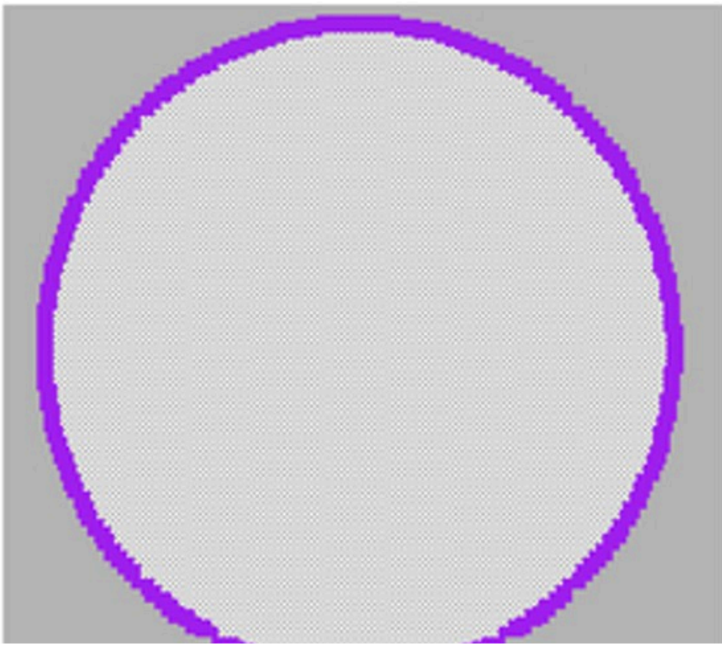
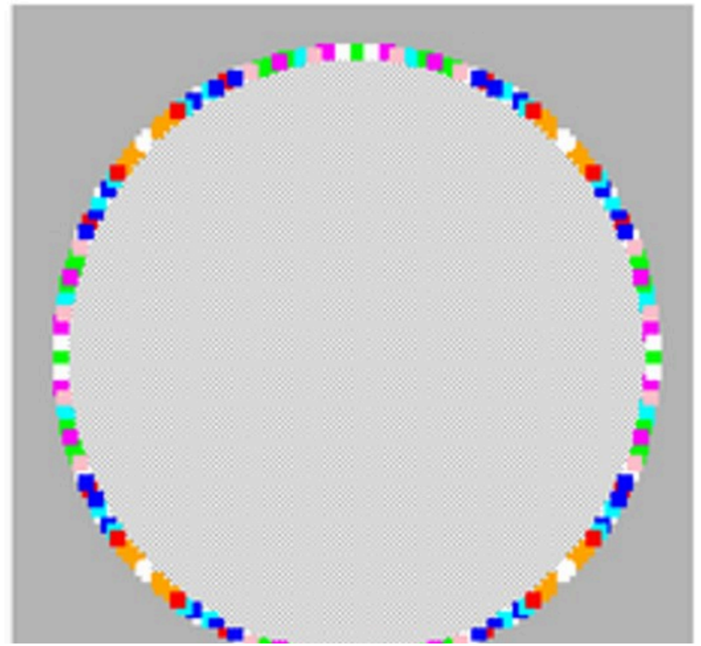




DIGITAL INDUSTRIES SOFTWARE

Machine Learning based error classification for curvilinear designs

Executive summary

Curvilinear design layout is becoming increasingly prevalent in the semiconductor manufacturing industry. One of the applications is silicon photonics. Silicon photonics design layout requires the use of curvilinear shapes to minimize the loss of signal strength.

Curvilinear design layouts pose new challenges to computational lithography tools that were developed mainly to handle Manhattan geometries. It has been found that the geometry-based error classification used in the optical proximity correction (OPC) verification flow has limitations in it's ability to support curvilinear designs. In this paper, we present our innovative work using Siemens EDA Calibre® OPCVerify Machine Learning (ML) Classify technology to classify error markers in the feature vector space instead of traditional pattern’s vertices and edges geometries.

There are five steps in our OPCVerify ML Classify flow:

1. Use primary checks to output error markers.
2. Design features and create feature vector kernels.

3. Collect feature vectors at error marker locations.
4. Prepare patterns with pre-knowledge of desired classify guidance and train the ML model.
5. Apply ML classify model to classify error markers on the full layout in the feature vector space.

In our experiments, ML Classify is successful in classifying OPC verification error markers in curvilinear designs. A drawn silicon photonics layout with 837,072 raw error locations has demonstrated our ML Classify tool's capability to reduce the unique class count from 221,085 - based on conventional geometry-based classify approach - down to 51. We also developed a feature that provides options to further sub-classify results by edge types, convex, concave, or straight line, and by polygon's internal width and external space to neighboring polygons. The 51 unique class count becomes 2493 after the further sub-classify process. This methodology is not only good for silicon photonics application, but also good for other curvilinear photomask applications, like CL MPCV, MEMS, on-chip metasurface optics, and in general even Manhattan designs.

Authors: Lianghong Yin , Shumay Shang , Fan Jiang , Le Hong , Robin Chia , Juli Opitz , Paul Adam , Ian Stobert and Yee Wee Koh

Contents

Introduction

Challenge in classifying locations in curvilinear design

ML-based classify in the feature vector space

Experimental setup and results

Comparisons between ML-based and geometry-based classify

Summary and conclusion

Acknowledgement

References

Introduction

Nowadays the complimentary-metal-oxide-semiconductor (CMOS) manufacturing industry has gone deep into the single digit technology node era. It's more and more challenging to develop the advanced technology nodes. The regular doubling of integration density in every two years appears to be slowing down. In the foreseeable future, there will be decreasing focus on shrinking the device size and increasing focus on the sequential introduction of increasingly diverse device technologies [1].

In the field of semiconductor manufacturing, from the breadth perspective of the future development, commercialized silicon photonics technology is a promising application. It is to make photonic and opto-electronic devices integrated onto a single microchip. To mass-produce the photonic and opto-electronic devices at low cost, the industry is leveraging the same manufacturing techniques as those used in traditional CMOS manufacturing processes. Silicon photonics devices are truly CMOS compatible in the form of silicon-on-insulator (SOI) waveguide. SOI waveguides are used to form modulators, switches, etc. photonics devices. They are also used to route light between photonics devices. This is possible because silicon is transparent in the wide spectral regions extending from near to mid infrared, especially importantly at the telecommunication wavelength windows from 1460 nm to 1565 nm. The refractive index of silicon can be modulated by electro-optic effects. These facts make it very promising for making passive and active opto-electronic devices [2–11]. Silicon Photonics design layouts require the use of curvilinear shapes to minimize the loss of signal strength. Some other wide applications of semiconductor manufacturing platform include MEMS and metasurface optics. They also require the use of curvilinear shapes. MEMS is an integrated system that combines mechanical and electrical components [12-14]. Metasurface optics is making optical system comprised of optical components such as lenses, splitters/couplers, waveplates, polarizers, etc., in a flat form in a thin layer on a substrate [15-17].

In the field of semiconductor manufacturing, from the depth perspective of the future development, there are trends that are leading to increased use of curvilinear designs in the mask data preparation (MDP) flow. For example, Inverse Lithography Techniques Optical Proximity Correction (ILT OPC) and multi-beam mask writers will increase the use of curvilinear design. As the industry pushes toward advanced nodes [18], ILT OPC is going to play a growing role in addressing challenges of photolithography. Based on the simulation results of Image Log Slope (ILS), Process Viability (PV) Band, and Depth of Focus (DOF), the benefit of using ILT is clear. Curvilinear result was demonstrated to be best in all different ILT results, although the magnitude is pattern dependent. To harvest the benefit of improved process window from ILT OPC, it is necessary to write mask in curvilinear shapes. During the MDP flow, for CL MPC, CL MPCV, and CL MRC, the input is as-if curvilinear design [19-21].

Curvilinear design layout is becoming increasingly prevalent in semiconductor manufacturing. The characteristics of curvilinear design include: 1. edges are in arbitrary angle, 2. vertex and edge counts are large, 3. possible consistency issue in snapping to nearest grid in computing tools. Because of these characteristics, curvilinear design layout poses new challenges to computational lithography tools that were developed mainly to handle Manhattan geometries.

A version of this paper was presented at the SPIE Advanced Lithography conference 2022 and published in the proceedings.

Challenge in classifying locations in curvilinear design

Background and problem statement

Many challenges in dealing with curvilinear designs have been discussed [18-22]. However, classifying the OPC verification checking result markers for reviewing purpose has not been discussed yet. In this paper, we describe the problem we have encountered at work and our development of a machine learning based classify solution in Calibre® OPCVerify ML Classify technology.

Each OPC verification checking result marker represents a location on the target layer. Classifying locations on a curvilinear target layer is challenging. One challenge is that some patterns placed at different X-Y coordinates may be classified differently even though they are the same pattern. This is due to the fact that the context target layer is curvilinear and hence prone to grid snapping inconsistency. Another challenge is that vertices and edges counts are large on a curvilinear layer, and there can be many small jogs. Hence two factors are important for classifying locations on a curvilinear layer. First, error markers should be anchored with context layer's vertices to be more resistant to grid snapping inconsistency. Second, classifier should allow some coarse match tolerance to be more resistant to small jogs on edges. The goal is to classify error categories so that very similar errors are grouped into a much smaller number of categories that can be reviewed. Error review cycle time can be improved with a smaller number of unique categories needed for review.

Geometry-based classify solution

We study the behavior of geometry-based basic classifier on curvilinear design. Consider a simple circle shape with radius 20 um, as shown in Figure 1. We densely put highlight markers around the edge and classify the markers using the circle as the context layer. These highlight markers could be generated by DRC checks, e.g., width or space check, or by ORC checks, e.g., pinch or bridge or Edge Placement Error (EPE) check. We use coloring scheme to represent classify results. Markers in different colors

are in different classes. Markers in the same color may be in the same class, with the reminder that because number of colors for use is limited and colors will be repeatedly used when all unique colors are used up. From a user's point of view, the expectation is that all the markers should be classified into the same class because a circle should be considered as one same pattern no matter where the detection location is on the edge. This is illustrated in Figure 1(a). We use a geometry-based basic classifier to classify the markers. Both anchoring to context layer's vertices and coarse match tolerance are used. The result is shown in Figure 1(b). We can see that markers that are symmetrically located along X or Y axis are classified into same class, while neighboring markers are classified into different classes. This means the behavior of the geometry-based basic classifier is edge-angle dependent. However, this does not satisfy the user's expectation in this specific circle pattern. This edge-angle dependency means a geometry-based basic classifier is not a good fit for curvilinear designs. We need to find a better solution to classify verification error markers for curvilinear design layouts. Ideally, users can control the classifier's behavior according to the specific use case.

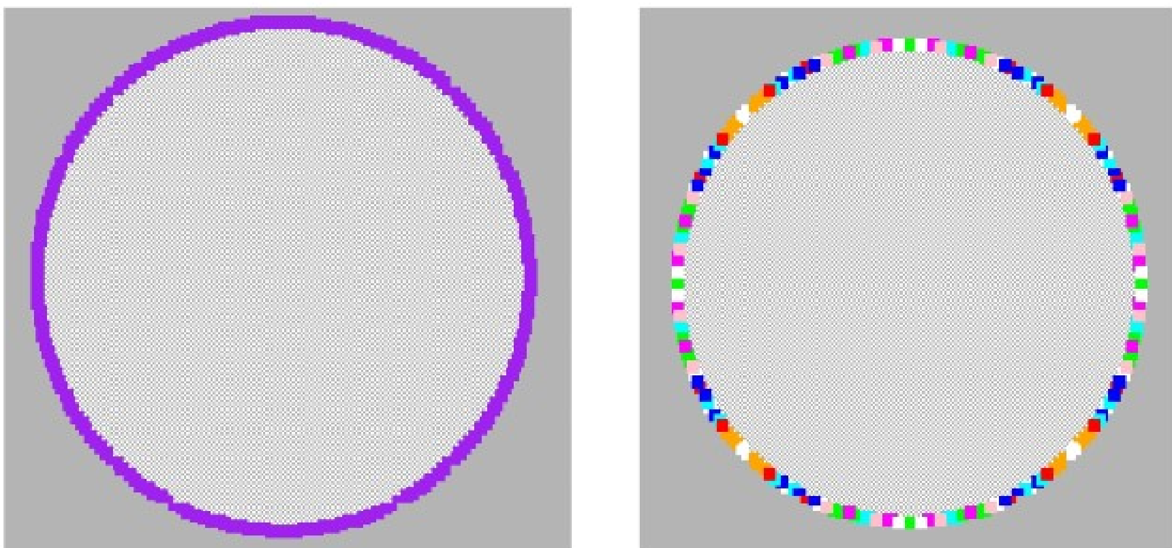


*Figure 1. A simple circle with radius 20 um, a typical pattern in curvilinear design layout. Highlight markers are densely placed on the edge. The markers are classified using different classifiers. The results are shown in (a) expectation of classifier results, (b) geometry-based basic classifier results. Coloring scheme is used to represent

classify results. Markers in different colors are in different classes. Markers in the same color may be in the same class, with the reminder that because number of colors for use is limited and colors will be repeatedly used when all unique colors are used up.

ML-based classify in the feature vector space

Calibre® ML platform

At Siemens EDA, our Calibre® ML platform is very powerful [23-26]. There are five modules as shown in Figure 2: ML Design, ML Process, ML Engine, ML Database, and ML Analytics. The ML Engine provides unsupervised, semi-supervised, and supervised ML model training. There are many applications in the ML Analytics module, including pattern reduction, pattern coverage, model coverage, layout comparison, hotspot prediction, defect classification. The OPCVerify ML classify application is using a small portion of our ML platform's capabilities. It collects geometry information of the context layer at each error marker location and performs semi-supervised classification.

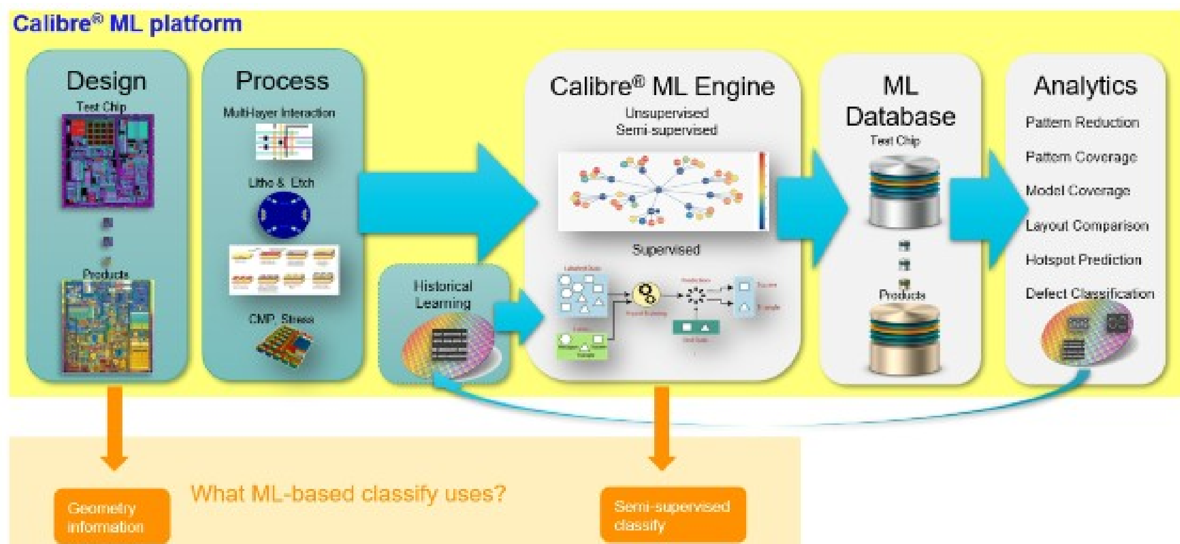


*Figure 2. Siemens EDA Calibre® ML platform. There are five modules: ML Design, ML Process, ML Engine, ML Database, and ML Analytics. The ML Engine provides unsupervised, semi-supervised, and supervised ML model training.

Steps in ML classify

There are five steps in our OPCVerify ML Classify, as shown in Figure 3:

1. Use primary checks to output error markers.
2. Design features and create feature vector kernels.
3. Collect feature vectors at error marker locations.
4. Prepare patterns with pre-knowledge of desired classify guidance and train ML model.
5. Apply ML classify model to classify error markers on the full layout in the feature vector space.

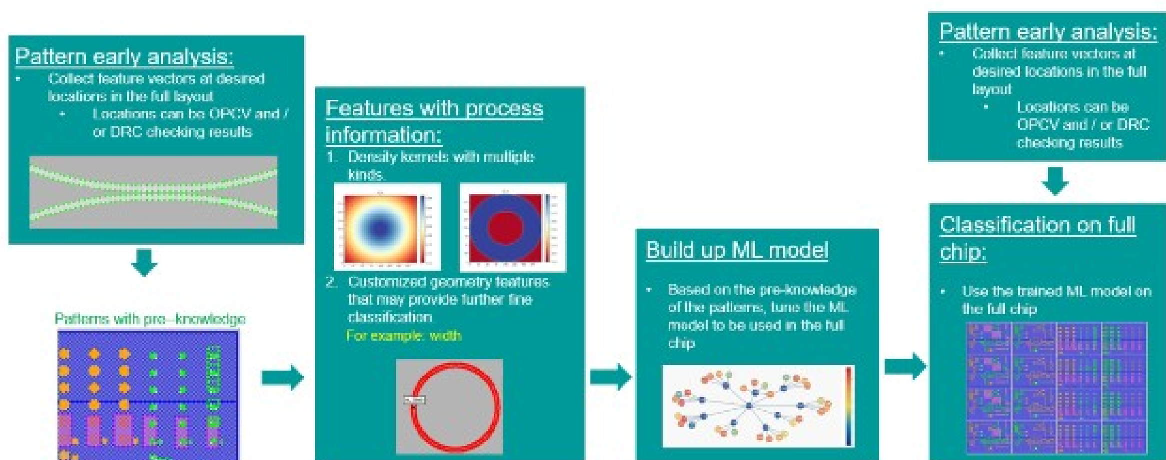


*Figure 3. Schematics of steps in ML classify.

Primary check

An OPCVerify primary check is used to prepare error markers on the layout. Any primary check can be used, for example, contour_diff, measure_epe, meefcheck, pinch, bridge, measure_distance, etc. Feature vector data are collected at the center of the error marker.

Design features and create feature vector kernels

Siemens EDA Calibre® OPCVerify ML Classify tool can classify patterns in the feature vector space. It can classify geometrically similar patterns and can also classify lithographically similar patterns. The success in classifying patterns relies on the proper design of features and creation of feature vector kernels. In the current task,

we focus on classifying geometrically similar patterns. So geometrical pattern density kernels are enough. In Calibre® ML Engine, we have gaussian, elliptical gaussian, tophat, ring, and their variant kernel designs to choose from, as shown in Figure 4. For the silicon photonics design that we are working on, the critical dimension is about 100 nm, halo size that we want to use for classification is 1 μm. We designed the density kernels accordingly.

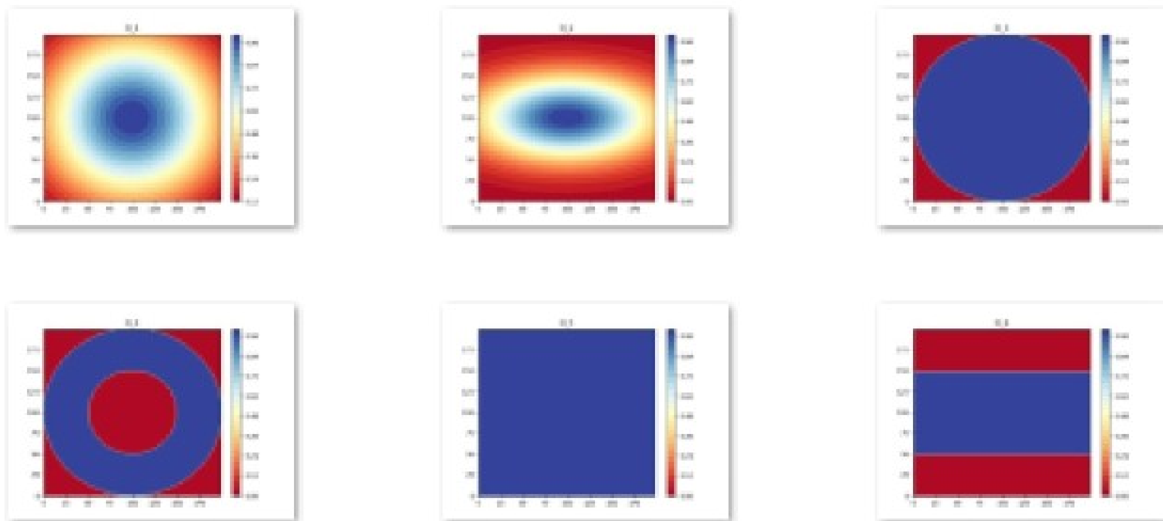


*Figure 4. Typical density kernels for feature vector data collection.

Collect feature vectors at error marker locations

After a primary check catches errors, feature vectors from the target layer are collected at the centers of the error markers using the feature vector kernels specified. Some examples of density kernels are given in Figure 4. All the feature vectors collected from a full chip comprise of the feature vector space.

Prepare patterns with pre-knowledge of desired classify guidance and train ML model

Our ML classify is semi-supervised. We provide pre-knowledge of desired classify guidance. For a simple example, we can provide two classify guidance rules.
Rule 1: Polygons with widths in large difference should be classified into different classes. This is illustrated in Figure 5(a).
Rule 2: Polygons with same width and similar pattern densities should be classified into same class. This is illustrated in Figure 5(b).

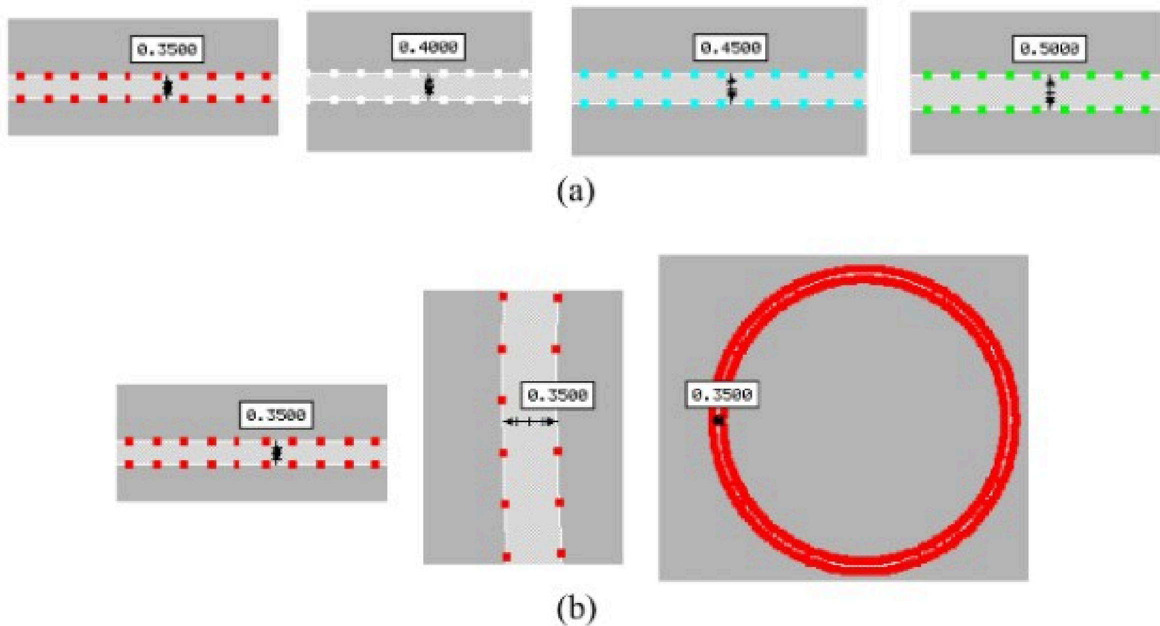


*Figure 5. Pre-knowledge of desired classify guidance rules. (a) Polygons with widths in large difference should be classified into different classes. (b) Polygons with same width and similar pattern densities should be classified into same class.

Before feature vector capturing, we tag some patterns that fall into the guidance rules. During feature vector capturing, we let the tool compute some geometrical properties and then label the tagged error markers based on the computed values.

After feature vectors are captured, we use the labeled feature vectors data points to train a semi-supervised ML classify model and save the model. This model can be trained once and then repeatedly used on future full chips.

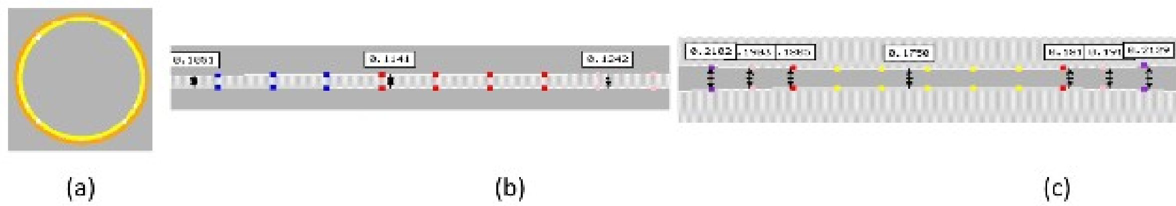
Classification on full chip

After we have trained the ML model, we apply it on the data of full chip.

Further sub-classify with user-input tolerance control

The ML classify is based on a coarse primary matching. We may want to further sub-classify the results with even finer matching based on user-input tolerance control. For example, a straight-line edge, a convex edge, and a concave edge may have the same pattern densities and are classified into the same class. We may want to put them into three different classes based on their edge types, as shown in Figure 6(a).

Another example, a taper line's width changes gradually. When the width change is within a tolerance limit, we put them into the same class. However, when the width change is beyond a tolerance limit, we want to put them into different classes, as shown in Figure 6(b). Still another example, a coupler's gap space changes gradually. When the space change is within a tolerance limit, we put them into the same class. However, when the space change is beyond a tolerance limit, we want to put them into different classes, as shown in Figure 6(c).



Experimental setup and results

Experimental layout

We drew a silicon photonics layout for the experiment. The idea of device layout is from reference [27], as shown in Figure 7.

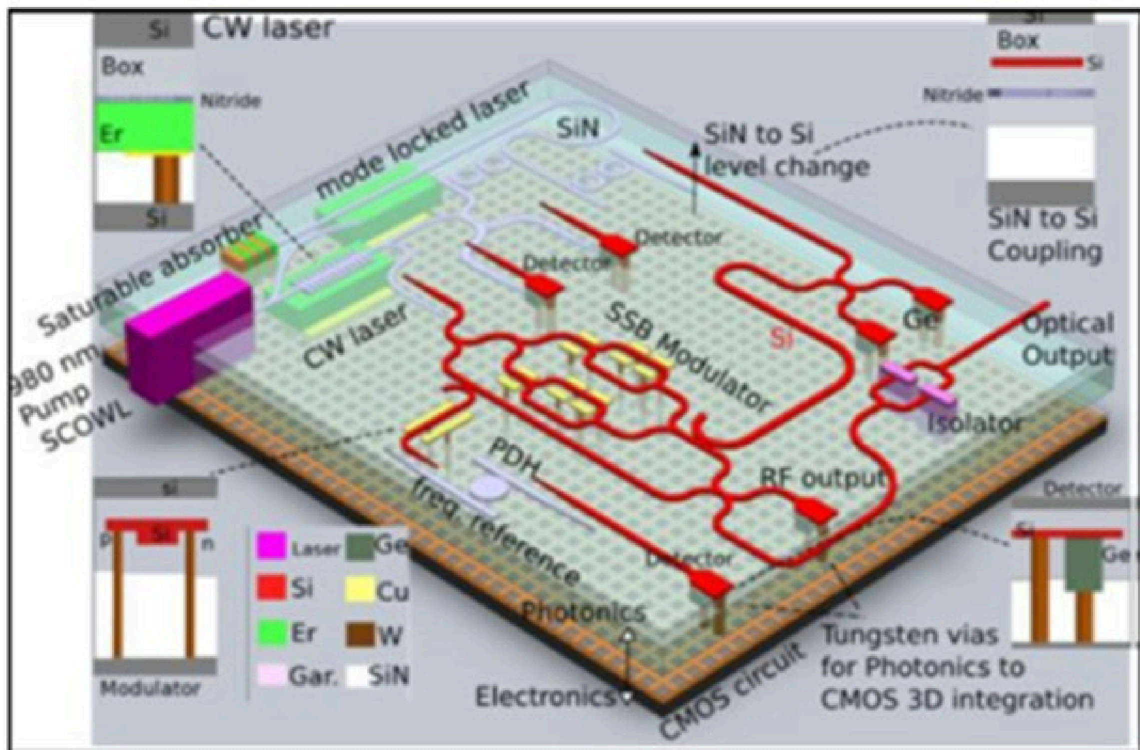


*Figure 7. Device layout is from reference [27] https://www.photonics.intec.ugent.be/download/pub_4128.pdf.

Our full layout size is 3.5 mm x 2.5 mm. We put in four variants of the silicon waveguide width: 350 nm, 400 nm, 450 nm, and 500 nm. We also drew circle shapes and concentric ring shapes. The radius, width, and space are modulated for better studying the classify results. The minimum width and minimum space in the design are both 100 nm. Our experimental full layout is shown in Figure 8.

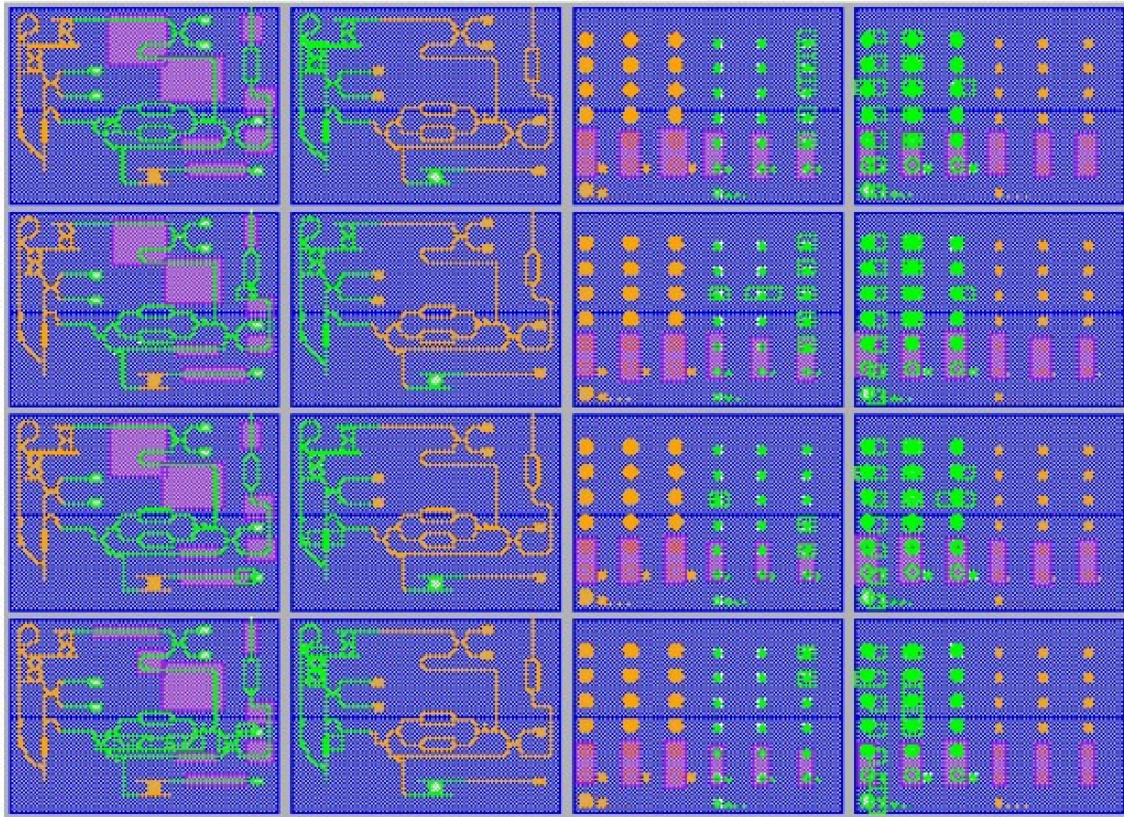


*Figure 8. Experimental full layout for ML classify study. Full layout size is 3.5 mm x 2.5 mm.

Experimental setup

Clalibre OPCVerify primary check measure_epe is used to output error markers. The constraint is absolutely loose and spacing is 400 nm, so that the epe value does not matter and there is an output error marker in every 400 nm along Silicon layer's edges. This setting is for testing the ML classifier's capability. The total error count is 837,072. Pattern density features are used for feature vector collection. We used purple marker layer to tag and label the error markers for ML model training. We compute curvature, polygon width and space at each error marker location for further sub-classify purpose.

ML classify results on full layout

The two classifying guidance rules, as shown in Figure 5, were used to train the ML classify model. After model training, the best model was saved and applied on the full layout. Total 837,072 error markers were classified into 51 unique classes. We can

define classify ratio as total error markers count divided by classified unique classes count. The classify ratio here is 16413. This is a desirable outcome, since it would be impractical to review over 800000 errors, while 51 canonical categories are much more manageable. In addition to the big improvement in classify ratio, the capability of differentiating unique patterns is not compromised, this has been verified through the pre-designed test patterns. The results are shown in Figure 9. The results show that the two guidance rules are carried out correctly on the full layout classified results: 1. silicon waveguides with width 350 nm, 400 nm, 450 nm, and 500 nm are put into different classes, 2. patterns with same width and similar pattern densities are put into same class, regardless the shape is straight line, or curved line, or ring.

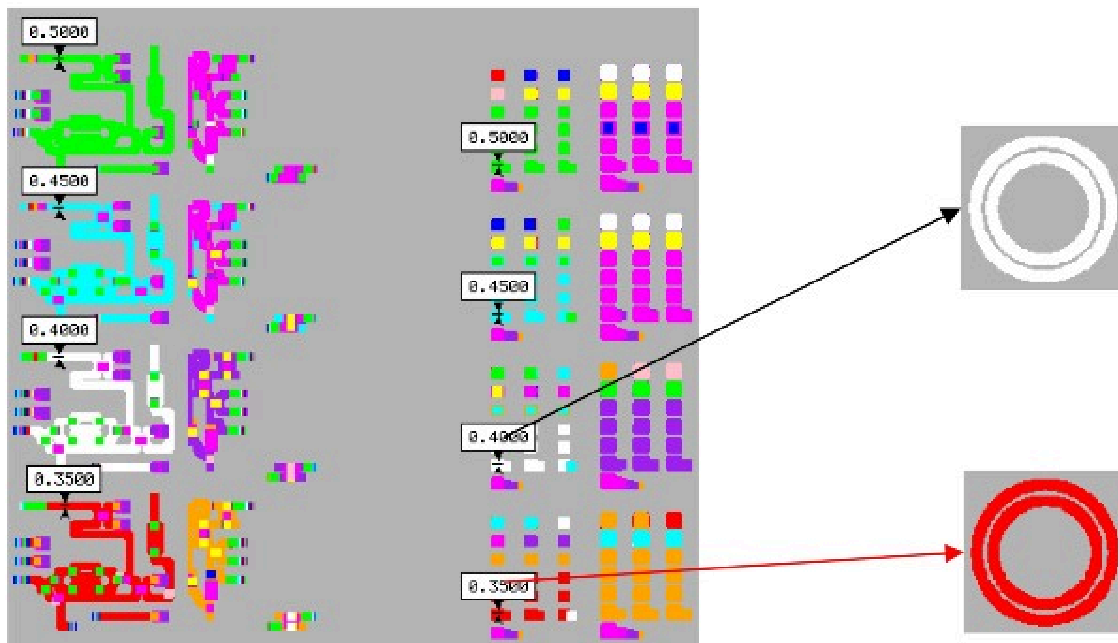


*Figure 9. ML classify results on full layout. Silicon waveguides with width 350 nm, 400 nm, 450 nm, and 500 nm are put into different classes. Patterns with same width and similar pattern densities are put into same class, regardless the shape is straight line, or curved line, or ring.

The results can also be reviewed in tree structure, as shown in Figure 10. The tree has two levels.

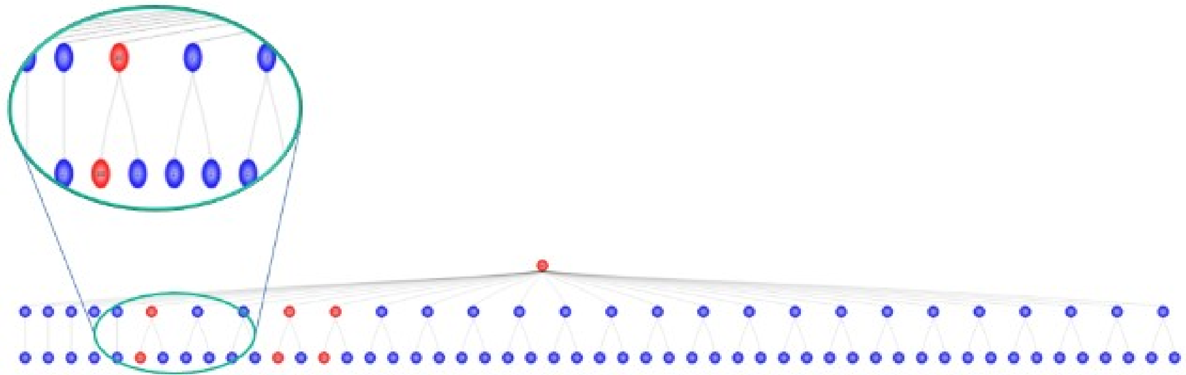


*Figure 10. Tree structure of the ML classify results of the full layout. The tree has two levels.

Further sub-classify results on full layout

After the total 837,072 error markers were classified into 51 unique classes, we can further sub-classify the results by edge type, polygon width, and gap space. The sub-classify of edge type is categorical, convex edge, concave edge, or straight-line edge. Tolerance of 10 nm is used to sub-classify the results by polygon width. Similarly, tolerance of 10 nm is used to sub-classify the results by gap space. After the further sub-classify, the 51 unique classes become 2493. The classify ratio is 336. The results are shown in Figure 11. This further sub-classify step is optional. User can choose from multiple features. This provides the flexibility and capability for user to further fine tune the behavior of the classifier so that the results meet user's expectation.

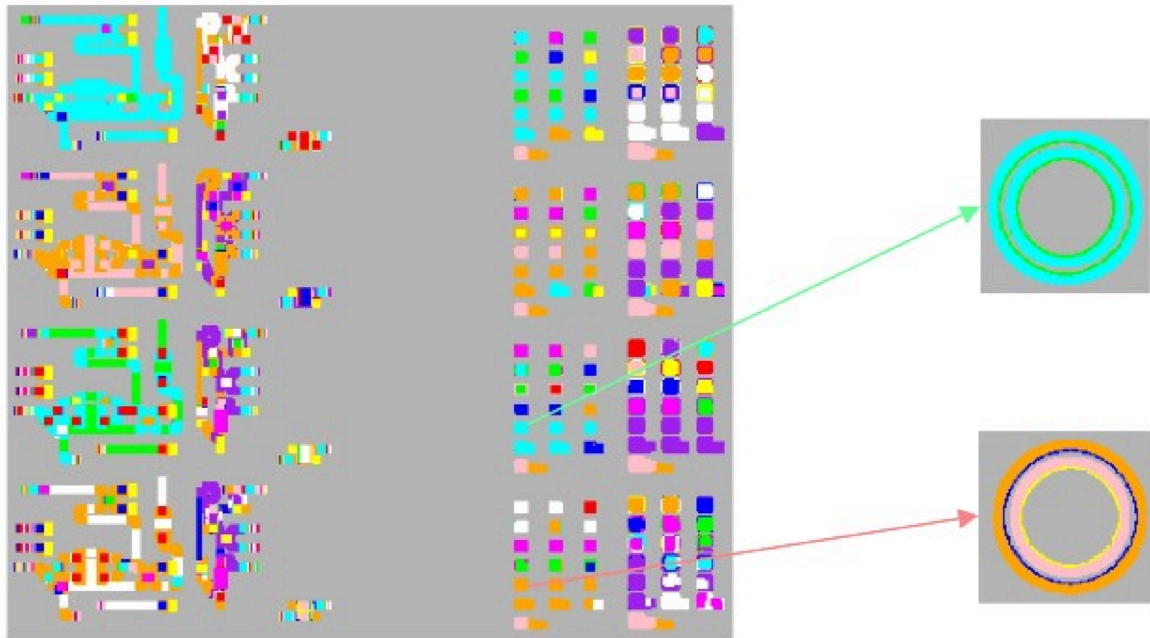


*Figure 11. Further sub-classify results on full layout. The 51 unique classes become 2493.

The results can also be reviewed in tree structure, as shown in Figure 12. The tree has four levels. The zoomed in view shows deeper branches of the tree.

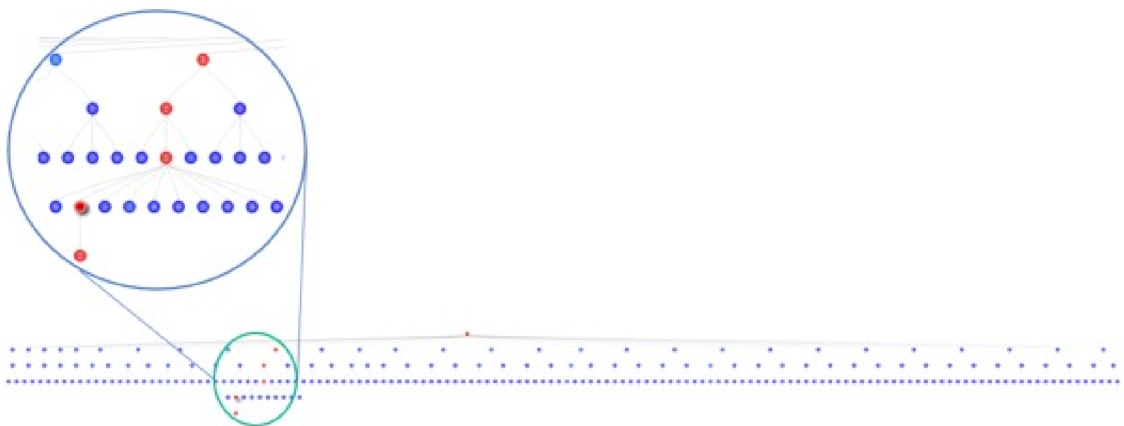


*Figure 12. Tree structure of the further sub-classify results of the full layout. The tree has four levels.

Comparisons between ML-based and geometry-based classify

In Calibre OPCVerify ML Classify, a narrowing down strategy is used, the primary matching is coarse, the optional secondary matching is finer. In geometry-based basic classify, the primary matching is an exact match, the optional secondary matching is relaxed by a user-input tolerance control. The comparison of classifying principles between the ML-based classify and the geometry-based basic classify is shown in Table 1.

	Features	**Primary matching**	**Secondary matching**
ML classify	Densities + selective geometries	coarse	fine
Basic classify	Vertices & edges	fine	coarse

*Table 1. Comparison table of classifying principles between ML classify and basic classify.

We compare the classify results on full layout as shown in Figure 13. Figure 13(a) is ML-based classify results with further sub-classify. Figure 13(b) is geometry-based basic classify results. For fair comparison, tolerance control of 10 nm is used in both classify solutions. The zoom in view shows ML-based classify results are pure, consistent, and satisfactory, however geometry-based basic classify results are strongly edge angle dependent, a simple ring pattern is classified into unnecessary large number of unique classes. The classify ratio of ML classify is 336, that of basic classify is only 4. The numbers are shown in Table 2.

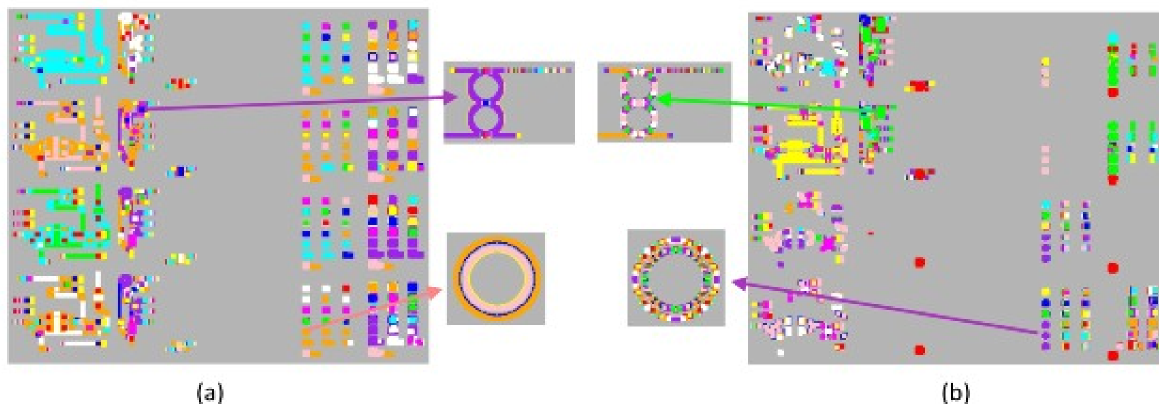


*Figure 13. Comparison of classifying results on full layout. (a) ML-based classify with further sub-classify. (b) Geometry-based basic classify. Tolerance control of 10 nm is used in both classify solutions.

	Total data count	Number of classes	Classify ratio
ML classify	837,072	2,493	336
Basic classify	837,072	221,085	4

*Table 2. Comparison table of classifying results on full layout between ML classify and basic classify.

Summary and conclusion

In summary, we have developed a machine learning-based verification error classification method. The good accuracy is achieved by properly design the density kernels. The tuning of density kernels is based on the dimensions of the features being checked. We get good results on full layout. This conclusion is drawn based on the ability to train the ML model in a way that allows the user to achieve good classification ratios (fewer errors to review) while still having the ability to differentiate between error examples that are sufficiently different to warrant separation in the classification categories. Our tool has large capacity in handling large data volume on full chip layouts. For this paper, we used the Calibre OPCVerify ML Classify solution on silicon photonics layouts. Next, we will apply the solution on other curvilinear layouts and Manhattan layouts as well. We are confident that our ML classify solution will continue to perform well in those new application use cases.

Acknowledgement

The authors would like to thank Liang Cao, Mohamed Bahnas, Jarrett Oney, Dingyi Hong, Evgueni Levine, Evgeny Flitman, Marko Chew, Chris Ashton, and Ilhami Torunoglu for helpful discussions and code implementations. The authors would like to thank Lei Sun for helping on metasurface optics discussions and references.

References

[1] T. N. Theis, H.-S. Wong, "The End of Moore's Law: A New Beginning for Information Technology," Computing in Science and Engineering, vol. 19, no. 2, pp. 41-50, Mar.-Apr. 2017, doi: 10.1109/MCSE.2017.29.
[2] L. Pavesi and D. J. Lockwood, Eds., "Silicon Photonics" (Springer, New York, 2004).
[3] G. T. Reed and A. P. Knights, "Silicon Photonics: An Introduction" (Wiley, Hoboken, NJ, 2004).
[4] M. Lipson, "Guiding, modulating, and emitting light on silicon—challenges and opportunities," J. Lightwave Technol. 23, 4222–4238 (2005).
[5] T. Tsuchizawa, K. Yamada, H. Fukuda, T. Watanabe, J. Takahashi, M. Takahashi, T. Shoji, E. Tamechika, E. Itabashi, and H. Morita, "Microphotonics devices based on silicon microfabrication technology," IEEE J. Sel. Top. Quantum Electron. 11, 232–240 (2005).
[6] R. A. Soref, "The Past, Present, and Future of Silicon Photonics," IEEE J. Sel. Top. Quantum Electron. 12, 1678- 1687 (2006).
[7] B. Jalali and S. Fathpour, "Silicon Photonics," J. Lightwave Technol. 24, 4600–4615 (2006).
[8] A. Barkai, Y. Chetrit, O. Cohen, R. Cohen, N. Elek, E. Ginsburg, S. Litski, A. Michaeli, O. Raday, D. Rubin, G. Sarid, N. Izhaky, M. Morse, O. Dosunmu, A. Liu, L. Liao, H. Rong, Y. Kuo, S. Xu, D. Alduino, J. Tseng, H. Liu, and M. Paniccia, "Integrated silicon photonics for optical networks," J. Optical Networking 6, 25–47 (2007).
[9] T. Barwicz, H. Byun, F. Gan, C. W. Holzwarth, M. A. Popovi´c, P. T. Rakich, M. R. Watts, E. P. Ippen, F. X. K¨artner, H. I. Smith, J. S. Orcutt, R. J. Ram, V. Stojanovic, O. O. Olubuyide, J. L. Hoyt, S. Spector, M. Geis, M. Grein, T. Lyszczarz, and J. U. Yoon, "Silicon photonics for compact, energy-efficient interconnects," J. Optical Networking 6, 63–73 (2007).
[10] L. Yin, Q. Lin, and G. P. Agrawal, "Dispersion tailoring and soliton propagation in silicon waveguides," Opt. Lett. 31, 1295–1297 (2006).
[11] Q. Lin, O. J. Painter, and G. P. Agrawal, "Nonlinear optical phenomena in silicon waveguides: Modeling and applications," Opt. Express 15, 16604–16644 (2007).
[12] N. Maluf and K. Williams, "An Introduction to Microelectromechanical Systems

Engineering", 2nd Edition (2002).
[13] M. K. Mishra, V. Dubey, P. M. Mishra, and I. Khan, "MEMS Technology: A Review. Journal of Engineering Research and Reports", 4(1), 1-24 (2019); https://doi.org/10.9734/jerr/2019/v4i116891.
[14] N. Quack et al., "MEMS-Enabled Silicon Photonic Integrated Devices and Circuits," in IEEE Journal of Quantum Electronics, 56(1), 1-10 (2020), Art no. 8400210; doi: 10.1109/JQE.2019.2946841.
[15] N. Yu, P. Genevet, M. A. Kats, F. Aieta, J. Tetienne, F. Capasso, Z. Gaburro, "Light Propagation with Phase Discontinuities: Generalized Laws of Reflection and Refraction", Science, 334 (6054), 333-337 (2011); DOI: 10.1126/science.1210713.
[16] N. Yu and F. Capasso, "Flat optics with designer metasurfaces", Nature Mater 13, 139–150 (2014); https://doi.org/10.1038/nmat3839
[17] R. C. Devlin, M. Khorasaninejad, W. T. Chen, J. Oh, and F. Capasso, "Broadband high-efficiency dielectric metasurfaces for the visible spectrum", Applied Physical Sciences 113 (38) 10473-10478; https://doi.org/10.1073/pnas.1611740113.
[18] V. Guo, F. Jiang, J. Rankin, I. Bork, A. Tritchkov, A. Wei, Y. Sun, S. Jayaram, L. Zhuang, X. Zhang, T. Bailey, J. Word, "Lithographic benefits and mask manufacturability study of curvilinear masks," Proc. SPIE 10810, Photomask Technology 2018, 108100P (2018); https://doi.org/10.1117/12.2501973.
[19] I. Bork, M. Reddy, B. Durvasula, N. Rao, M. Sharma, P. Buck, "CLMPC: curvilinear MPC in a mask data preparation flow," Proc. SPIE 10451, Photomask Technology 2017, 1045109 (2017); https://doi.org/10.1117/12.2282502
[20] J. Choi, S. Ryu, S. Lee, M. Kim, J. Park, P. Buck, I. Bork, B. Durvasula, S. Gharat, N. Rao, R. Pai, S. Koranne, A. Tritchkov, "Study on various curvilinear data representations and their impact on mask and wafer manufacturing," J. Micro/Nanopattern. Mats. Metro. 20(4) 041403 (2021); https://doi.org/10.1117/1.JMM.20.4.041403.
[21] I. Bork, A. Tritchkov, S. Shang, E. Levine, M. Zuo, "MRC for curvilinear mask shapes," Proc. SPIE 11518, Photomask Technology 2020, 115180R (2020); https://doi.org/10.1117/12.2575474.
[22] C. Spence, Q. Zhang, V. Shu, B. Chen, S. Baron, Y. Saito, M. Hamaji, Y. Horima, S. Ohara, "Manufacturing challenges for curvilinear masks," Proc. SPIE 10451, Photomask Technology 2017, 1045104 (2017); https://doi.org/10.1117/12.2280470.
[23] Y. Ma, F. Wang, Q. Xie, L. Hong, J. Mellmann, Y. Sun, S. W. Gao, S. Singh, P. Venkatachalam, J. Word, "Machine learning based wafer defect detection," Proc. SPIE

10962, Design-Process-Technology Co-optimization for Manufacturability XIII, 1096208 (2019); https://doi.org/10.1117/12.2513232.
[24] Y. Ma, L. Hong, J. Word, F. Jiang, V. Liubich, L. Cao, S. Jayaram, D. Kwak, Y. C. Kim, G. Fenger, A. Raghunathan, J. Mellmann, "Reduction of systematic defects with machine learning from design to fab," Proc. SPIE 11329, Advanced Etch Technology for Nanopatterning IX, 1132909 (2020); https://doi.org/10.1117/12.2551703.
[25] M. Ismail, M. Bahnas, T. Reimann, I. Torunoglu, K. Madkour, "A quantified approach of dataset selection for training ML models on hard-to-classify patterns," Proc. SPIE 11614, Design-Process-Technology Co-optimization XV, 116140A (2021); https://doi.org/10.1117/12.2586265.
[26] J. Kwan, F. Jiang, L. Hong, "Applying machine learning methods to accelerate advanced process node yield ramp," Proc. SPIE 11614, Design-Process-Technology Co-optimization XV, 116140T (2021); https://doi.org/10.1117/12.2584008.
[27] W. Bogaerts, "Introduction to Silicon Photonics Circuit Design", Short Course 454 - OFC 2018. https://www.photonics.intec.ugent.be/download/pub_4128.pdf

Siemens Digital Industries Software

Offices

Headquarters

Granite Park One 5800 Granite Parkway
Suite 600
Plano, TX 75024 USA
+1 972 987 3000

Americas

Granite Park One 5800 Granite Parkway
Suite 600
Plano, TX 75024 USA
+1 314 264 8499

Europe

Stephenson House
Sir William Siemens Square Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific

Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

For additional numbers, click here.

About us

Siemens Digital Industries Software helps organizations of all sizes digitally transform using software, hardware and services from the Siemens Xcelerator business platform. Siemens' software and the comprehensive digital twin enable companies to optimize their design, engineering and manufacturing processes to turn today's ideas into the sustainable products of the future. From chips to entire systems, from product to process, across all industries. Siemens Digital Industries Software – Accelerating transformation.

About the authors

Lianghong Yin
Shumay Shang
Fan Jiang

Fan Jiang is a product manager at Siemens Digital Industries Software, working on Calibre semi-manufacturing products.

Le Hong

Le Hong is a director of product engineering at Siemens Digital Industries Software for the Calibre semi-manufacturing products.

Robin Chia
Juli Opitz
Paul Adam
Ian Stobert
Yee Wee Koh

10/2023