ISSN 2616-5775 Vol. 9, Issue 2: 30-38, DOI: 10.25236/AJCIS.2026.090205

FPN and BAM Module-Enhanced Swin Transformer for Lithography Hotspot Detection

Lu Chunyan

School of Physics and Electronic Science, Hunan University of Science and Technology, Xiangtan, Hunan, China

Abstract: *To address the persistent challenges of low F1-score and high false alarm rate (FAR) in lithography hotspot detection, this study proposes a new detection framework based on a Dual Attention Swin transformer (DA-Swin). The proposed architecture leverages the multiscale feature fusion capability of the Feature Pyramid Network (FPN) to enhance the model's sensitivity to scale-invariant features. Meanwhile, the Bottleneck Attention Module (BAM) establishes a channel-spatial cooperative attention mechanism, achieving a 22.2% reduction in FAR. Compared to CrossEntropyLoss, the adoption of BCEWithLogitsLoss significantly reduces output layer parameters by approximately 50%. Benchmark tests on ICCAD demonstrate that our method achieves an accuracy of 98.67% and an F1-score of 0.857, surpassing prior works (TCAD'18, ICCAD'20, and JM3'24). Ablation studies further confirm the synergistic optimization of each module, delivering a highprecision (>98%) and costeffective (14.85% faster inference speed) solution for advancednode IC layout verification.*

Keywords: *Lithography hotspot detection; Deep learning; Feature pyramid; Swin Transformer*

1. Introduction

With the continuous scaling down of integrated circuit dimensions, traditional lithography techniques are facing increasingly severe challenges. When feature sizes shrink significantly below the exposure wavelength, the Optical Proximity Effect (OPE) induces deviations between the printed pattern and the design layouts, potentially causing critical defects such as line breaks or bridging during wafer fabrication[1]. To address these issues, Optical Proximity Correction (OPC)[2] is typically employed to minimize imaging errors by adjusting the mask layout. However, post-OPC patterns may still exhibit defects like bridging, short circuits, or open circuits referred to as Lithography Hotspots (LHs). These hotspots can severely impact IC performance, making hotspot detection indispensable for ensuring chip yield and functionality.

Current researches on hotspot detection primarily adopt three approaches: lithography simulation [3-5] , pattern matching[6,7], and machine learning[1,8]. While lithography simulation methods achieve high accuracy by modeling pattern transfer effects (e.g., optical systems, mask patterns, photoresist characteristics), they suffer from prohibitive computational costs, makingfull-chip analysis impractical. Pattern matching techniques, though efficient, rely on predefined templates and fail to generalize to novel hotspot patterns. In contrast, machine learning methods outperform others in detecting unknown hotspots but face challenges such as high false positive rates due to imbalanced datasets and excessive post-processing overhead.

Building on recent transformer advancements in layout analysis[9], we propose a Swin Transformer-based framework tailored for lithography hotspot detection, optimizing both computational efficiency and accuracy. Our key contributions include:

● A dual-module feature optimization framework: A Feature Pyramid Network (FPN) enables multi-scale hierarchical feature extraction, while a Bottleneck Attention Module (BAM) eliminates redundant layout noise. This synergy ensures feature completeness and purity, drastically reducing false alarms.

● An optimized training strategy: BCEWithLogitsLoss is adopted for binary classification, reducing output layer parameters and enhancing model lightweightness without sacrificing performance.

2. Improved Swin Transformer LHD Model

This paper employs a Swin-Transformer[10] as the baseline model for hotspot detection. Recognized as an advanced Transformer-based architecture[11], it incorporates hierarchical structures, shifted windows, and local self-attention mechanisms, exhibiting superior image processing capabilities. Having been widely adopted for classification and recognition tasks, its network architecture is illustrated in Figure 1.

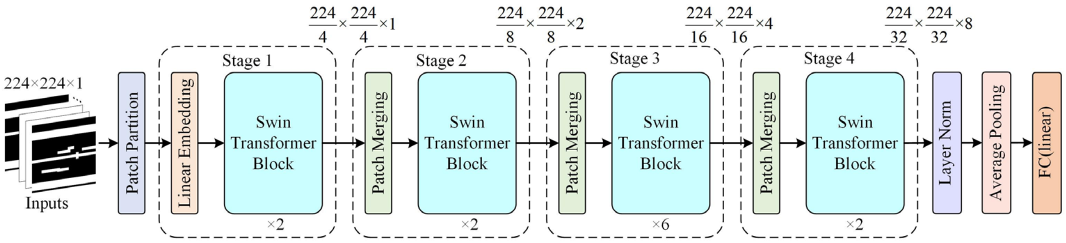

Figure 1: Swin Transformer Model (Tiny) Architecture.

During lithography layout processing, the input is first divided through patch partitioning, resulting in small patches (tokens). These tokens are then progressively transformed through four distinct stages to construct feature maps. Each stage's processing pipeline systematically reduces the spatial resolution while expanding the channel dimensionality, followed by sequential processing through layer normalization, global average pooling, and a fully-connected layer to generate the final classification output. The Swin Transformer module, which serves as the core algorithmic component, is detailed in Figure 2. This hierarchically designed module comprises four key components: Window based Multi-head Self-Attention (W-MSA), Shifted Window based MSA (SW-MSA), Multilayer Perceptron (MLP) and Layer Normalization (LN) layers.

The complete computational workflow of the module can be expressed mathematically as follows:

$$\begin{aligned}
\hat{X}^{l} &= \text{W-MSA}(\text{LN}(X^{l-1})) + X^{l-1} \\
X^{l} &= \text{MLP}(\text{LN}(\hat{X}^{l})) + \hat{X}^{l} \\
\hat{X}^{l+1} &= \text{SW-MSA}(\text{LN}(X^{l})) + X^{l} \\
X^{l+1} &= \text{MLP}(\text{LN}(\hat{X}^{l+1})) + \hat{X}^{l+1}
\end{aligned} \tag{1}$$

Here, $\hat{X}^{l}$ and $\hat{X}^{l+1}$ denote the output feature vector of the W-MSA and SW-MSA, respectively. X^{l} and X^{l+1} are the output of the MLP module after residual connections.

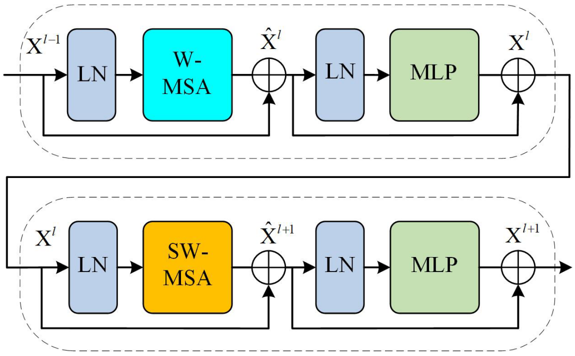

Figure 2: Swin Transformer Block.

Traditional MSA splits the input sequence into multiple heads, where each head independently calculates attention weights between all tokens and performs weighted fusion. The outputs of all heads are then concatenated. The Swin Transformer introduces two innovative modules: Window-based Multi-Head Self-Attention (W-MSA) and Shifted Window Multi-Head Self-Attention (SW-MSA), which employ localized window computations for self-attention, significantly reducing computational complexity as illustrated in Figure 3. Figure 3(b) shows W-MSA's fixed window partitioning of the input image. In each processing cycle, the windows are shifted upward and leftward by half a window size (1/2 window), causing the blue and red regions in Figure 3(c) to relocate to the image's bottom and right sides, as shown in Figure 3(d). SW-MSA employs this shifted window partitioning scheme to achieve cross-window exchange.

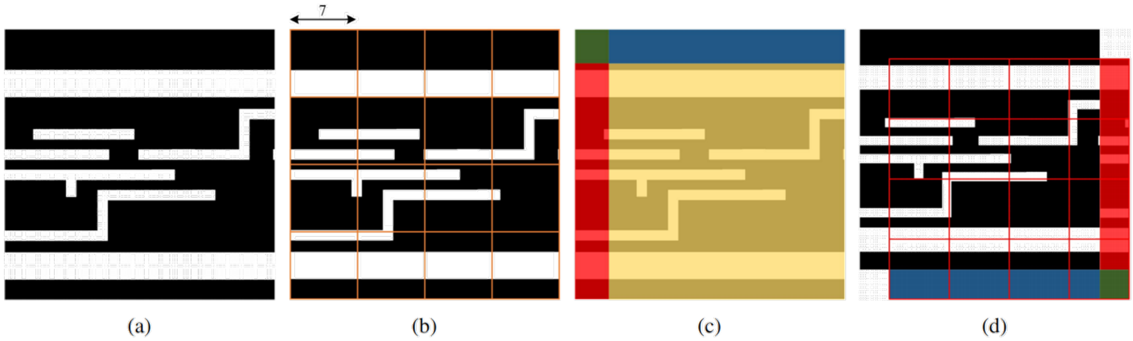

Figure 3: The Shifted Window Mechanism. (a) Input feature map. (b) W-MSA partitioning. (c) Shift operation. (d) SW-MSA partitioning.

2.1. Feature Pyramid Network

This study adopts a Feature Pyramid Network (FPN) architecture[12], as shown in Figure 4. Its multi-scale feature fusion capability is particularly effective for hotspot classifications due to its ability to preserve both local details and global context. We integrate hierarchical features from all four stages with FPN to create a unified feature re-presentation system, substantially improving multi-scale hotspot classification accuracy. The FPN operates in two key phases:

● Bottom-up feature extraction: Each Swin Transformer Block's output features form the FPN's base input, constructing an initial feature pyramid through hierarchical feature extraction. This harnesses the Transformer's rich feature representation capacity, providing a robust foundation for multi-scale fusion.

● Top-down feature fusion: Initially, stage 4's high-level features are compressed via 1×1 convolution and upsampled 2× (bilinear interpolation) to produce P3. Through lateral connections, P3 combines with stage 3 features via element-wise summation, yielding P2 with enhanced semantic-spatial information. This iterative process culminates in P1, which feeds into the classification head. This mechanism enables effective multi-scale feature fusion.

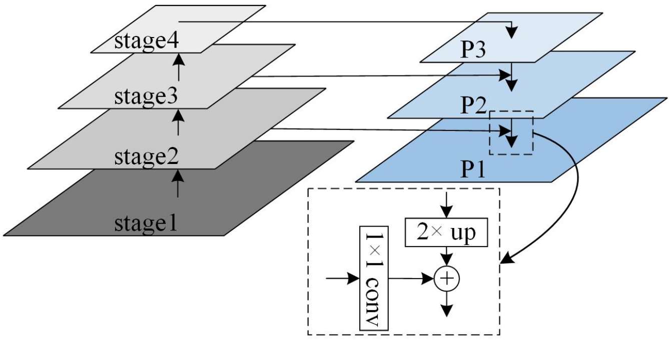

Figure 4: Feature Pyramid Network.

2.2. Bottleneck Attention Module

The Bottleneck Attention Module (BAM)[13] implements a dual parallel attention mechanism for joint spatial channel feature modeling in Figure 5. For the input feature tensor $F \in R^{C \times H \times W}$, it generates a 3D attention map $M(F) \in R^{C \times H \times W}$.

$$F' = F + F \otimes M(F) \tag{2}$$

$$M(F) = \sigma(M_C(F) + M_S(F)) \tag{3}$$

Where $\otimes$ denotes element-wise multiplication, F' represents the final processed feature map, $M_C \in R^{C \times 1 \times 1}$ corresponds to the feature map from the channel attention branch, and $M_S \in R^{1 \times H \times W}$ indicates the feature map from the spatial attention branch. Notably, both attention branches must be adjusted to

$\mathrm{M(F)} \in \mathrm{R}^{C \times H \times W}$ before being summed. σ refers to the sigmoid function, which normalizes the weights to the range [0, 1].

$\mathrm{M_C(F)}$ and $\mathrm{M_S(F)}$ are defined as follows:

$$\mathrm{M_C(F)} = BN(MLP(AvgPool(\mathrm{F}))) \tag{4}$$

$$\mathrm{M_S(F)} = BN(f_3^{1\times1}(f_2^{3\times3}(f_1^{3\times3}(f_0^{1\times1}(\mathrm{F}))))) \tag{5}$$

Here, BN is the Batch Normalization, f denotes convolution (superscript: kernel size k×k; subscript: index of the convolution layer). The 1×1 convolution reduces channel, while 3×3 dilated convolutions expand the receptive field.

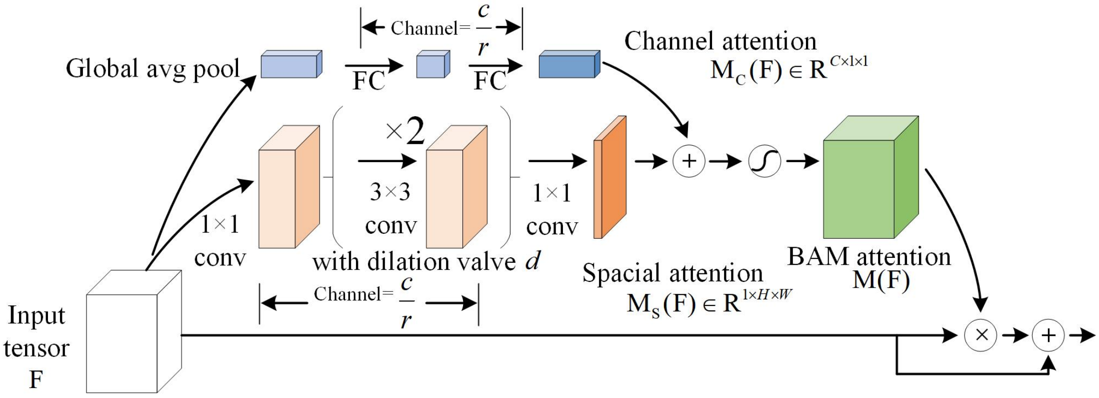

Figure 5: Bottleneck Attention Module Structure.

2.3. Lithographic Hotspot Detection Model

Figure 6 illustrates our proposed hotspot detection network, comprising three core modules: feature extraction, feature fusion and classification.

The feature extraction module adopts a four-stage Swin Transformer architecture to hierarchically extract multiscale features. augmented by BAM to amplify critical features representations. For the feature fusion, we designed an improved FPN architecture optimized for classification tasks. This structure adaptively merges feature maps at different resolutions, significantly enhancing the network's ability to perceive multi-scale features.

The classification module adopts a three-stage cascaded design:

- Layer Normalization: Normalizes feature maps along the channel dimension, an operation that not only stabilizes the training process but also significantly enhances the model's generalization capability.

- Global Pooling: Compresses spatially distributed features into compact global descriptor vectors, preserving channel-level semantic information while reducing dimensionality.

- Linear Projection: Maps processed features to the categorical space via a fully-connected layer, completing the final classification decision for lithography hotspot detection.

This meticulously designed cascaded structure achieves efficient identification and classification of lithography hotspot features while maintaining model efficiency.

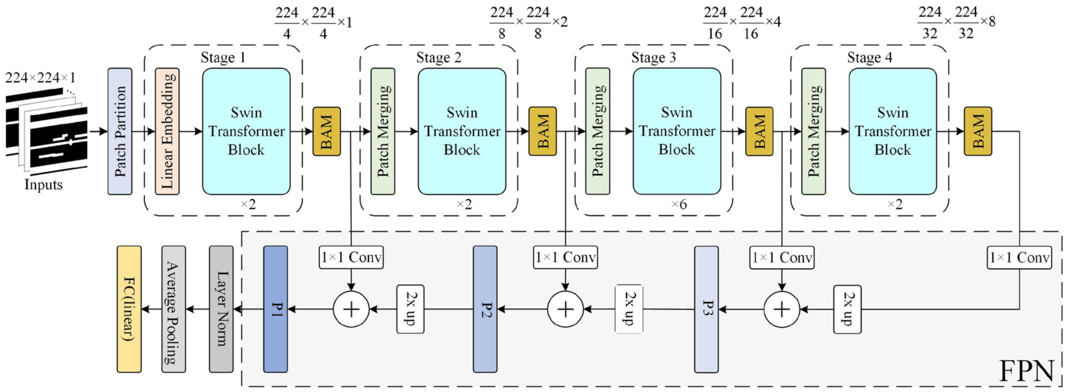

Figure 6: Improved Swin Transformer LHD Model.

2.4. Loss Function

Lithographic hotspot detection is fundamentally a binary classification task requiring definitive "hotspot/nonhotspot" decisions for each pattern. We employ BCEWithLogitsLoss which combines sigmoid activation and binary cross-entropy for numerical stability. The loss function is defined as:

$$L=-\frac{1}{N}\sum_{i=1}^{N}[y_i\cdot\log(\sigma(p_i))+(1-y_i)\cdot\log(1-\sigma(p_i))] \tag{6}$$

$$\sigma(p_i)=\frac{1}{1+e^{-p_i}} \tag{7}$$

Here, N denotes the total samples. y_i stands for the label value of the i-th sample, when $y_i=1$,it indicates that the sample belongs to a positive sample (hotspot). Meanwhile, p_i is the logit output by the model (the raw value without Sigmoid activation). $\sigma(\cdot)$ is the sigmoid function, which maps the logit to the interval (0,1) as the predicted probability. In binary classification tasks, BCEWithLogitsLoss is typically employed to process the model's raw logits (i.e., unnormalized outputs before Sigmoid activation). It takes the logits as input, performs probability estimation via the Sigmoid function, and subsequently computes the binary cross-entropy loss. During computation, BCEWithLogitsLoss first applies the Sigmoid activation to the logits and then derives the predicted probabilities along with the binary cross-entropy loss.

To align with the single-logit output format of BCEWithLogitsLoss, we optimize the output layer from the original Linear(D, 2) to Linear(D, 1). This modification not only matches the semantic design of the loss function but also reduces the output layer parameters from 1,538 to 769, achieving approximately 50% reduction. Although the impact on total parameter count is marginal, this optimization holds practical significance for lightweight model deployment. The comparative parameter counts of the model's output heads are detailed in Table 1.

Table 1: Parameter Comparison of Model Output Head

Loss Function	Output Structure	Weights	Bias	Total
CrossEntropyLoss	Linear(768,2)	1536	2	1538
BCEWithLogitsLoss	Linear(768,1)	768	1	769

3. Experiment and discussion

To evaluate the effectiveness of the proposed method, this paper employs the ICCAD 2012 competition benchmarks[14] for model training and validation. The training set includes all augmented data, whereas the test set consists of the original ICCAD 2012 Layout Dataset without modification. All model training and testing procedures are performed on a hardware platform featuring an Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz and NVIDIA Tesla P40 GPUs, with Ubuntu 20.04, Python 3.7, and PyTorch 1.12.1.

3.1. Training Configurations and Benchmark Information

The model is trained using the AdamW optimizer for 100 epochs, with an initial learning rate set to 1.25×10^{-4} and a weight decay coefficient of 0.05. A cosine annealing learning rate scheduler with a 15 epoch linear warm-up phase ensures training stability. To accommodate the geometric integrity of layout data, we have specifically designed data augmentation (random horizontal/vertical flipping) and disabled standard Swin Transformer augmentation (Mixup and Cutmix). Additional parameters include a batch size of 128 and a gradient clipping threshold of 5.0, as summarized in Table 2.

Table 2: Training configurations

Training Configuration	**Value**
Optimizer	AdamW
Base learning rate	$1.25\text{x}10^{-4}$
Weight decay	0.05
Training epochs	100
Learning rate schedule	Cosine
Warm epochs	15
Batch size	128
Mixup	0.0
Cutmix	0.0
Gradient clip	5.0

Before training, the dataset underwent Discrete Cosine Transform (DCT) processing[15]. Due to the insufficient hotspots in the original training data, we perform dataset augmentation (see Table 3). To validate the scalability, all 28nm test cases are merged into a unified “ ICCAD” set. Columns “#HS” and “#NHS” show the total number of hotspots and non-hotspots in the training and test sets.

Table 3: ICCAD 2012 contest benchmark (expanded)

Dataset	**Technology(nm)**	**Train**		**Test**	
		#HS	***#NHS***	***#HS***	***#NHS***
Benchmark2	28	1392	5285	498	41298
Benchmark3	28	909	4643	1808	46333
Benchmark4	28	760	4452	177	31890
Benchmark5	28	208	2716	41	19327
ICCAD(B2-B5)	28	3269	17096	2524	138848

3.2. Comparison

To validate the model’s efficacy, we conducted comparative experiments with hotspot detection methods on the 28nm process ICCAD-2 to ICCAD-5 benchmark sets, as show in Table 4. Since some studies only reported results for these four benchmarks, we first present the comparative analysis of these four datasets.

Table 4: Comparison of Existing Methods

Pattern	**Model**	**Accuracy**	**#FP**	**CPU (s)**	**ODST(s)**
Benchmark2	***JM3’16***[16]	98.80	1790	208	18108
	SPIE’17[17]	**99.80**	561	390	6000
	JM3’19[18]	99.21	352	**123**	3643
	Ours	99.45	**229**	598	**2888**
Benchmark3	***JM3’16***	97.50	7077	321	71091
	SPIE’17	97.80	2660	434	27034
	JM3’19	**98.22**	2735	**145**	27495
	Ours	97.75	**999**	825	**10818**
Benchmark4	***JM3’16***	93.80	892	129	9049
	SPIE’17	96.40	1785	333	18183
	JM3’19	96.32	789	**92**	7982
	Ours	**98.31**	**528**	542	**5822**
Benchmark5	***JM3’16***	92.70	172	81	1801
	SPIE’17	95.10	242	232	2652
	JM3’19	97.32	80	**56**	**856**
	Ours	**99.69**	**58**	329	909

A comprehensive analysis of Table 4 reveals that the our method outperforms competitors in both detection accuracy and efficiency: it achieves state-of-the-art accuracy rates of 99.31% and 99.69% on ICCAD-4 and ICCAD-5 datasets, respectively, while maintaining the lowest or second-lowest false positive rates across all datasets (with a 63.47% reduction on ICCAD-3). Although a single-detection runtime is slightly longer, the significant reduction in false positives enables the ODST benchmark to achieve optimal performance across all datasets. While the 97.75% accuracy on ICCAD-3 is marginally lower thanJM3'19, the drastic reduction in false positives (from 2735 to 999) ensures an excellent precision-efficiency balance, make it ideal for large-scale layout verification.

Additionally, Table 5 presents a comparison of our results against four recent studies that reported ICCAD benchmark finding. The proposed model exhibits remarkable superiority in both detection performance and computational efficiency: Compared with the JM3'24 method, it reduces thefalse alarm rate by 22.24% while improving detection accuracy by 0.17, achieving an F1-score of 0.857—a 46.20% average enhancement over the other three competing methods. Although the single-test runtime is slightly higher than rival approaches, our ODST is the lowest, indicating the minimal total runtime within the detection pipeline. We adopt the same evaluation metrics from previous work to assess the performance of our proposed lithography hotspot detection model. Hotspot detection results can be categorized into the following scenarios:

True Positive(TP): Correctly identifying hotspot graphics as hotspots.

True Negative(TN): Correctly identifying non-hotspot graphics as non-hotspot.

False Positive(FP): Incorrectly identifying non-hotspot graphics as hotspots.

False Negative(FN): Incorrectly identifying hotspots as non-hotspot graphics.

Table 5: Comparison of Existing Methods

Pattern	Model	Accuracy	F1-score	#FP	CPU(s)	ODST(s)
ICCAD(B2-B5)	***ICCAD'16*[19]**	97.70	0.520	4497	-	-
	***TCAD'18*[15]**	98.40	0.580	3535	554	35904
	***ICCAD'20*[20]**	98.42	-	2481	**144**	24954
	***JM3'24*[21]**	98.50	0.680	2280	528	23328
	Ours	**98.67**	**0.857**	**1773**	2395	**20125**

In practical VLSI design flows, all detected hotspot clips must undergo rigorous lithography simulation verification. To evaluate the engineering applicability, we adopts the overall detection and simulation time (ODST) metric from the prior work[19], which combines: Lithography simulation time for all predicted hotspots, and Model inference time. Notably, the industrial-grade lithography simulator[22] employed in this study requires 10 seconds per clip for simulation. Thus, the ODST is quantified as:

$$\text{ODST} = \text{Test time} + 10\text{s} \times \#\text{of False Alarm} \tag{8}$$

3.3. Ablation Experiment

The ablation study reveals significant synergistic effects between the BAM and FPN modules. When deployed individually, BAM improves accuracy but trades runtime, while FPN boosts both accuracy and computational efficiency. Their integrated implementation achieves optimal balance: it attains 98.67% accuracy (a +0.25% increase) and an F1-score of 0.857 (a +2.26% improvement), while reducing false alarms by 16.09% and decreasing ODST time by 14.85%, as detailed in Table 6.

Table 6: Ablation experimental results on ICCAD(B2-B5)

Method		Accuracy	F1-score	#FA	CPU(s)	ODST(s)
BAM	***FPN***					
		98.42	0.838	2113	2505	23635
√		98.53	0.846	1974	3639	23379
	√	98.64	0.855	1814	2999	21139
√	√	**98.67**	**0.857**	**1773**	2395	**20125**

4. Conclusion

This study proposes an enhanced Swin Transformer-based framework for lithography hotspot detection, which achieves multi-scale feature fusion and dynamic noise suppression through the integration of dual FPN and BAM modules. Experimental results demonstrate the scheme's superior performance on the ICCAD(B2-B5) benchmark: it reduces FAR by 16.09% and decreases the ODST by 14.85%, thereby providing an efficient, production-ready solution for advanced semiconductor process verification in large-scale layout inspection.

References

[1] Duo Ding, Andres J Torres, Fedor G Pikus, and David Z Pan. High performance lithographic hotspot detection using hierarchically refined machine learning. In 16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011), pages 775–780. IEEE, 2011.

[2] Ao Chen, Yee Mei Foong, Thomas Thaler, Ute Buttgereit, Angeline Chung, Andrew Burbine, John Sturtevant, Chris Clifford, Kostas Adam, and Peter De Bisschop. Aerial image metrology for opc modeling and mask qualification. In 33rd European Mask and Lithography Conference, volume 10446, pages 181–194. SPIE, 2017.

[3] Juhwan Kim and Minghui Fan. Hotspot detection on post-opc layout using full-chip simulation-based verification tool: a case study with aerial image simulation. In 23rd Annual BACUS Symposium on Photomask Technology, volume 5256, pages 919–925. SPIE, 2003.

[4] Ed Roseboom, Mark Rossman, Fang-Cheng Chang, and Philippe Hurat. Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs. In Design for Manufacturability through Design-Process Integration, volume 6521, pages 120–128. SPIE, 2007.

[5] Puneet Gupta, Andrew B Kahng, Sam Nakagawa, Saumil Shah, and Puneet Sharma. Lithography simulation-based full-chip design analyses. In Design and Process Integration for Microelectronic Manufacturing IV, volume 6156, pages 277–284. SPIE, 2006.

[6] Hailong Yao, Subarna Sinha, Charles Chiang, Xianlong Hong, and Yici Cai. Efficient process-hotspot detection using range pattern matching. In Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design, pages 625–632, 2006.

[7] Wan-Yu Wen, Jin-Cheng Li, Sheng-Yuan Lin, Jing-Yi Chen, and Shih-Chieh Chang. A fuzzy-matching model with grid reduction for lithography hotspot detection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 33(11):1671–1680, 2014.

[8] Yen-Ting Yu, Geng-He Lin, Iris Hui-Ru Jiang, and Charles Chiang. Machine-learning-based hotspot detection using topological classification and critical feature extraction. In Proceedings of the 50th annual design automation conference, pages 1–6, 2013.

[9] Rajesh Rohilla et al. Exploring vision transformer model for detecting lithography hotspots. In 2022 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), volume 2, pages 200–205. IEEE, 2022.

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision, pages 10012–10022, 2021.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[13] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. arXiv preprint arXiv:1807.06514, 2018.

[14] J Andres Torres. Iccad-2012 cad contest in fuzzy pattern matching for physical verification and benchmark suite. In Proceedings of the International Conference on Computer-Aided Design, pages 349–350, 2012.

[15] Haoyu Yang, Jing Su, Yi Zou, Bei Yu, and Evangeline FY Young. Layout hotspot detection with feature tensor generation and deep biased learning. In Proceedings of the 54th Annual Design Automation Conference 2017, pages 1–6, 2017.

[16] Moojoon Shin and Jee-Hyong Lee. Accurate lithography hotspot detection using deep convolutional neural networks. Journal of Micro/Nanolithography, MEMS, and MOEMS, 15(4),043507 (18 November 2016).

[17] 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),033504 (24 August 2017).

[18] Jing Chen, Yibo Lin, Yufeng Guo, Maolin Zhang, Mohamed Baker Alawieh, and David Z Pan. Lithography hotspot detection using a double inception module architecture.Journal of Micro/Nanolithography, MEMS, and MOEMS, 18(1), 013507 (22 March 2019).

[19] Hang Zhang, Bei Yu, and Evangeline FY Young. Enabling online learning in lithography hotspot detection with information theoretic feature optimization. In Proceedings of the 35th International Conference on Computer-Aided Design, pages 1–8, 2016.

[20] Hao Geng, Haoyu Yang, Lu Zhang, Jin Miao, Fan Yang, Xuan Zeng, and Bei Yu. Hotspot detection via attention-based deep layout metric learning. In Proceedings of the 39th International Conference on Computer-Aided Design, pages 1–8, 2020.

[21] Hui Xu, Ye Yuan, Ruijun Ma, Pan Qi, Fuxin Tang, Xinzhong Xiao, Wenxin Huang, and Huaguo Liang. Lithography hotspot detection through multi-scale feature fusion utilizing feature pyramid network and dense block. Journal of Micro/Nanopatterning, Materials, and Metrology, 23(1), 013202 (10 February 2024).

[22] Shayak Banerjee, Zhuo Li, and Sani R Nassif. Iccad-2013 cad contest in mask optimization and benchmark suite. In 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pages 271–274. IEEE, 2013.