2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT) | 979-8-3503-3416-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/VLSI-TSA/VLSI-DAT57221.2023.10133968

Layout Hotspot Pattern Clustering Using a Density-based Approach

Ciao-Syun Lin[1], Pin-Yian Tsai[2], Yan-Hsiu Liu[2], Yi-Ting Li[1], Yung-Chih Chen[3], and Chun-Yao Wang[1]

[1]National Tsing Hua University, Taiwan, R.O.C.
[2]United Microelectronics Corporation, Taiwan, R.O.C.
[3]National Taiwan University of Science and Technology, Taiwan, R.O.C.

Abstract— Since the number of hotspot patterns detected on a layout using machine learning technique is very large, it takes designers a lot of time to classify these hotspot patterns for subsequent modification. These hotspot patterns are diverse and complex in shape. Therefore, we propose a density-based hotspot pattern clustering approach to classify these hotspot patterns into groups, which extracts the density feature of hotspot patterns while considering the shifted and distorted polygons on hotspot patterns. Experimental results show that our approach can classify the hotspot patterns more efficiently than SIFT method with similar results in each group.

I. Introduction

With the advance of integrated circuit fabrication, the feature size of transistors becomes smaller such that the layout is more complex. Although the full layout of a design has passed the design rule check (DRC) prior to manufacturing, the fabricated chip may be still deformed in some regions. These distortions, which are called hotspot patterns of a layout, may even become yield killers to the manufacturing process. The root causes of hotspot patterns on the layout are quite complicated and hotspot patterns seem to occur randomly or irregularly. However, these still exist certain similarities implicitly among hotspot patterns. Since it is difficult to find out the rules of the hotspot patterns manually, many previous works [6][8][10][12][13] proposed methods to detect hotspot patterns.

Hotspot pattern identification detects specific patterns that are inappropriate to the manufacturing process on the layout. However, the hotspot patterns on a layout are various and the amount of them is enormous such that dealing with each hotspot pattern is troublesome for designers. Thus, it is better to classify these hotspot patterns into different groups before dealing with each group of hotspot patterns.

Some previous works [3][4] focused on finding repeated patterns or similar patterns on a layout. The repeated patterns are the same patterns that appear repeatedly on the layout, whereas the similar patterns occur in two situations. The first situation is that the polygons of a pattern distort in shape. The second situation is that the polygons of a pattern shift in location. But the previous work [3] assumes these two situations cannot occur simultaneously, though polygon shapes and locations are usually changed at the same time. For example in Fig. 1, six hotspot patterns are similar and clustered into the same group. If we limit that polygons of the hotspot patterns in each group cannot be distorted and shifted simultaneously, these similar hotspot patterns will not be clustered together.

The previous work [14] proposed a method to solve this classification problem. It used the SIFT [9] algorithm to extract important characteristics of each hotspot pattern. Similar hotspot patterns with distorted or shifted polygons turn out to have similar characteristics. However, the method needed a large amount of time to classify the hotspot patterns.

In this work, we propose a density-based approach, which efficiently classifies similar hotspot patterns into a group. To classify these hotspot patterns, we extract the density feature of each hotspot pattern. Specifically, we cut an image into grids and extract the density of each grid. Experimental results show that our approach classifies hotspot patterns into several groups more efficiently with similar results.

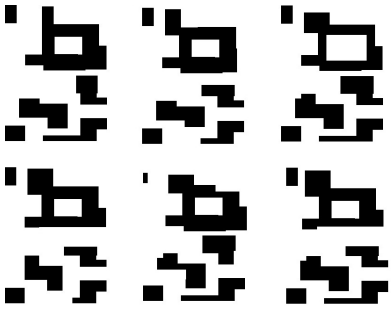

Fig. 1. Similar hotspot patterns in a group.

II. Background

A. Hotspot Pattern

In the manufacturing process, hotspot patterns are those easily deformed patterns on a layout. In this work, all the hotspot patterns have been converted to grayscale images as shown in Fig. 1. A hotspot pattern has $n \times n$ pixels, where n is the width and height of a hotspot pattern. The intensity of each pixel is either 0 or 255, which means that only black (0) or white (255) are on the hotspot pattern. The black regions on an image represent polygons of the hotspot pattern.

B. Density of a Grid

A hotspot pattern is cut into pieces with $m \times m$ grids, where m is an integer greater than one. The density is calculated as the area ratio of the black region and a grid. Thus, we can obtain a density feature matrix with m^2 elements by calculating area ratios of all the grids.

For example, a given hotspot pattern is shown in Fig. 2(a). We cut the hotspot pattern by imposing 4×4 grids as shown in Fig. 2(b). Then, we calculate the area ratio in each grid. The 4×4 density feature matrix with 16 elements is shown in Fig. 2(c), which can be flattened from left to right and top to bottom as a density feature vector as shown in Fig. 2(d).

By calculating the density of each grid, we have the location information of polygons on each hotspot pattern. The precision of this information depends on the size of the grid. The smaller the grid size is, the closer it will be to the origin hotspot pattern.

C. Center of Gravity of a Polygon

Since two similar hotspot patterns may contain the same but shifted polygons, we cannot directly use their density feature vectors for comparison. We propose to align two hotspot patterns with respect to their centers of gravity before calculating their density feature vectors. Hence, we first introduce how to compute the center of gravity of a polygon in this subsection. The center of gravity is related to the concept of image moments [7].

As in a grayscale polygon, the moment of order $(p+q)^{th}$ of a polygon can be written as:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y)$$

This work is supported in part by the Ministry of Science and Technology of Taiwan under MOST 109-2221-E-007-082-MY2, MOST 109-2221-E-155-047-MY2, MOST 110-2224-E-007-007, MOST 111-2218-E-007-010, MOST 111-2221-E-007-121, MOST 111-2221-E-011-137-MY3, and UMC: I-2021-08-198.

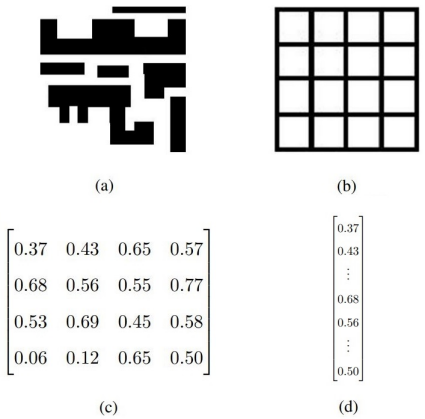

Fig. 2. Extraction of the density feature. (a) A grayscale hotspot pattern. (b) 4×4 grids. (c) The density feature matrix with 4^2 elements. (d) The flattened density feature vector with 4^2 elements.

where $f(x, y)$ is the pixel's intensity in the (x, y) coordinate of a polygon on a hotspot pattern. When $p = q = 0$, the equation becomes

$$m_{00} = \sum_x \sum_y f(x, y)$$

which accumulates the intensity of each pixel of a polygon.

For example, given a grayscale hotspot pattern as shown in Fig. 3(a). Since the intensities of the polygons are all 0s (black), we use a bitwise-NOT operations to exchange the intensity of black (0) and white (255) region of the hotspot pattern. The color-exchanged hotspot pattern is shown in Fig. 3(b). Note that this exchange is just for calculating the center of gravity. For example, for the largest polygon in Fig. 3(c), its total intensity m_{00} is calculated as 13064×255, where 13064 can be interpreted as its area.

The coordinates (x_c, y_c) of the center of gravity for a polygon in a color-exchanged hotspot pattern can be calculated by using m_{00}, m_{10}, and m_{01} as follows:

$$x_c = \frac{\sum_x \sum_y x f(x, y)}{\sum_x \sum_y f(x, y)} = \frac{m_{10}}{m_{00}}$$

$$y_c = \frac{\sum_x \sum_y y f(x, y)}{\sum_x \sum_y f(x, y)} = \frac{m_{01}}{m_{00}}$$

In this example, the coordinate of the center of gravity for the largest polygon is $(128, 81)$, which is shown in Fig. 3(d).

III. Proposed Approach

A. Density Feature Vector of a Hotspot Pattern

Using the method mentioned in the last section, we can obtain the center of gravity of each polygon on a hotspot pattern. We next compute the center of gravity of the hotspot pattern, which is used to align different hotspot patterns.

By using the following two equations, we can obtain the coordinate (H_x, H_y) of center of gravity of a hotspot pattern.

$$H_x = \frac{\sum_p x_c m_{00}}{\sum_p m_{00}}$$

$$H_y = \frac{\sum_p y_c m_{00}}{\sum_p m_{00}}$$

where p represents a polygon on the hotspot pattern, and m_{00} is the total intensity of a polygon p.

For example, after obtaining the center of gravity of each polygon on the hotspot pattern as shown in Fig. 4(a), we calculate the center of gravity of the hotspot pattern. The $(m_{00}, (x_c, y_c))$ of the six polygons on the hotspot pattern is: $(912 \times 255, (12, 19))$, $(13064 \times 255, (128, 81))$, $(3715 \times 255, (169, 179))$, $(4770 \times 255, (82, 226))$, $(1344 \times 255, (21, 274))$, $(3019 \times 255, (166, 269))$. By using the above two equations, H_x equals $\frac{255\times((912\times12+13064\times128+3715\times169+4770\times82+1344\times21+3019\times166))}{255\times(912+13064+3715+4770+1344+3019)}$, and H_y equals $\frac{255\times((912\times19+13064\times81+3715\times179+4770\times226+1344\times274+3019\times269))}{255\times(912+13064+3715+4770+1344+3019)}$. We round the H_x and H_y to obtain an integer coordinate. Hence, the center of gravity of the hotspot pattern is $(120, 149)$ as shown in the original hotspot pattern of Fig. 4(b).

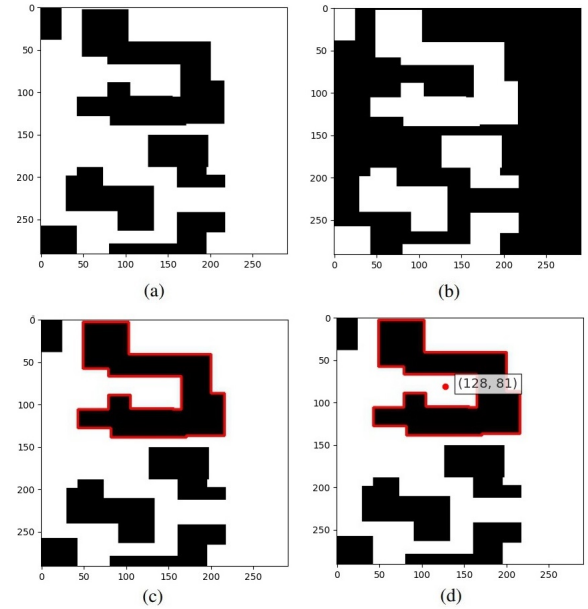

Fig. 3. The center of gravity of the hotspot pattern. (a) A 300×300 grayscale hotspot pattern. (b) The color-exchanged hotspot pattern. (c) The area of the largest polygon is 13064. (d) The point $(128, 81)$ is the center of gravity of the largest polygon.

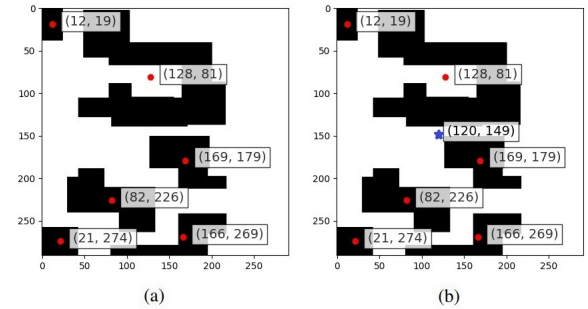

Fig. 4. Center of gravity of the hotspot pattern.(a) The center of gravity of each polygon on a hotspot pattern. (b) The star point $(120, 149)$ is the center of gravity for the hotspot pattern.

Then, we compute the density feature vector of a hotspot pattern with respect to its center of gravity. This is because the center of gravity of a hotspot pattern is not always at the center of hotspot pattern. We pad the hotspot pattern with white background for extending the boundary of the hotspot pattern, and impose the $m \times m$ grids on it with respect to its center of gravity. Hence, we can obtain the density feature vector of the hotspot pattern, which considers the effect of polygon shift on a hotspot pattern.

For example, given a hotspot pattern with its center of gravity in Fig. 5(a), we pad the hotspot pattern with white background as shown in Fig. 5(b). Then, we can obtain the density feature vector of the hotspot pattern from Fig. 5(c).

Next, we use an example to demonstrate the effectiveness of the proposed approach about the center of gravity alignment. Assume that we have three hotspot patterns in Fig. 6. Figs. 6(a) and 6(b) are similar hotspot patterns with slightly polygon shift and

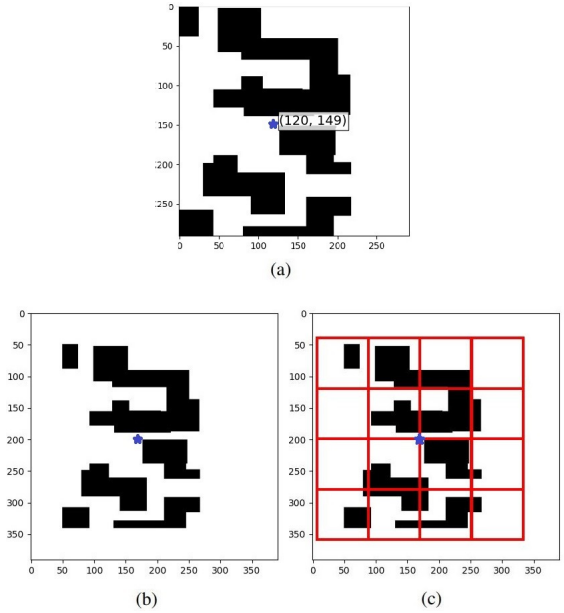

Fig. 5. Density feature vector computation. (a) The star point is the coordinate of center of gravity of a hotspot pattern. (b) The padded hotspot pattern with its center of gravity. (c) Imposing $m \times m$ grids for computing the density feature vector.

distortion. Figs. 7(a), 7(b) and 7(c) show their density feature vectors using 4×4 grids without considering their centers of gravity. We calculate the difference of two density feature vectors pairwise by accumulating the difference in each dimension as follows:

$$\sum_{i=1}^{n} |A_i - B_i|$$

where A_i and B_i are the i^{th} element of two density feature vectors A, B with n dimensions.

If we directly compare these density feature vectors pairwise, their differences between (a)&(b), (a)&(c), and (b)&(c) are 2.97, 2.78, and 3.53, respectively, which indicate that Figs. 6(a) and 6(c) are more similar. Figs. 7(d), 7(e) and 7(f) show the density feature vectors with considering the center of gravity of the hotspot pattern instead. If we compare these density feature vectors pairwise, their differences between (d)&(e), (d)&(f), and (e)&(f) are 0.06, 3.38, and 3.34, respectively. Although Figs. 6(a) and 6(b) have distorted and shifted polygons, our approach successfully identifies that they are similar hotspot patterns.

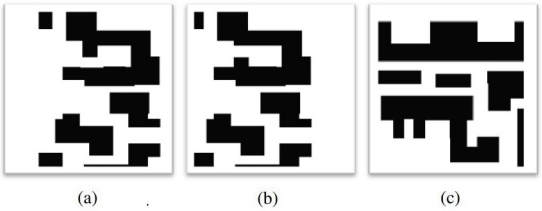

Fig. 6. Three hotspot patterns.

B. Rotational Hotspot Pattern Clustering

Some existing hotspot patterns that are similar to other's after rotation should be clustered in the same group as well. However, the density feature vectors obtained from last subsection naturally contains the location information of polygons on a hotspot pattern, which leads to the rotational hotspot patterns have quite different density feature vectors. Thus, we propose a method to deal with this issue in this subsection. We first obtain four hotspot patterns from the original hotspot pattern H_1 by rotating 90°, 180°, 270°, and

(a)	(b)	(c)	(d)	(e)	(f)
0.05	0.29	0.15	0.15	0.15	0.12
0.46	0.52	0.27	0.58	0.58	0.21
0.39	0.21	0.30	0.34	0.34	0.23
0.12	0.00	0.13	0.00	0.00	0.09
0.00	0.08	0.47	0.00	0.00	0.51
0.21	0.50	0.60	0.42	0.42	0.64
0.54	0.66	0.61	0.63	0.62	0.64
0.49	0.00	0.47	0.14	0.14	0.45
0.00	0.18	0.41	0.05	0.05	0.38
0.26	0.29	0.65	0.32	0.34	0.64
0.38	0.50	0.45	0.60	0.60	0.47
0.32	0.00	0.29	0.04	0.04	0.33
0.05	0.31	0.02	0.23	0.22	0.05
0.34	0.45	0.07	0.36	0.34	0.1
0.46	0.38	0.55	0.44	0.44	0.58
0.29	0.00	0.14	0.07	0.07	0.15

Fig. 7. The density feature vectors of Fig. 6 w/wo considering their centers of gravity.

360°, which can be represented by four density feature vectors. As a result, a hotspot pattern has four corresponding density feature vectors. When we have another hotspot pattern H_2, which is the hotspot pattern of H_1, the four density feature vectors of H_2 are one-to-one mapping to the four density feature vectors of H_1. Hence, we can use an $m \times 1$ vector, where $m = 4 \times k$, and k is the number of groups in the classification process, for recording the mapping of density feature vectors of a hotspot pattern.

For example, assume that we have three hotspot patterns as shown in Fig. 8(a). We observe that the first two hotspot patterns are hotspot patterns to each other. We rotate these hotspot patterns 90°, 180°, 270°, and 360°as shown in Fig. 8(b). A hotspot pattern corresponds to the four density feature vectors. Assume that we want to classify these three hotspot patterns into $k = 2$ groups, then $m = 4 \times k = 8$, which means that we use an 8×1 vector to record the mapping as shown in Fig. 8(c). As a result, the rotational hotspot patterns have the same 8×1 vector such that they will be put in one group in the process later.

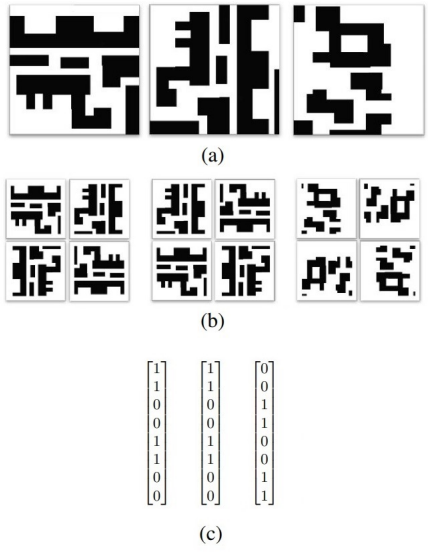

Fig. 8. Rotational Hotspot Pattern Clustering. (a) Three hotspot patterns. (b) Three hotspot patterns by rotating 90°, 180°, 270°, and 360°. (c) The 8×1 vector for recording the mapping.

We obtain the $m \times 1$ vector representing each hotspot pattern using the rotational hotspot pattern clustering. Then, the K-means algorithm clusters the $m \times 1$ vectors of all the hotspot patterns into k groups. After that, the hotspot pattern in a group that has the minimum difference among its $m \times 1$ vector and other hotspot patterns' $m \times 1$ vectors in the group is selected as the representative hotspot pattern.

For example, given three hotspot patterns in Fig. 8(a) and their

8×1 vectors as shown in Fig. 8(c). We use the 8×1 vector to represent each hotspot pattern and cluster the three hotspot patterns into two groups. The first two hotspot patterns will be clustered to the same group and the last hotspot pattern will be in the other group.

C. Overall Flow

Fig. 9 shows the overall flow of the proposed approach. We load given hotspot patterns and compute the center of gravity of each hotspot pattern to align all hotspot patterns. We obtain the density feature vector of each hotspot pattern with their centers of gravity. In the hotspot pattern clustering, we rotate the hotspot pattern 90°, 180°, 270°, and 360°and obtain the four corresponding density feature vectors. We then record the one-to-one mapping of the four density feature vectors in the $m \times 1$ vector. The $m \times 1$ vectors are used in the hotspot pattern clustering.

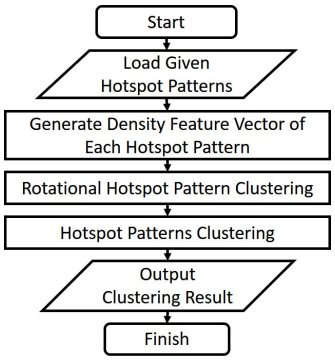

Fig. 9. Overall flow of the proposed layout hotspot pattern clustering.

IV. Experimental Results

The proposed density-based hotspot pattern clustering approach was implemented in Python language with Scikit-learn package and OpenCV. The benchmarks are two datasets from a semiconductor manufacturing company, which include necking1 and short2. The experiments are conducted on Linux 3.10.0 platform with Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz and 511GB RAM.

We cluster the dataset into 20, 40, 60, and 80 groups and accumulate the total differences of hotspot patterns in each group in our experiments. Considering that a group of hotspot patterns will contain rotated hotspot patterns, we rotate the representative hotspot pattern of each group 90°, 180°and 270°, and 360°as shown in Fig. 10.

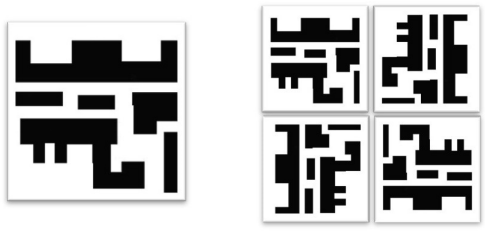

Fig. 10. The representative hotspot pattern of a group and the four rotated representative hotspot patterns of a group. (a) A representative hotspot pattern of a group. (b) Four representative hotspot patterns of a group.

The total differences in a group are calculated by accumulating the minimum difference between each hotspot pattern and its four rotated representative hotspot patterns.

The minimum difference between two hotspot patterns can be obtained by overlapping two hotspot patterns from the top left to the bottom right. During the movement for overlapping, the difference of two hotspot patterns can be calculated by XOR operation between the hotspot pattern and the representative hotspot pattern. The minimum difference between two hotspot patterns is the minimum difference obtained in the overlapping process.

Necking1 dataset has 9354 hotspot patterns and short2 dataset has 12998 hotspot patterns. The total differences and the CPU time of clustering these datasets are shown in TABLEs. I $\sim$ II.

The experimental results show that our approach is more efficient than [14], and the total differences are very close to [14].

TABLE I
Classification results of necking1 dataset.

#groups	Total Differences (%)			CPU (s)		
	[14]	Ours	Ours - [14]	[14]	Ours	Speedup
20	17.21	20.30	3.09	1356.57	179.41	7.56
40	16.87	18.28	1.92	1236.83	310.91	3.98
60	15.62	17.06	1.59	1938.35	410.64	4.72
80	14.96	15.92	0.96	845	356.9	2.37

TABLE II
Classification results of short2 dataset.

#groups	Total Differences (%)			CPU (s)		
	[14]	Ours	Ours - [14]	[14]	Ours	Speedup
20	23.73	22.44	-1.29	1436.96	260.55	5.52
40	18.80	19.09	0.29	1612.42	334.54	4.82
60	15.87	17.08	1.21	1700.50	430.24	3.95
80	14.26	15.45	1.19	1829.55	468.02	3.91

V. Conclusion

In this paper, we propose an efficient density-based clustering approach, which classifies hotspot patterns using their density feature vectors. The proposed density feature extraction deals with the polygon shifted problem of similar hotspot patterns using the center of gravity. The experimental results show that our approach classifies the hotspot patterns more efficiently with similar results as compared to the previous work.

References

[1] Bradski, Gary. "The openCV library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25.11 (2000): 120-123.

[2] Buitinck et al., "API design for machine learning software: experiences from the scikit-learn project," *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013.

[3] Chang, Wei-Chun, et al.,"iClaire: A Fast and General Layout Pattern Classification Algorithm" *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017.

[4] Jingsong Chen, James Shiely, Evangeline F. Y. Young, "Fast detection of largest repeating layout pattern," *Design-Process-Technology Co-optimization for Manufacturability XIII.*, vol. 10962. SPIE, 2019.

[5] Charles Elkan, "Using the Triangle Inequality to Accelerate K-Means," *ICML'03: Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pp. 147-153, 2003.

[6] Tianyang Gai, Tong Qu, Xiaojing Su, Shuhan Wang, Lisong Dong, Libin Zhang, Rui Chen, Yajuan Su, Yayi Wei, Tianchun Ye, "Multi-level layout hotspot detection based on multi-classification with deep learning," *Proc. SPIE 11614, Design-Process-Technology Co-optimization XV, 116140W (22 February 2021)*; https://doi.org/10.1117/12.2583726.

[7] Gonzalez, R.C. and Woods, R.E., *Digital Image Processing.* Pearson Education. ISBN:9781292223070

[8] J. Jiang et al., "Reducing Systematic Defects using Calibre Wafer Defect Engineering and Machine Learning Solutions," *2020 International Workshop on Advanced Patterning Solutions (IWAPS)*, 2020, pp. 1-3, doi: 10.1109/IWAPS51164.2020.9286791.

[9] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150-1157 vol.2, doi: 10.1109/ICCV.1999.790410.

[10] Yuansheng Ma, Feng Wang, Qian Xie, Le Hong, Joerg Mellmann, Yuyang Sun, Shao Wen Gao, Sonal Singh, Panneerselvam Venkatachalam, James Word, "Machine learning based wafer defect detection," *Proc. SPIE 10962, Design-Process-Technology Co-optimization for Manufacturability XIII, 1096208 (20 March 2019)*); https://doi.org/10.1117/12.2513232.

[11] Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[12] H. -C. Shao et al., "From IC Layout to Die Photograph: A CNN-Based Data-Driven Approach," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 5, pp. 957-970, May 2021.

[13] W. Wen, J. Li, S. Lin, J. Chen and S. Chang, "A fuzzy-matching model with grid reduction for lithography hotspot detection.," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 11, pp. 1671-1680, Nov. 2014.

[14] An In-house tool in a semiconductor manufacturing company.