A Fast Hotspot Detector Based on Local Features Using Concentric Circle Area Sampling

Hidekazu Takahashi Shimpei Sato Atsushi Takahashi
Department of Information and Communications Engineering, Tokyo Institute of Technology, Japan
h_takahashi@eda.ict.e.titech.ac.jp satos@ict.e.titech.ac.jp atsushi@ict.e.titech.ac.jp

Abstract— With the advances of technology nodes, a defective circuit pattern has occurred on a chip. Local regions on a mask that may cause defects such as opens/shorts are called hotspots, which induce yield loss, so they should be eliminated in the design phases. To detect hotspots, a conventional method extensively relies on lithography simulation, which can achieve good accuracy but may suffer from huge computational time. Recently, methods that introduce image recognition techniques are proposed. In this paper, we propose a hotspot detector based on probability distributions of layout features, where feature optimization and classification are guided by the distributions. Experimental results show that our proposed method achieves 98 % accuracy while False Positive Rate is less than 1 %, and its computation is 8 times faster than conventional machine learning based methods on ICCAD2012 benchmark suite.

I. Introduction

Lithography transfers a pattern on a photomask to an wafer. Photoresist coated on the wafer reacts with exposure light which passes through the photomask, and a circuit pattern is formed on the wafer.

In order to form a desired circuit pattern on an wafer, various design for manufacturing (DFM) techniques such as optical proximity correction (OPC) have been introduced. With the advances of technology nodes, defects such as opens/shorts may occur even after DFM techniques are applied. Local regions on a mask that may cause defects are called hotspots, which induce yield loss, so they should be eliminated in the design phases [1].

Exposure lights with shorter wavelength have been developed to improve the fidelity of pattern on an wafer. Although extreme ultra-violet (EUV) whose wavelength is 13.5 nm is being introduced as exposure light, it is still not mainstream in mass production because of huge manufacturing cost, and DFM techniques still have much attention.

Various hotspot detection methods have been introduced to detect hotspots. A local area on a mask, called clip, can be evaluated whether hotspot or not by lithography simulation. A good hotspot detection accuracy can be achieved by relying heavily on lithography simulation, but it would suffer from huge computational time [2]. A fast hotspot detector without accuracy loss is required to reduce manufacturing cost in practice.

Recently, several hotspot detection methods that filtered out non-hotspot clips without lithography simulation are proposed [3–15]. In these methods, a number of clips are filtered out in advance by classifiers, and the number of lithography simulations required is reduced. Fig. 1 shows a flow of such methods as well as a conventional method where all clips are evaluated by lithography simulation. Classifiers to filter out non-hotspot clips are often designed based on pattern matching or machine learning.

Pattern matching based methods [3, 4], which identify hotspots through comparing the geometric shape of layout patterns in a clip with that of the patterns in hotspot libraries, typically achieve good accuracy for known layout patterns, but not good for unknown ones.

Machine learning based methods [5–15], which construct a two-class classifier model using training data, determine hotspot/non-hotspot accurately even for unknown layout patterns. In this approach, the selection of layout features and learning algorithm affects accuracy and efficiency.

As existing layout features, density-based layout feature (DBLF) [5], which uses the ratio of the area occupied by wiring, histogram of oriented light propagation (HOLP) [7], which approximately quantifies the diffraction based on DBLF, and new features [10] focusing on the length between wires have been proposed.

As machine learning algorithms, support vector machine (SVM) [6], Boosting [7–10], Neural Network [11–15] have been applied.

Although existing methods combine some techniques such as clustering, multiple kernel learning (MKL) and data augmentation to achieve high accuracy, there is still room for improvement in terms of computational time and accuracy.

In this paper, we propose a hotspot detector based on the difference of the probability distributions of layout features between hotspot and non-hotspot clips. Our proposed detector consists of multiple weak classifiers and a strong classifier which follows them. The input of each weak classifier is a feature value derived from a clip, and the output is a real value which estimates the possibility

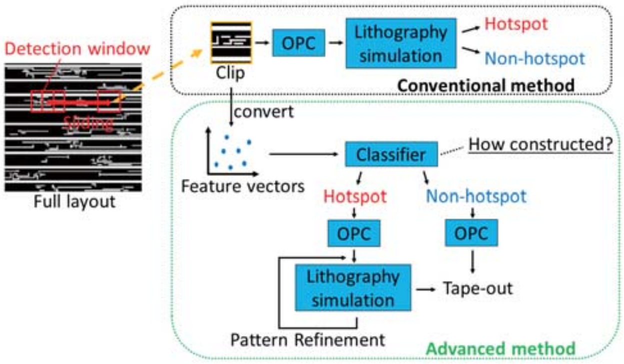

Fig. 1. Comparison of conventional and recent advanced methods. In advanced methods, a number of clips are filtered out in advance by classifiers, and the number of lithography simulations required is reduced.

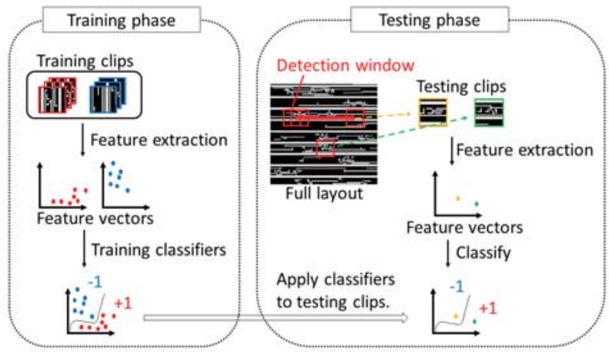

Fig. 2. The basic idea of machine learning based hotspot detection. Using known hotspot/non-hotspot clips given as training dataset, the classifier is trained. It is applied to testing clips extracted from a mask, and judge whether each clip is hotspot or not.

that the clip is a hotspot. If the sum of outputs exceeds the threshold, a strong classifier regards it as hotspot. Our key contributions are as follows:

- The feature optimization based on the probability distributions of layout features, which provides an efficient description of layout patterns in low-dimension, leads to good performance in terms of both accuracy and computational time.
- We propose a new hotspot detector, which consists of multiple weak classifiers defined based on the above feature optimization method.
- Our proposed detector achieves 98 % accuracy while False Positive Rate is less than 1 %, and its computation is 8 times faster than conventional machine learning based methods under ICCAD2012 benchmark suites.

The rest of the paper is organized as follows. In Section II, we define hotspot detection problem and describe machine learning based hotspot detection. In Section III, our proposed hotspot detector is described. Section IV presents the experimental results. We conclude this paper in Section V.

II. PRELIMINARIES

A. Hotspot detection problem

As mentioned in the introduction, with the development of technology nodes, defects such as opens/shorts may occur even after the process of DFM techniques. Local regions, which may cause these defects, are called hotspots, which induce yield loss, so they should be eliminated in the design phases.

The hotspot detection problem is to find hotspots from a mask layout (Fig. 1). A focused area of a mask is called a detection window. The detection window has the same size as the known hotspot/non-hotspot clips, and scans an entire mask to judge whether the area is hotspot or not every time.

B. Machine learning based method

Our goal is to correctly judge whether given clips are hotspot or not. In a machine learning based method (Fig. 2), the objective is to construct an efficient two-class classifier that can achieve high accuracy.

In a training phase, known hotspot/non-hotspot clips are given as training dataset. Features are extracted from each clip to represent the geometrical feature appropriately. The effective features and machine learning algorithms are combined to train the classifier.

In a testing phase, the trained classifier is applied to each clip extracted from a mask by a detection window scanning. It judges whether each clip is hotspot or not.

III. Proposed hotspot detector

We propose a hotspot detector based on the difference of the probability distributions of layout features between hotspot and non-hotspot clips. Our proposed detector, which consists of multiple weak classifiers and a strong classifier which follows them is shown in Fig. 4. In a training phase, multiple weak classifiers are defined based on feature optimization using given distributions of layout features. The distributions can be easily computed using training data. The input of each weak classifier is a feature value, and the output is a real value which estimates the possibility that the testing clip is hotspot. The output of a weak classifier is determined according to the difference between the probability of the feature value derived from hotspot clips and that from non-hotspot clips. If the sum of outputs exceeds the threshold, a strong classifier regards the testing clip as hotspot.

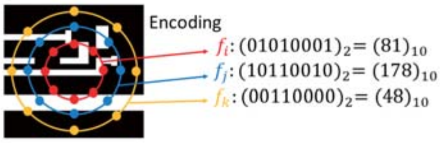

Fig. 3. The process of calculating layout features using CCAS. Each sampling circle is encoded to compact representation.

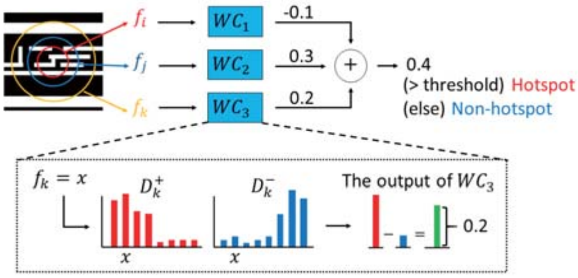

Fig. 4. The configuration of the hotspot detector is shown. In this example, the detector consists of three weak classifiers corresponding to the determined circles. The output of each weak classifier is determined according to the difference between the probability of the feature value derived from hotspot clips and that from non-hotspot clips. If the sum of outputs exceeds the threshold, an input clip is regarded as hotspot.

To calculate layout features, concentric circle area sampling (CCAS) [9, 16] is used. CCAS shows good results in hotspot detection [9] and building regression models to predict parameters in OPC [16]. CCAS is a sampling method, which intends that the diffraction light spreads concentrically. For a clip, circles concentrically are arranged from the center (Fig. 4). Each circle is uniformly sampled p points, every point is a binary number 0 or 1, where 1 represents that the corresponding point is contained in a layout pattern and 0 represents that the point contains no patterns. A bit sequence obtained from ith circle is a feature value to represent the geometrical feature.

In this chapter, the probability distributions of layout features, and the similarity of them are defined. Next, the process of training the detector and applying it is shown in detail.

A. *Definition*

Given hotspot clips $c_m^+ : m = 1, ..., M$ and the non-hotspot clips $c_n^- : n = 1, ..., N$ as training data. The subscripts of parameters $+$ and $-$ represent hotspot and non-hotspot, respectively. W^+, W^- are the set of training data weights which represent the clip that has the higher weight should be classified preferentially. X_i^+, X_i^- are the set of feature values obtained from the ith circle derived from training data. D_i^+, D_i^- are the set of probabilities that each feature value appears, which means the

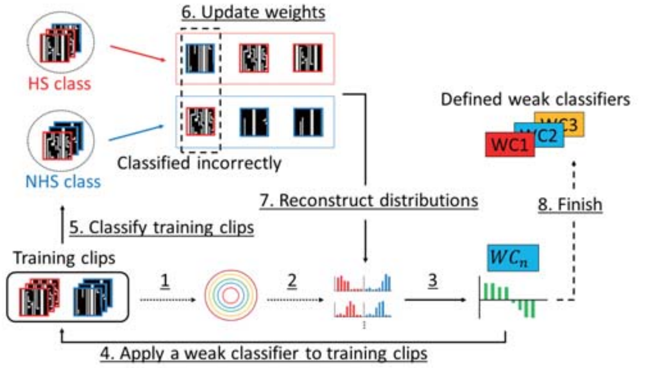

Fig. 5. The flow of training the detector is shown. The probability distributions can be easily computed using training data(flow.1-2). A weak classifier is defined based on feature optimization using them(flow.3). It's applied to training clips and classifies them(flow.4-5). By increasing the weights of clips classified incorrectly, distributions are reconstructed(flow.6-7). Flow.3-7 are repeated to define multiple weak classifiers.

probability distributions of layout features. $y \in \{-1, 1\}$ is training data label, where 1 represents hotspot and -1 represents non-hotspot.

- Parameters of hotspot clip
  - $W^+ = \{w_1^+, w_2^+, ..., w_m^+, ..., w_M^+\}$
  - $X_i^+ = \{x_1^+, ..., x_m^+, ..., x_M^+\}$
  - $D_i^+ = \{p_i(k|y = 1)|0 \leq k \leq 2^p - 1)\}$
- Parameters of non-hotspot clip
  - $W^- = \{w_1^-, w_2^-, ..., w_n^-, ..., w_N^-\}$
  - $X_i^- = \{x_1^-, ..., x_n^-, ..., x_N^-\}$
  - $D_i^- = \{p_i(l|y = -1)|0 \leq l \leq 2^p - 1)\}$

A.1. Probability distribution of feature vectors

An example of calculating D_i^+ is shown in Fig. 6. The procedure is shown below.

procedure Calculate possibility distribution
    for $m = 1$ to M **do**
        if $x_m^+ = k$ **then**
            $p_i(k|y = 1) += w_m^+$
        end if
    end for
    for $n = 1$ to N **do**
        if $x_n^- = l$ **then**
            $p_i(l|y = -1) += w_n^-$
        end if
    end for
    Normalize D_i^+ and D_i^-
end procedure

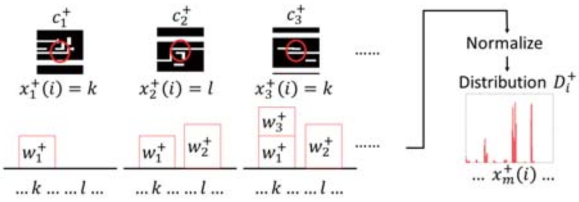

Fig. 6. The flow of calculating the probability distribution.

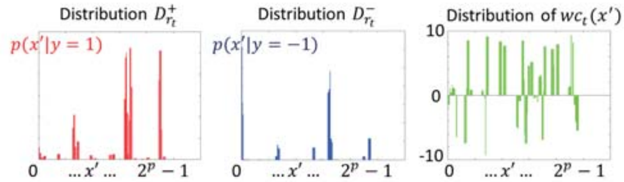

Fig. 7. Examples of D_i^+, D_i^- and a distribution of $wc_t(x')$ are shown. A weak classifier WC_t has a lookup table according to a distribution of $wc_t(x')$.

A.2. Similarity of the two distributions

Batacharya distance Z_i is used as a measure of how similar D_i^+, D_i^- are. The larger Z_i is, the more similar the shapes of the two distributions are. So, as Z_i is smaller, it can be judged as a feature that clearly classifies into hotspot and non-hotspot. This is defined by the following formula.

$$Z_i = \sum_{k=0}^{2^p-1} \sqrt{p_i(k|y=1)p_i(k|y=-1)} \tag{1}$$

B. *Training a detector*

As machine learning algorithms, Adaboost [17] [18] is used which is one of the machine learning algorithms and expected to be efficient. The proposed detector consists of multiple weak classifiers and a strong classifier which follows them. In this section, how to train the detector, in other words, how to select circles and define a weak classifier is stated. The flow to select circles is followed by the procedure below. Let R be the number of densely sampled circles, and the detector consists of T weak classifies, where T is set by users.

```
procedure TRAINING CLASSIFIERS
    for t = 1 to T do
        for i = 1 to R do
            Create D_i^+, D_i^-
        end for
        Find r_t, it is i when Z_i is minimum
        Define a weak classifier WC(r_t)
        Update all weights of training samples
    end for
end procedure
```

r_t is a radius when Z_i is minimum in round t. WC_t is a weak classifier, where the input is a feature value x' obtained from the circle with the radius of r_t, and the output $wc_t(x')$ is defined as following formula.

$$wc_t(x') = \ln \frac{p(x'|y=1)}{p(x'|y=-1)} \tag{2}$$

$wc_t(x')$ is the log likelihood ratio. The higher the value, the more likely the testing clip is a hotspot. Fig. 7 shows examples of D_i^+, D_i^- and a distribution of $wc_t(x')$.

In updating all weights step, the weights of training clips are updated. By increasing the weights of clips which cannot be classified correctly, they are classified preferentially by the weak classifier selected in next round. This step follows the formula below.

$$w_m^+ = w_m^+ \exp(-y_m h(x_m^+)) \tag{3}$$
$$w_n^- = w_n^- \exp(-y_n h(x_n^-)) \tag{4}$$

Here, let $y_m, y_n \in \{-1, 1\}$ be training data lebels, and let $h(.) \in \{-1, 1\}$ be a result of classification.

C. *Hotspot detection using our proposed detector*

Let wc_{all} be the sum of outputs of weak classifiers selected in the training step. If wc_{all} is larger than the threshold, the detector regards the testing clip as hotspot. On the other hand, if wc_{all} is smaller, it is regarded as non-hotspot. wc_{all} is formulated by the formula below.

$$wc_{all} = \sum_{t=1}^{T} wc_t(x') \tag{5}$$

IV. EXPERIMENTAL RESULTS

A. *Experimental Setup*

TABLE I
ICCAD2012 BENCHMARK STATICS

	Case1	Case2	Case3	Case4	Case5
Techonology	32nm	28nm	28nm	28nm	28nm
Training HS	99	174	909	95	26
Training NHS	340	5285	4643	4452	2716
Testing HS	226	498	1808	177	41
Testing NHS	319	4146	3541	3386	2111

We implement our proposed detector in C++ programming languages, and test it on a machine with four core 4.2 GHz CPUs and 32 GB memory. The performance of the proposed detector is evaluated on ICCAD2012 benchmark suite [19], which is divided into five cases consisting of known hotspot/non-hotspot clips. TABLE I shows the benchmark details.

TABLE II
Comparison of the proposed method and existing machine learning based methods

	SPIE'15 [8]			ICCAD'16 [9]			Ours (#WCs=10)			Ours (#WCs=50)		
	CPU(s)	FP	Recall	CPU(s)	FP	Recall	CPU(s)	FP	Recall	CPU(s)	FP	Recall
Case1	7	0	100.0%	7	0	100.0%	0.5	0	98.7%	2.0	0	98.7%
Case2	351	0	98.6%	51	0	99.4%	6.7	19	97.0%	31.0	138	96.6%
Case3	297	0	97.2%	66	3	97.5%	10.0	2	98.6%	47.8	0	98.6%
Case4	170	1	87.0%	35	0	97.7%	3.4	11	96.1%	16.5	57	89.8%
Case5	69	0	92.9%	24	0	95.1%	2.1	0	100.0%	10.1	0	100.0%
Average	178.8	0	95.1%	36.6	0	97.9%	4.5	6	98.1%	21.5	39	96.7%

TABLE III
Confusion Matrix

	Hotspot	Non-Hotspot
Predicted as Hotspot	TP (True Positive)	FP (False Positive)
Predicted as Non-Hotspot	FN (False Negative)	TN (True Negative)

B. Comparison with other methods

To evaluate the effectiveness of the proposed detector, we first compare the hotspot detection results with two great hotspot detectors [8] [9] in TABLE II. SPIE'15 [8] uses DBLF and Adaboost algorithm. ICCAD'16 [9] uses CCAS and Smooth Boosting algorithm.

The evaluation indicators are **"CPU(s)"**,**"FP"** and **"Recall"** corresponding to runtime of model evaluation in second, false positive number and recall rate. Recall is the most important indicators, which represents how many hotspot clips can be detected without missing. If the mask is applied to chip manufacturing with hotspots remaining, it causes yield loss. Recall is described below referring to TABLE III

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

FP is the number of non-hotspot clips detected as hotspot incorrectly. A big FP causes an increase in the number of lithography simulation iterations. The key is to reduce FP while achieving high Recall.

In this experiment, two existing detectors and two proposed detectors, where one consists of 10 weak classifiers and the other consists of 50 weak classifiers are evaluated. We can see from TABLE II that our detector with 10 weak classifiers achieves similar Recall and its computation is 8 times faster comparing with the detector proposed in ICCAD'16 [9]. In Case5, 100 % Recall can be achieved. However, in other cases, Recall is the latter half of 90 %. Although insufficient, we obtain a good recognition accuracy in its way.

Comparing the detector with 10 weak classifies and the other with 50, the former has a better result in terms of all evaluation indicators. We want to propose a better detector by analyzing deeply, such as examining which circles the detectors use.

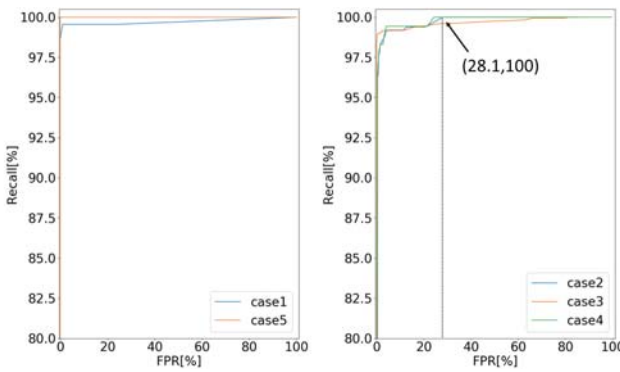

Fig. 8. ROC curve in case1 to case5. Recall and FPR are in the relationship of trade-off. The closer the curve is to the upper left, the better result it shows. In case2, if you lower the threshold until 100 % Recall is achieved, FPR will increase to 28 %.

C. Evaluation of Recall

This section shows how much false positive rate (FPR) does we compromise to achieve 100 % Recall. FPR represents how many non-hotspot clips are detected as hotspot incorrectly. It is described below referring to TABLE III

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

Recall and FPR are in the relationship of trade-off. Fig. 8 shows receiver operating characteristic (ROC) curves which graph the transition of Recall and FPR while changing the threshold. In case1, 100 % Recall cannot be achieved. In case2, if you lower the threshold until 100 % Recall is achieved, FPR will increase to 28 %. In case3 and case4, FPR will increase to 82 % and 24 %. We need to analyze layout clips which cannot be classified correctly.

V. CONCLUSION

In this paper, we propose the hotspot detector based on the difference of the probability distributions of feature vectors, where feature optimization and classification are guided by the distribution. Experimental results show that our proposed method achieves 98 % accuracy while False Positive Rate is less than 1 %, and its computation is 8 times faster than conventional machine learning based methods on ICCAD2012 benchmark suite.

Our future works include further experiments to analyze layout clips which cannot be classified correctly, and which circles the detectors use. In addition, it will be required to create more reliable benchmarks to ICCAD2012 benchmark suite.

References

[1] A. Takahashi, S. Sato, H. Ogura, Y. Sung, and T. Wang. Pattern similarity metrics for layout pattern classification and their validity analysis by lithographic responses. In *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 494–497, July 2018.

[2] Youngkeun Kim Joung Il Park Seung weon Paek Kee-sup Kim Mark C. Simmons, Jae-Hyun Kang. A state-of-the-art hotspot recognition system for full chip verification with lithographic simulation. In *Proceedings of SPIE*, volume 7974, 2011.

[3] Sheng-Yuan Lin, Jing-Yi Chen, Jin-Cheng Li, Wan-yu Wen, and Shih-Chieh Chang. A novel fuzzy matching model for lithography hotspot detection. In *ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2013.

[4] W. Wen, J. Li, S. Lin, J. Chen, and S. Chang. A fuzzy-matching model with grid reduction for lithography hotspot detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 33(11):1671–1680, Nov 2014.

[5] J. Wuu, F. G. Pikus, A. Torres, and M. Marek-Sadowska. Rapid layout pattern classification. In *16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011)*, pages 781–786, Jan 2011.

[6] Duo Ding Xuan Zeng David Z. Pan Bei Yu, Jhih-Rong Gao. Accurate lithography hotspot detection based on principal component analysis-support vector machine classifier with hierarchical data clustering. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 14(1):1–12, 2014.

[7] Y. Tomioka, T. Matsunawa, C. Kodama, and S. Nojima. Lithography hotspot detection by two-stage cascade classifier using histogram of oriented light propagation. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 81–86, Jan 2017.

[8] Tetsuaki Matsunawa, Jhih-Rong Gao, Bei Yu, and David Z. Pan. A new lithography hotspot detection framework based on adaboost classifier and simplified feature extraction. In *Proceedings of SPIE*, volume 9427, 2015.

[9] H. Zhang, Bei Yu, and E. F. Y. Young. Enabling online learning in lithography hotspot detection with information-theoretic feature optimization. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, Nov 2016.

[10] G. Kataoka, M. Inagi, S. Nagayama, and S. Wakabayashi. Novel feature vectors considering distances between wires for lithography hotspot detection. In *2018 21st Euromicro Conference on Digital System Design (DSD)*, pages 85–90, Aug 2018.

[11] Y. Yu, G. Lin, I. H. Jiang, and C. Chiang. Machine-learning-based hotspot detection using topological classification and critical feature extraction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 34(3):460–470, March 2015.

[12] Moojoon Shin and Jee-Hyong Lee. Accurate lithography hotspot detection using deep convolutional neural networks. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 15(4):1–13, 2016.

[13] Haoyu Yang, Luyang Luo, Jing Su, Chenxi Lin, and Bei Yu. Imbalance aware lithography hotspot detection: a deep learning approach. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 16(3):1–13, 2017.

[14] H. Yang, Y. Lin, B. Yu, and E. F. Y. Young. Lithography hotspot detection: From shallow to deep learning. In *2017 30th IEEE International System-on-Chip Conference (SOCC)*, pages 233–238, Sep. 2017.

[15] Haoyu Yang, Jing Su, Yi Zou, B. Yu, and E. F. Y. Young. Layout hotspot detection with feature tensor generation and deep biased learning. In *ACM/IEEE Design Automation Conference (DAC)*, pages 62:1–62:6, June 2017.

[16] Tetsuaki Matsunawa, Bei Yu, and David Z. Pan. Optical proximity correction with hierarchical bayes model. In *Proceedings of SPIE*, volume 9426, 2015.

[17] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[18] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336, dec 1999.

[19] J. A. Torres. ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 349–350, Nov 2012.