer maps with random and non-random defect clusters are generated using Poisson point process, which is given by

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

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. Once all points are generated for a single wafer map, a binary map is created by checking presence or absence of defects on a particular die.

B. Deep Convolutional Encoder-Decoder Neural Network Architecture

Fig. 1 shows the schematics of the first convolutional encoder-decoder architecture based on SegNet. The network has the encoder and decoder with the sigmoid activation layer for pixel-wise classification at the last layer. The inputs to the network are wafer maps with defect cluster(s) and random defects. The target outputs have only the defect cluster(s) as illustrated in Fig. 1. There are two convolutional layers, followed by a batch normalization, and then a rectified linear activation (ReLU) is applied at each layer. 2×2 max-pooling is performed after each 2-convolutional layer. For the decoder, we have 2×2 up-sampling followed by 2-convolutional layer. The receptive field size is 3×3 for all convolutional layers and the channel sizes are 128, 64 and 32, indicated by the size of rectangle boxes. We use a binary cross entropy as a loss function. The input wafer map size is 344×480.

Fig. 2 illustrates the schematics of the second convolutional encoder-decoder architecture based on U-Net. In this architecture, we have the two convolutional layers with 32, 64, 128 channel sizes similar to the previous architecture. The difference is the merge layer illustrated in yellow boxes. The local information from the shallow layers are merged

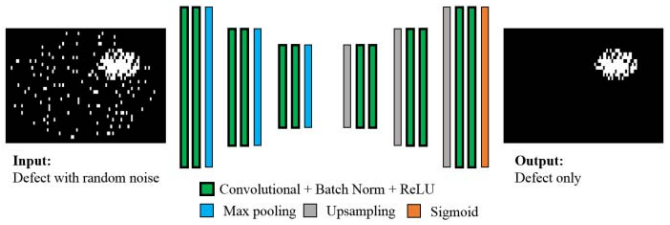

Fig. 1. Deep convolutional encoder-decoder neural network schematics based on SegNet.

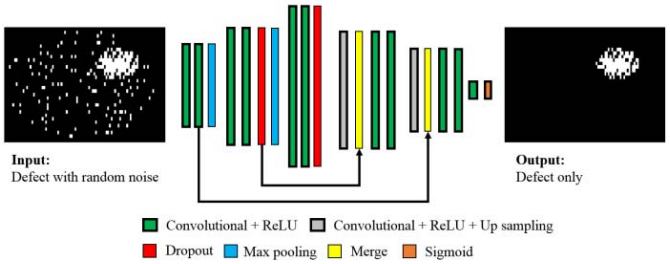

Fig. 2. Deep convolutional encoder-decoder neural network schematics based on U-Net.

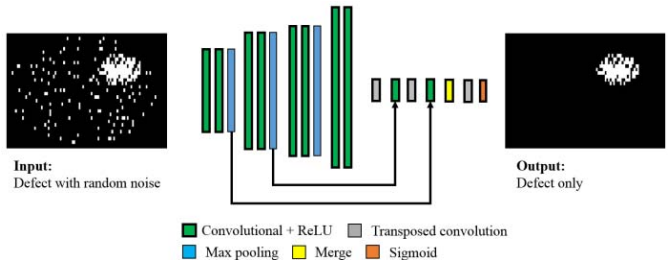

Fig. 3. Deep convolutional neural network schematics based on FCN.

to the deeper layers. The receptive field size is 3 × 3 for all convolutional layers except the last layer whose field size is 1 × 1 with the sigmoid activation.

The third architecture is shown in Fig. 3 based on FCN. We have the two convolutional layers with 32, 64, 128 channel sizes with the receptive field size of 3 × 3 and the stride of 2 × 2. Then two convolutional layers with 256 are added with the stride of 7 × 7 and 1 × 1 respectively. The next layers are the combination of two transpose convolutional layers and convolutional layers whose input comes from the shallow layers' pooling layers.

Fig. 4 shows examples of input and target pattern pairs used for the neural network training and validation. The images left show the wafer map pattern with the random defects and abnormal defect cluster. The images right illustrate the abnormal defect cluster only, which is our target detection.

III. Result

A. Wafer Map Pattern Generation

We define two different groups of synthetic patterns to test our model. The first dataset is the basis defect patterns used for the model training, validation and test. The second dataset is used only for testing our model capability to detect unseen defect patterns. They are either the combination of

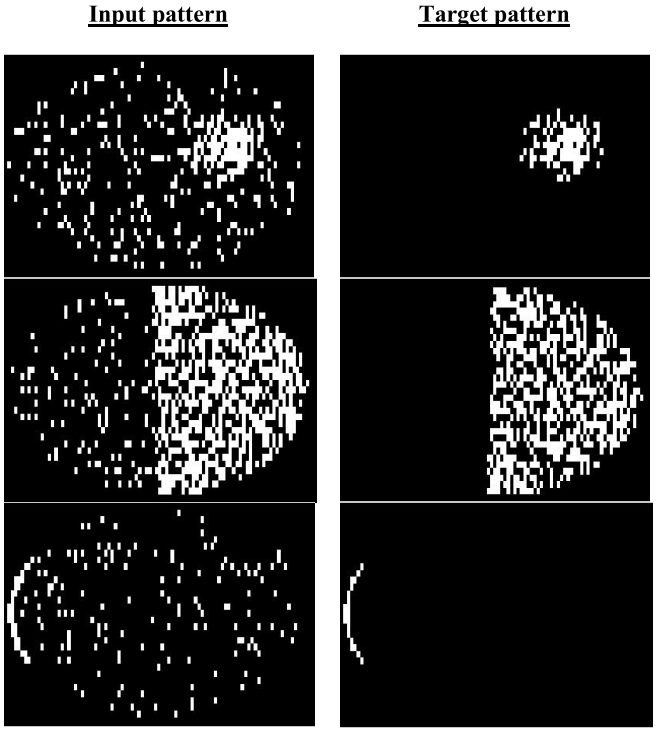

Fig. 4. Input and target wafer map pattern pairs.

TABLE I
List of Basis Pattern

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	Non-random cluster defect
C7	Gross defect at left half of wafer
C8	Gross defect at right half of wafer

TABLE II
List of Pattern to Test Unseen Defects

Class label	Wafer map defect class name
C9	Gross edge damage defect
C10	Curved scratch defect
C11	Line scratch with non-random cluster defect

basis defect patterns or new patterns. Table I is the list of the basis defect patterns and Fig. 5 shows the corresponding wafer map examples.

Table II is the list of the unseen defect patterns and Fig. 6 shows the corresponding wafer map examples. To check the performance of detecting multiple defect patterns in a single wafer, line scratch with non-random cluster defect pattern is added.

B. Model Training and Test Result

We have 17,000 total dataset and split them into 70% training and 30% validation dataset. In addition, 3,300 test dataset is used for testing the model performance, which is

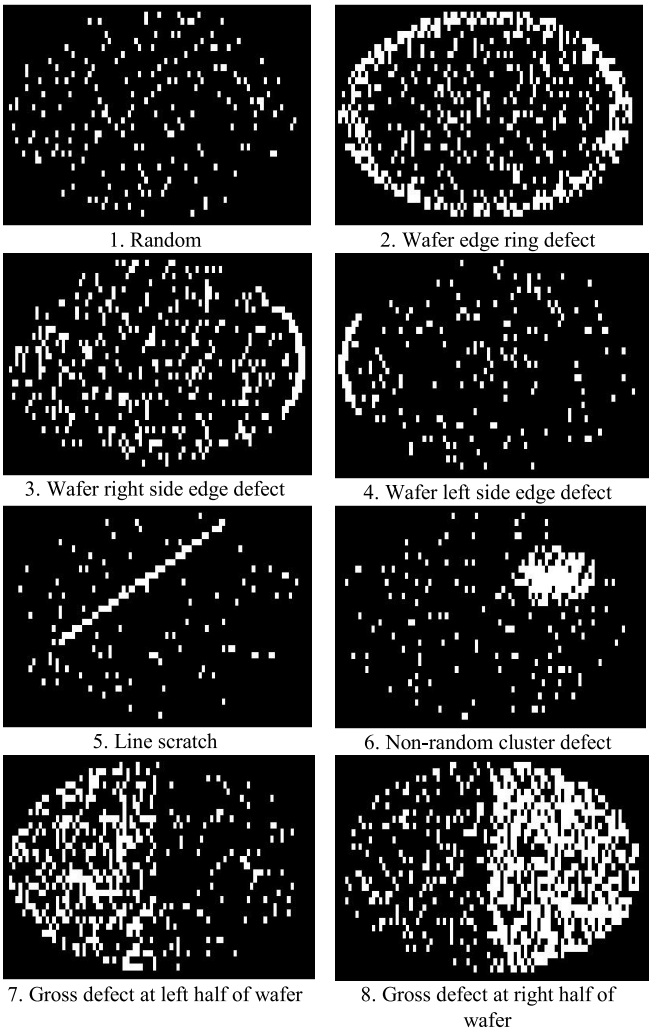

Fig. 5. The examples of the synthetic wafer maps used for model training and validation.

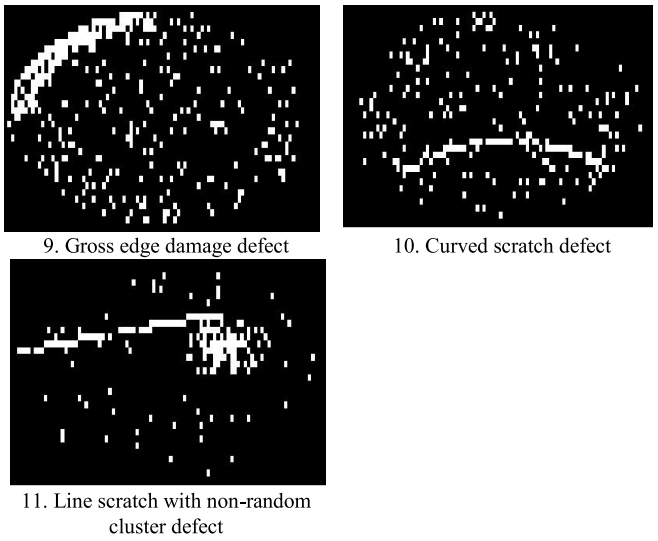

Fig. 6. The example of the synthetic wafer maps used for unseen defect pattern detection test.

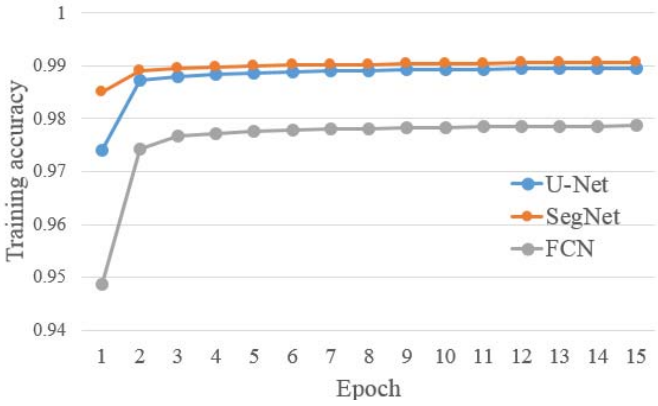

Fig. 7. Training accuracy versus epoch for each model.

not used during training and validation phases. Fig. 7 illustrates the training accuracy for each model. Among these models, SegNet and U-net based architectures show similar training accuracy. FCN based architecture shows the lower performance as compared with these two models. The mean training accuracy between epoch 5 and 9 is 0.989, 0.990, 0.978 for U-Net, SegNet and FCN based architecture respectively. After around 5 epochs, there is no significant increase in accuracy performance for all models.

Table III describes the average training time in minutes per epoch.

In order to determine object detection performance, we use intersection over union (IoU), which is defined as follows, per each image. For class level evaluation, we use mean intersection over union (mIoU).

$$J(A, B) = \frac{|A \bigcap B|}{|A \cup B|} \tag{2}$$

where A and B are a set. The intersection of the sets A and B in the numerator is the set that contains all elements of A that also belong to B. The union in the denominator is the set of all elements in the sets A and B.

TABLE III
TRAINING TIME IN MINUTES FOR EACH ARCHITECTURE

Architecture	SegNet	U-Net	FCN
Avg. time/epoch [min]	19.9	10.0	4.2

Table IV shows the mean +/− one standard deviation mIoU results for the basis pattern test dataset and Fig. 8 illustrates inference results with relatively high (0.94) and low (0.71) IoU examples. For all 7 basis defect patterns, mIoU are greater than 0.5, which in general is considered as a successful detection.

C. Test Results for Unseen Defect and Real Wafer Defect Patterns

Table V shows the mean +/− one standard deviation mIoU results for the unseen defect patterns. Fig. 9 is the object detection result. IoU for the image top is 0.94 whereas it is 0.68 for the image bottom. The system can capture the defect cluster at bottom right but it also captures some random defects. SegNet and U-Net based architectures have better performance as compared with FCN based architecture.

TABLE IV
MIOU RESULT FOR BASIS PATTERN TEST SET

Pattern\Architecture	SegNet	U-Net	FCN
Wafer edge ring	0.655+/-0.12	0.728+/-0.08	0.428+/-0.08
Right side edge	0.879+/-0.07	0.850+/-0.11	0.804+/-0.09
Left side edge	0.883+/-0.11	0.794+/-0.19	0.330+/-0.26
Line scratch	0.649+/-0.25	0.663+/-0.23	0.538+/-0.23
Non-random cluster	0.815+/-0.09	0.817+/-0.09	0.646+/-0.11
Gross defect at left	0.753+/-0.10	0.799+/-0.06	0.459+/-0.08
Gross defect at right	0.876+/-0.04	0.877+/-0.05	0.640+/-0.04

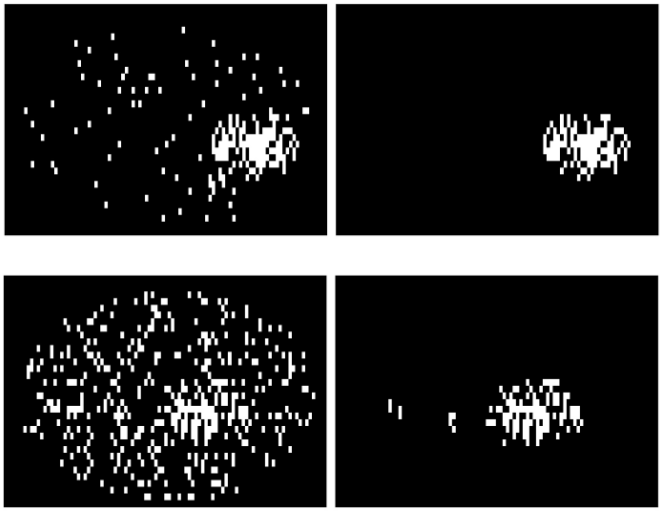

Fig. 8. The example detection results for relatively high IoU (top) and low IoU (bottom).

TABLE V
MIOU RESULT FOR UNSEEN DEFECT PATTERNS

Pattern\Architecture	SegNet	U-Net	FCN
Gross edge damage	0.912+/-0.05	0.894+/-0.07	0.776+/-0.08
Curved scratch	0.824+/-0.10	0.808+/-0.12	0.769+/-0.10
Line scratch with non-random cluster	0.743+/-0.08	0.771+/-0.09	0.610+/-0.09

To understand defect detection performance based on the defect density differences, Figs. 10-12 show the relationship between IoU and the defect density (each data point represents a single wafer). The defect density is calculated by the number of bad units divided by the total number of units in the wafer.

Tables VI and VII show a slope of linear fit and R square metric to understand the correlation between IoU and defect density. The first observation is that although FCN based architecture gives the lower mIoU result, IoU does not decrease as the defect density increases as compared with the other two architectures. The second observation is that between SegNet and U-Net based architectures, IoU from SegNet does not decrease as much, compared with that from U-Net based architecture, as the defect density increases. The performance for the defect density less than 0.1 is comparable between the two architectures.

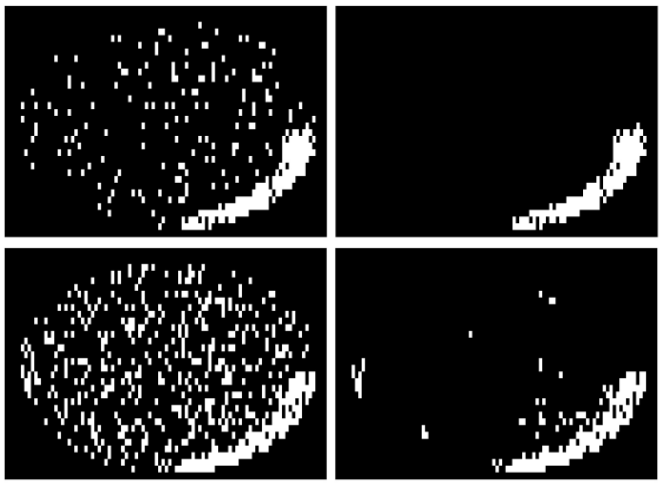

Fig. 9. The example detection results for relatively high IoU (top) and low IoU (bottom).

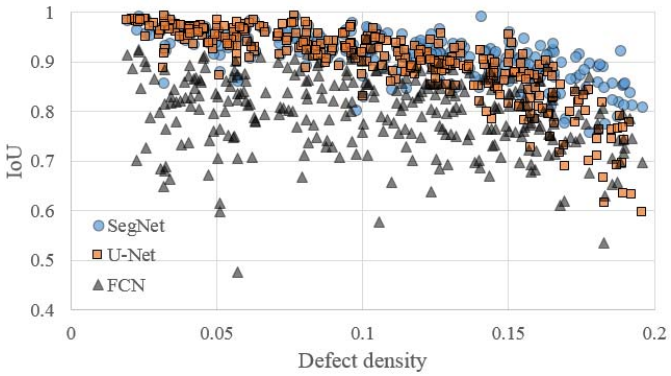

Fig. 10. IoU versus defect density correlation for unseen defect gross edge damage.

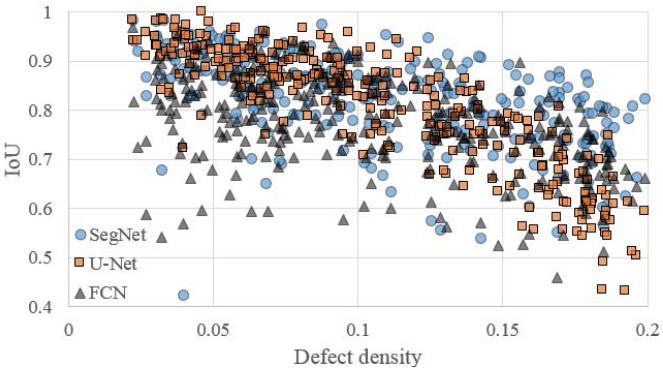

Fig. 11. IoU versus defect density correlation for unseen defect curved scratch.

Finally, we test the defect pattern detection capability using the dataset from 1,191 real wafers. For this real wafer dataset, we use the SegNet based architecture. Among 1,191 wafers, 22.9% is the real defects and 77.1% of wafers show only random defects. Table VIII shows the result. The model detects all the real defect, so detectability is 100%. For the random defects, if the output is zero across the entire wafer map field,

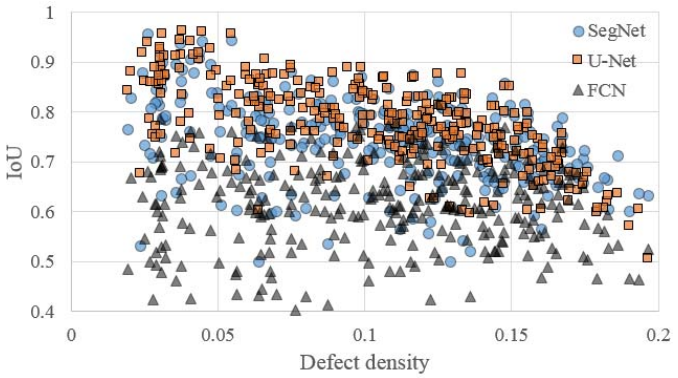

Fig. 12. IoU versus defect density correlation for unseen defect line scratch with non-random cluster.

TABLE VI
SLOPE OF LINEAR FIT FOR UNSEEN DEFECT PATTERNS

Pattern\Architecture	SegNet	U-Net	FCN
Gross edge damage	-0.7	-1.2	-0.3
Curved scratch	-1.0	-2.2	-0.9
Line scratch with non-random cluster	-0.8	-1.2	0.03

TABLE VII
R SQUARE FOR UNSEEN DEFECT PATTERNS

Pattern\Architecture	SegNet	U-Net	FCN
Gross edge damage	0.5	0.65	0.06
Curved scratch	0.3	0.74	0.21
Line scratch with non-random cluster	0.16	0.43	0.0002

we call it as the correct detection. If, on the other hands, some units remain in the output, we call it as the wrong detection.

Fig. 13 shows some qualitative examples for the defective wafer maps. The images left show the original real wafer maps and the images right are the inference results. Some real wafers show unique patterns that are not similar to any training patterns including wafer maps showing multiple defects, yet the SegNet based model can detect these patterns successfully.

Fig. 14 shows the random wafer examples for the correct detection (top) and wrong detection (bottom). Some units are not removed well from the wafer maps. For most of the cases, these remaining unit areas are small and localized, so additional filtering algorithms, such as removing small cluster areas, could be implemented for further improvements.

D. Scalability Consideration

Although our primary intent of this paper is to demonstrate the methodology of using synthetic dataset for network training and perform acceptable defect detection tasks for the real wafers, it is also important to consider the scalability of this method. To demonstrate this, we increase our total synthetic wafer dataset size from 17,000 to 26,000 and compare the training time and the overall mIoU for the same test dataset in Table IX. In this study, we use the SegNet based architecture. The training time and overall mIoU increase from 19 minutes to 30 minutes, 0.796 to 0.828 respectively. We believe this is reasonable since the total training time is about a few hours to achieve reasonable performance.

TABLE VIII
DEFECT DETECTION PERCENTAGE

Defect type	Defect	Random
Correct Detection	100%	94.3%
Wrong Detection	0%	5.7%

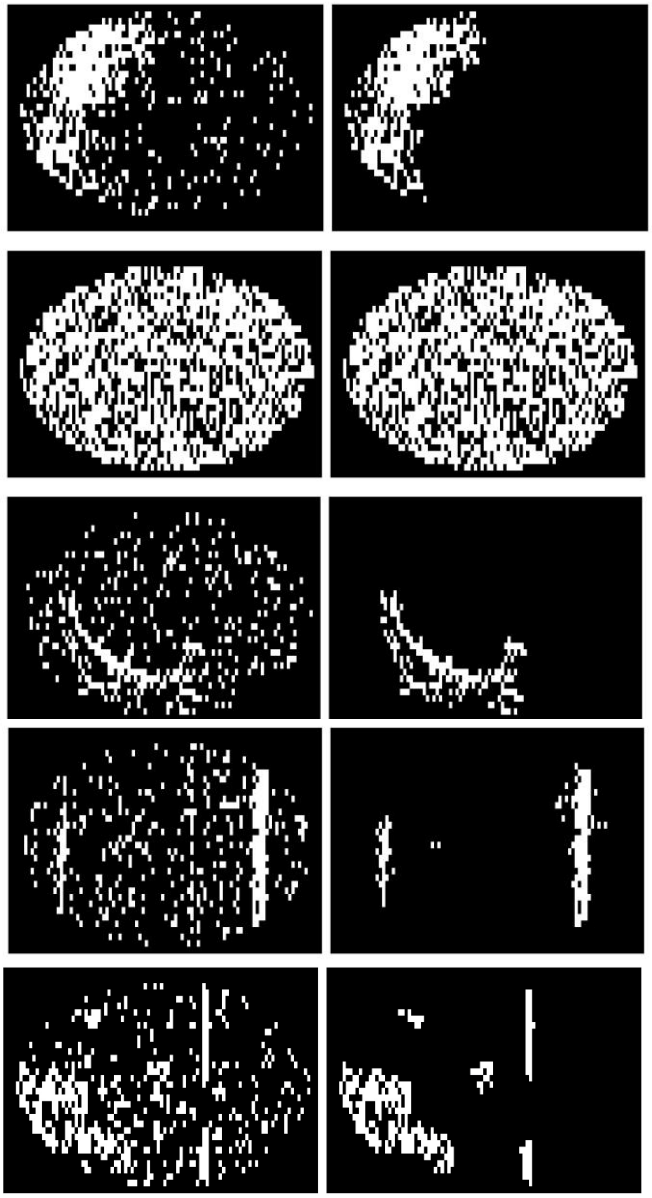

Fig. 13. The defect pattern examples from real wafer maps. The patterns from the real wafers (left) and inference results (right).

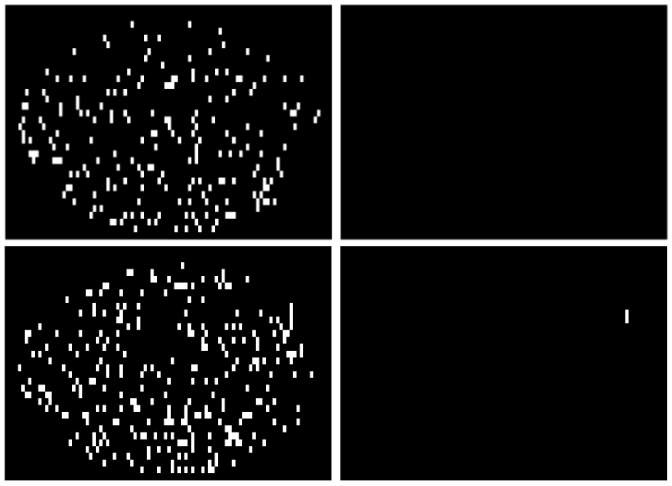

Fig. 14. The random pattern examples from real wafer maps. The patterns from real wafers (left) and inference results (right). The first example show correct detection and the second image show the wrong detection.

TABLE IX
TRAINING TIME IN MINUTES AND MIOU COMPARISON

Architecture	17,000 dataset	26,000 dataset
Avg. time/epoch [min]	19.9	30.5
Overall mIoU	0.796	0.828

IV. CONCLUSION

In this paper, we present a method for abnormal defect pattern detection and segmentation using deep convolutional encoder-decoder neural network architectures. Anomaly detection from wafer maps plays a key role in the semiconductor manufacturing to prevent any excursion events and to understand process and tool issues. We demonstrate that by training the models with only synthetic wafer maps, the models can successfully detect unseen defect patterns from the real wafer maps. This capability of detecting abnormal signals without using training dataset from the real wafers is useful since excursion events happen rarely, which means that we don't have the enough number of training dataset initially. During technology development phase, the engineers need to understand defective cluster characteristics more than just knowing the defect cluster classes. Our proposed solution can be used as the initial step to extract the pattern and subsequent data analysis can be performed, for example, using standard image processing methods and machine learning techniques to obtain relevant information such as locations, size, orientation etc.

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] C.-J. Huang, C.-F. Wu, and C.-C. Wang, "Image processing techniques for wafer defect cluster identification," *IEEE Design Test Comput.*, vol. 19, no. 2, pp. 44–48, Mar./Apr. 2002.

[4] 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.

[5] 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.

[6] T. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[7] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[8] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. ICCV*, 2015, pp. 1520–1528.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, 2015, pp. 234–241.

[12] T. Nakazawa and D. 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, May 2018.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–12.

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