applied sciences

MDPI

Article

Lithography Hotspot Detection Method Based on Transfer Learning Using Pre-Trained Deep Convolutional Neural Network

Lufeng Liao [1,2], Sikun Li [1,2,*], Yongqiang Che [3], Weijie Shi [4] and Xiangzhao Wang [1,2,*]

[1] Laboratory of Information Optics and Opto-Electronic Technology, Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai 201800, China; liaolufeng@siom.ac.cn
[2] Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
[3] Semiconductor Manufacturing North China (Beijing) Corporation, Beijing 100176, China; stephen_che@smnchina.com
[4] Dongfang Jingyuan Electron Ltd., Beijing 100176, China; weijie.shi@dfjy-jx.com
* Correspondence: lisikun@siom.ac.cn (S.L.); wxz26267@siom.ac.cn (X.W.)

Abstract: As the designed feature size of integrated circuits (ICs) continues to shrink, the lithographic printability of the design has become one of the important issues in IC design and manufacturing. There are patterns that cause lithography hotspots in the IC layout. Hotspot detection affects the turn-around time and the yield of IC manufacturing. The precision and F1 score of available machine-learning-based hotspot-detection methods are still insufficient. In this paper, a lithography hotspot detection method based on transfer learning using pre-trained deep convolutional neural network is proposed. The proposed method uses the VGG13 network trained with the ImageNet dataset as the pre-trained model. In order to obtain a model suitable for hotspot detection, the pre-trained model is trained with some down-sampled layout pattern data and takes cross entropy as the loss function. ICCAD 2012 benchmark suite is used for model training and model verification. The proposed method performs well in accuracy, recall, precision, and F1 score. There is significant improvement in the precision and F1 score. The results show that updating the weights of partial convolutional layers has little effect on the results of this method.

Keywords: hotspot detection; optical lithography; transfer learning; convolutional neural network

Citation: Liao, L.; Li, S.; Che, Y.; Shi, W.; Wang, X. Lithography Hotspot Detection Method Based on Transfer Learning Using Pre-Trained Deep Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 2192. https://doi.org/10.3390/app12042192

Academic Editors: Xiuguo Chen, Guanhao Wu and Yuki Shimizu

Received: 20 January 2022
Accepted: 16 February 2022
Published: 19 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The lithographic tool is important equipment for very large-scale IC manufacturing. Its function is to transfer the mask pattern into the photoresist on wafer. The process of lithography determines the integration of IC, which pushes forward the Moore's law. Nowadays, the designed feature size of IC is below 10 nm, and the number of transistors of an IC is as high as tens of billions [1,2]. With the demand for high integration and better performance, the physical design of IC continues to shrink, and the lithographic printability has become one of the critical issues in IC design and manufacturing [3,4]. Affected by the layout design and lithography process, the lithography results of some patterns in the layout are quite different from the target patterns, resulting in short-circuit or open-circuit problems. This problem will cause lithography hotspots. In order to reduce lithography hotspots, hotspot detection and layout correction are carried out in turn in the layout design stage. The performance of hotspot detection affects the turn-around time and the yield of IC manufacturing. Hotspot detection is one of the important techniques for IC design and manufacturing [3–6].

A lot of hotspot-detection research has been carried out. The available hotspot-detection methods include the lithography simulation-based detection method [7,8], the pattern matching-

based detection method [5,9,10], and the machine-learning-based detection method [11,12]. The lithography simulation-based hotspot-detection method predicts the lithography result on the wafer by physically simulating the lithography process, so as to find the corresponding hotspot areas on the layout. This method has high detection accuracy, but it is extremely time-consuming [13]. The pattern matching-based detection method detects hotspots by evaluating the degree of similarity between the layout patterns and the registered hotspot patterns. This method is fast, but it is invalid for unknown hotspot pattern [12–16]. The machine-learning-based detection method trains a model based on layout data, and the trained model is used to detect the hotspot patterns on the layout. This method is valid for unknown hotspot patterns with good detection performance. The machine-learning-based detection method has the advantage of high detection speed and has been widely studied [11–14,16].

Feature extraction and model design are important parts of the machine-learning-based hotspot-detection method. In order to obtain better performance, feature extraction has evolved from the artificially designed feature-extraction method [14,17,18] to using the convolutional neural network (CNN) [16,19]. Model design has also been developed from shallow networks to deep networks [16,20,21]. In general, the deep learning networks have many layers, which require more training parameters and have a high cost of model training. Transfer learning can use a model trained with other datasets as a pre-trained model and fine-tune the pre-trained model with the target dataset to obtain a suitable model for the specified target design. In recent years, transfer learning has developed rapidly and has been widely used [22–24]. A transfer-learning-based hotspot-detection method has begun to emerge [25,26]. Accuracy, recall, precision, and F1 score are commonly used as evaluation indicators for machine learning [27]. For hotspot detection, recall is related to the hotspot detection rate, precision is related to the false alarm, and the F1 score indicates the comprehensive performance of the model in terms of recall and precision. A good hotspot-detection model should perform well in F1 score, which means it has a high hotspot-detection rate and low false-alarm rate. Although the available machine-learning-based hotspot-detection methods perform well in recall, they still have insufficient precision and F1 score [16,18,19,25,26]. A high false-alarm rate will increase the post-processing steps and increase the turn-around time of IC manufacturing. In this paper, a lithography hotspot detection method based on transfer learning using pre-trained deep CNN is proposed. The proposed method uses the VGG13 network [28] trained with the ImageNet dataset [29] as the pre-trained model. In order to obtain a model suitable for hotspot detection, the pre-trained model is trained with some down-sampled layout pattern data and takes cross entropy as the loss function. ICCAD 2012 benchmark suite [30] is used for model training and model verification. Comparisons with Samsung's hotspot-detection method based on deep CNNs [16] and the hotspot-detection methods based on transfer learning in the past two years [25,26] were carried out. The results show that the proposed method performs well in accuracy, recall, precision, and F1 score. Additionally, there is significant improvement in the precision and F1 score. Compared with Samsung's deep CNN-based hotspot-detection method, the average precision and F1 score are improved by 298% and 159%, respectively. In order to test the effect of updating the weights of the convolutional layers on the results, partial convolutional layers were released for model training. Compared with freezing all convolutional layers, the results show that updating the weights of partial convolutional layers has little effect on the results of this method.

2. Methods

2.1. Workflow

The machine-learning-based hotspot-detection method obtains a hotspot detection model through model training based on layout data. The CNN has good image-classification performance, and layout data can be converted to pattern data, so CNN can be used for hotspot detection [19]. The CNN-based hotspot-detection method is shown in Figure 1. In the model-training phase, the layout pattern data are used to train a model suitable for hotspot detection. When performing hotspot detection, the trained model takes the layout

pattern as input, and the trained model will identify whether the input layout pattern is a hotspot pattern or not.

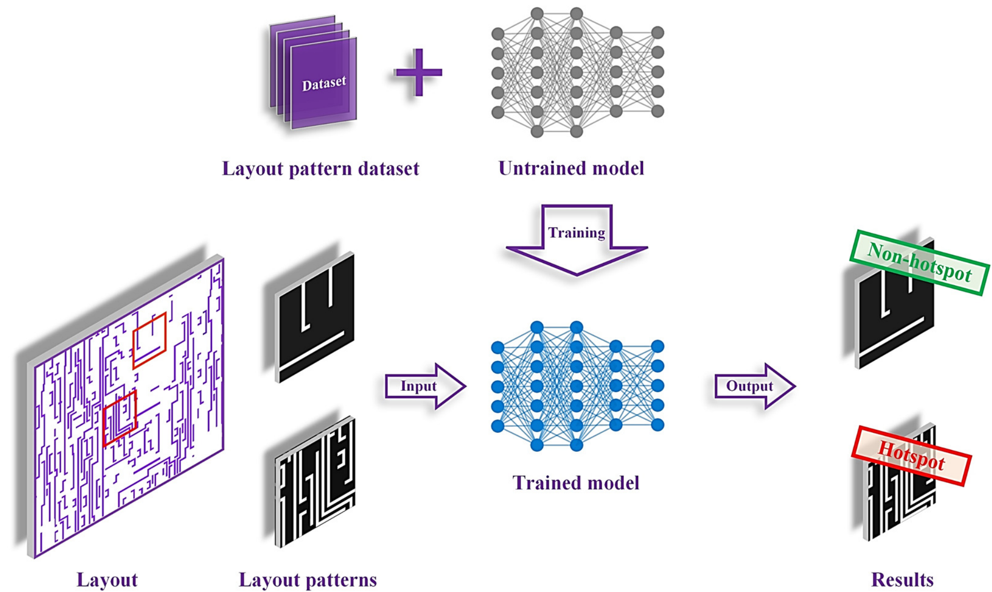

Figure 1. Schematic diagram of the CNN-based hotspot-detection method.

The workflow of the proposed lithography hotspot detection method based on transfer learning using pre-trained deep CNN is shown in Figure 2. The proposed method includes three phases: preparation, model training, and model verification. In the preparation phase, both the data and model are prepared. In order to reduce the cost of model training, after the input layout data are converted to pattern data, data compression is required. In addition, it is necessary to deal with the problem of imbalance between positive and negative samples of the training data. The pre-trained VGG13 model based on the ImageNet dataset is open access. In the model-training phase, it is necessary to modify the pre-trained VGG13 network architecture to make it suitable for hotspot detection, and then the model is trained with the training data. In the model-verification phase, the test layout data are used to evaluate the performance of the trained model. In order to evaluate the performance of the proposed hotspot-detection method, it is necessary to evaluate the accuracy, recall, precision, and F1 score. The result of hotspot detection is defined as follows:

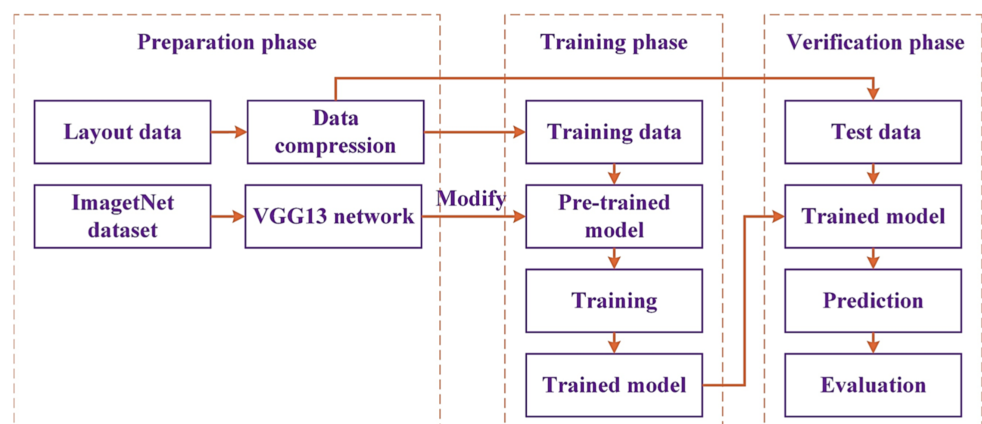

Figure 2. The workflow of the proposed method.

True positive (TP): the hotspot pattern is identified as a hotspot pattern.
True negative (TN): the non-hotspot pattern is identified as a non-hotspot pattern.

False positive (FP): the non-hotspot pattern is identified as a hotspot pattern.
False negative (FN): the hotspot pattern is identified as a non-hotspot pattern.
The accuracy, recall, precision, and F1 score of the hotspot-detection method are defined as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (1)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (2)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (3)$$

$$\text{F1} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}, \quad (4)$$

where the accuracy refers to the ratio of the correctly identified patterns to all the patterns, the recall refers to the ratio of patterns correctly identified as hotspot and all the hotspot patterns, and the precision refers to the ratio of real hotspot patterns and patterns identified as hotspot patterns. The F1 score is a factor for comprehensive evaluation of recall and precision. The values of accuracy, recall, precision, and F1 score are all between 0 and 1. The higher accuracy value means the more patterns are correctly identified. The higher recall value means the more hotspot patterns are identified. The higher precision value means the false alarm is lower.

2.2. Data Preparation

2.2.1. Data Compression

For the CNN-based hotspot-detection method, model training and model verification are based on layout pattern data. The design resolution of the IC layout is as high as 1 nm. For a layout pattern with an area of 1 μm^2, the number of pixels is as high as 1000 × 1000. Model training based on high-resolution pattern data requires a high-performance hardware system. Therefore, it is necessary to perform data compression on the original high-resolution patterns. The density-based feature-extraction method [20,21] is a conventional feature-extraction method for hotspot detection, which can be used for data compression. The proposed method takes the strategy of calculating the local density to down-sample the original pattern, thereby achieving data compression. The proposed method is different from the other methods [19,26]. On the one hand, the resolution of the down-sampled patterns is higher, which is 240 × 240 pixels. On the other hand, the resolution of the down-sampled patterns is similar to the resolution of the ImageNet dataset, but not completely the same.

The schematic diagram of pattern down-sampling is shown in Figure 3. After the high-resolution pattern is obtained, the pattern is gridded, and each grid window corresponds to a pixel of the down-sampled pattern. The pattern down-sampling can be expressed by Equation (5), and the pixel value of the down-sampled pattern is the average pixel value of the corresponding window. In Equation (5), p_k represents the kth pixel value of the down-sampled pattern, $win(k)$ refers to the kth window in the grid, I_i refers to the ith pixel value of the original pattern in $win(k)$, and N refers to the total number of pixels in the window.

$$p_k = \frac{1}{N} \sum_{i \in win(k)} I_i, \quad (5)$$

2.2.2. Data Balance

There are more non-hotspot patterns than hotspot patterns in the IC layout. Therefore, there is an imbalance problem between positive and negative sample sets in hotspot detection [30]. As shown in Figure 4, in the training data of ICCAD 2012 benchmark suite, the proportion of hotspots and non-hotspots is seriously unbalanced. In Benchmark 5, the proportion of hotspots training data is less than 1%. For the machine-learning-based

hotspot-detection method, the imbalance between the hotspots and the non-hotspots of the training data will affect the performance of the trained model [31]. In order to address the imbalance between positive and negative samples, the proposed method takes the under-sampling strategy [32] to randomly sample non-hotspot data. In the training data of ICCAD 2012 benchmark suite, the randomly sampled non-hotspot data and all hotspot data constitute a complete training dataset. On the one hand, the proposed method can address the serious imbalance between the positive and negative samples of the training data. On the other hand, only partial the non-hotspot data of the training data are used, which reduces the overall training data volume.

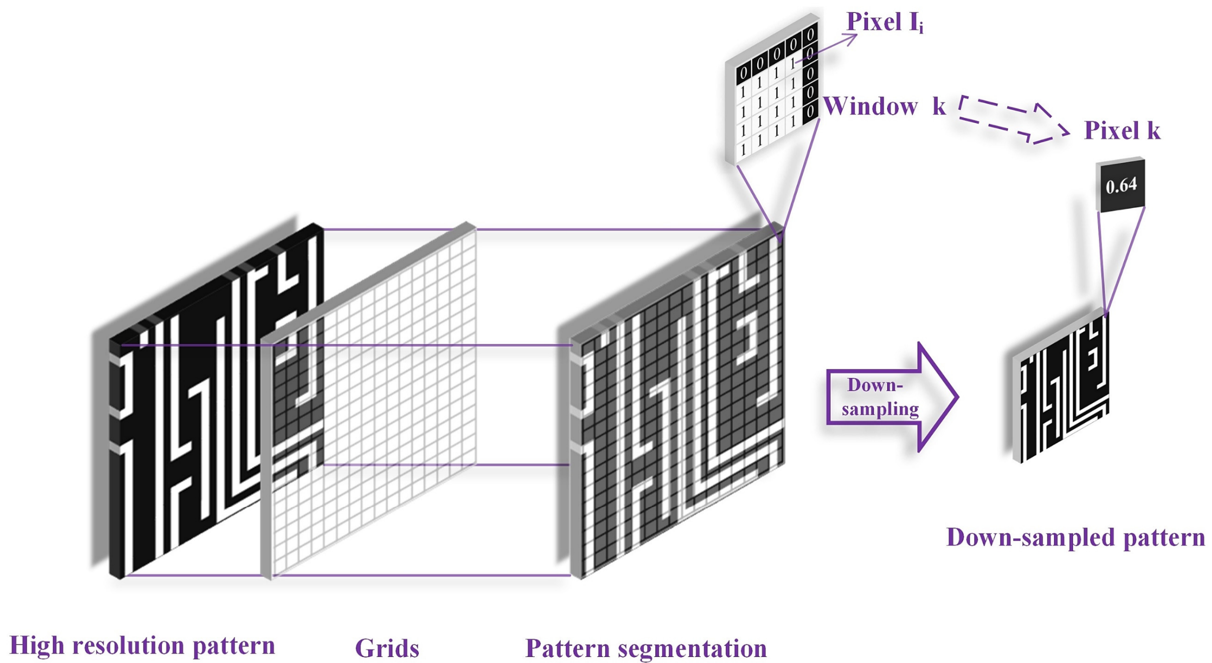

Figure 3. Schematic diagram of pattern down-sampling.

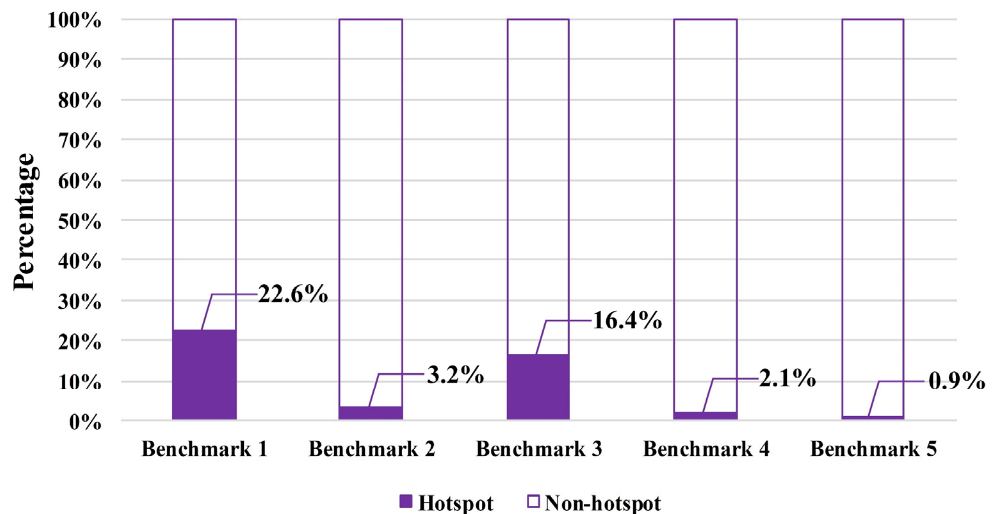

Figure 4. The percentage of hotspot training data in ICCAD 2012 benchmark suite.

2.3. Model Modification and Model Training

2.3.1. Model Modification

The VGG13 network is a deep CNN with a convolution kernel size of 3 × 3. The architecture of the VGG13 network is shown in Figure 5, which includes convolutional layers, pooling layers, and fully connected layers. Only the convolutional layers and the fully connected layers have weights, and the number of the layers with weights is 13. For the convolutional layers, the number of convolution kernels increases layer by layer, from 64 in the first layer to 512 in the last layer. For the input pattern, the feature maps are extracted by the convolutional layers, and the feature maps are downsampled by the pooling layers. The output features of the last pooling layer are connected with the fully connected layers. The fully connected layers act as the classifier to obtain the final classification output.

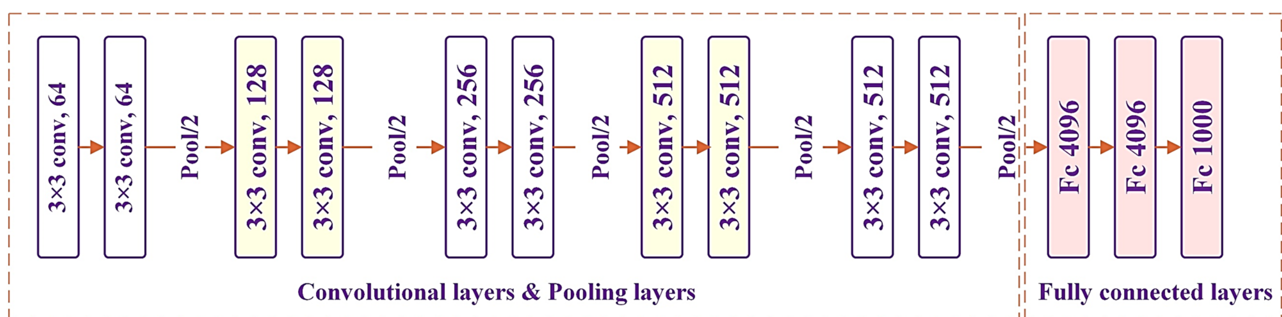

Figure 5. The diagram of VGG13 network architecture.

The ImageNet dataset used for pre-training contains 1000 categories. The output dimension of the pre-trained VGG13 network is 1000, while the hotspot-detection method has only two outputs. The VGG13 network trained with the ImageNet dataset cannot be used for hotspot detection directly. The convolutional layers and the pooling layers do not need to be modified. The dimension of the last fully connected layer is related to the number of output categories. Therefore, only the fully connected layers of the pre-trained VGG13 network need to be modified. The modified VGG13 network architecture is shown in Figure 6. The output dimension of the VGG13 network for hotspot detection is two, which corresponds to a hotspot pattern and non-hotspot pattern.

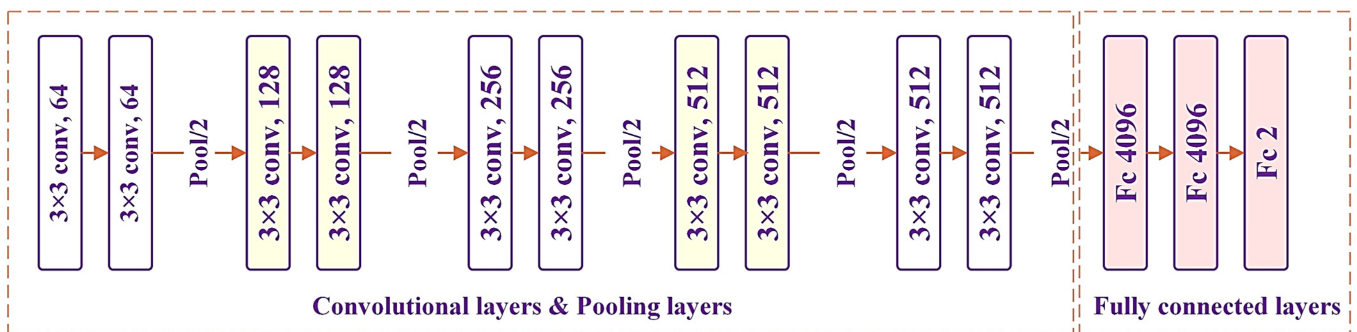

Figure 6. The diagram of VGG13 network architecture for hotspot detection.

2.3.2. Model Training

The proposed method takes cross entropy as the loss function. For an input layout pattern, the output result of the model is the probability distribution of each category. The labels of the training data are known, so the actual probability distribution for each layout pattern is fixed. The goal of model training is to reduce the gap between the probability distribution of model output and the actual probability distribution. Relative entropy is

a measure of the distance between two random distributions. The definition of relative entropy is shown in Equation (6), where n is the total number of data categories, p_i is the actual probability of the sample, and q_i is the model output probability of the sample. Since the labels of the training data are known and the actual probability distribution p_i is fixed, the first half of Equation (6) is constant. The definition of cross entropy is shown as Equation (7); relative entropy can be expressed as the sum of cross entropy and a constant. The gap between the probability distribution of model output and the actual probability distribution is narrowed by using cross entropy as the loss function. Hotspot detection is a binary classification problem; the cross entropy loss function of a single layout pattern is shown as Equation (8). For all data in a batch, the cross entropy loss is shown as Equation (9), where m is the number of samples in the batch. In the model-training phase, the cross entropy loss is reduced by adopting the batch gradient descent method.

$$D(p \parallel q) = \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right) = \sum_{i=1}^{n} p_i \log(p_i) - \sum_{i=1}^{n} p_i \log(q_i), \quad (6)$$

$$\text{cross entropy} = -\sum_{i=1}^{n} p_i \log(q_i), \quad (7)$$

$$\text{Loss}_{\text{binary}} = -[p_1 \log(q_1) + (1 - p_1)\log(1 - q_1)], \quad (8)$$

$$\text{Loss}_{\text{batch}} = -\frac{1}{m}\sum_{i=1}^{m}\left[p_1^i \log\left(q_1^i\right) + \left(1 - p_1^i\right)\log\left(1 - q_1^i\right)\right], \quad (9)$$

The proposed method takes the transfer learning strategy to apply the convolution kernels of the pre-trained VGG13 network to the hotspot detection. In the model-training phase, the convolutional layers of the pre-trained VGG13 network can be completely frozen or partially released. When the convolutional layers are partially released, the released convolution kernels will be updated. Due to the redesign of the fully connected layers, the weights of the fully connected layers need to be trained. For the proposed method, only partial layers of the VGG13 network need to be trained; the number of parameters required for training is reduced. When all the convolutional layers are frozen, the comparison between the number of parameters of the fully connected layers and the entire VGG13 network is shown in Figure 7.

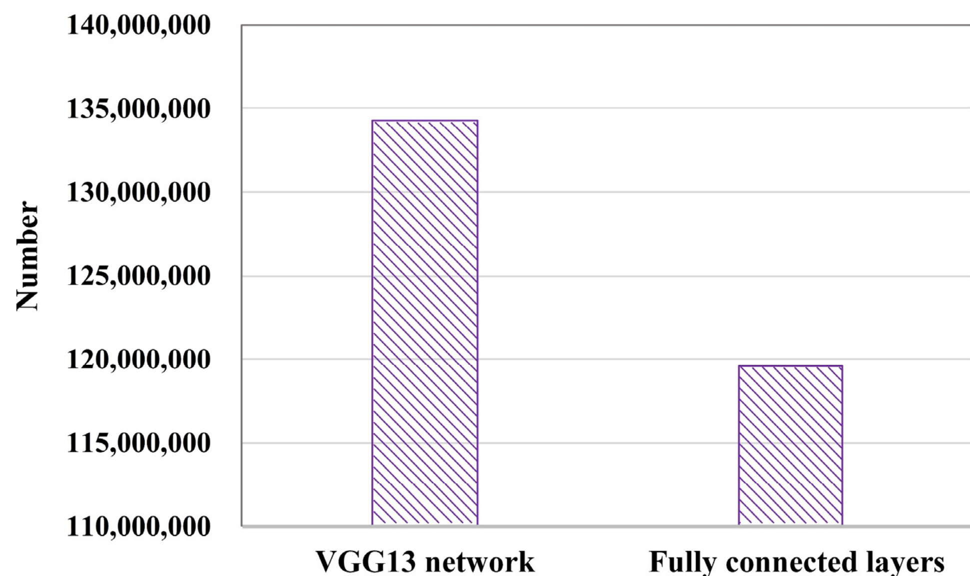

Figure 7. The number of parameters.

3. Results and Discussion

In order to verify the validity of the proposed method, the VGG13 network trained with the ImageNet dataset was used for the pre-trained model, and ICCAD 2012 benchmark suite [30] is used for model training and model verification. ICCAD 2012 benchmark suite consists of five benchmarks, each of which contains training data and testing data. The composition of ICCAD 2012 benchmark suite is shown in Table 1. Based on the data-compression method in the preparation phase, the layout data of ICCAD 2012 benchmark suite are converted to 240 × 240 pixels pattern data. According to the data-balance method, non-hotspot data are randomly sampled in the training data of ICCAD 2012 benchmark suite. The randomly sampled non-hotspot data and all hotspot data constitute a complete training dataset; the number of data in the training dataset is 5054. The model verification dataset is the combination of the testing data of ICCAD 2012 benchmark suite. Model training and model verification were performed on a server platform with Intel Xeon Gold 5118 CPU, 128 GB RAM, and Nvidia Tesla V100 GPU.

Table 1. The composition of ICCAD 2012 benchmark suite.

Benchmarks	Training Data		Testing Data	
	Hotspots	Non-Hotspots	Hotspots	Non-Hotspots
Benchmark 1 (B1)	99	340	224	319
Benchmark 2 (B2)	174	498	498	4146
Benchmark 3 (B3)	909	1808	1808	3541
Benchmark 4 (B4)	95	4452	177	3386
Benchmark 5 (B5)	26	2716	41	2111

The pre-trained VGG13 network was trained with the prepared training dataset. The training curves and validation curves are shown in Figure 8, after 10 epochs of training, the validation curves almost no longer change. After 10 epochs of training, the model is used for model verification to evaluate the performance. Samsung's hotspot-detection method based on deep CNN [16] (Shin's method) and the hotspot-detection methods based on transfer learning in the past two years [25,26] (Xiao's method and Zhou's method) are used as references. The hotspot-detection method of Ref. [25] is based on different workflow, and it is based on Inception-v3, ResNet50, and VGG16 networks. Additionally, the hotspot-detection method of Ref. [26] is based on the GoogLeNet network and different workflow. The results of accuracy, recall, precision, and F1 score are compared with the results of references. Since the convolutional layers can be partially released, model training and model verification were performed in the following situations:

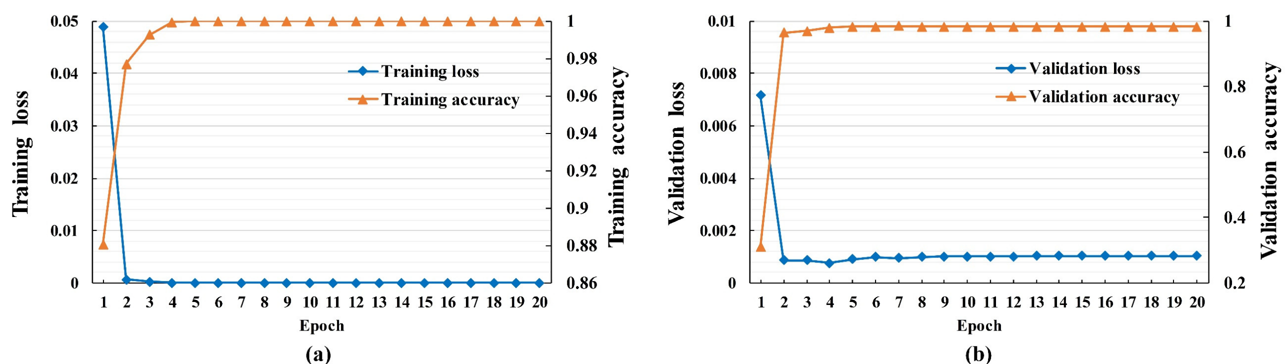

Figure 8. The loss and accuracy curves. (**a**) The loss and accuracy curves of training; (**b**) the loss and accuracy curves of validation.

(1) In the model-training phase, only the fully connected layers of the pre-trained VGG13 network were released. After 10 epochs of training, the model was evaluated.

(2) In the model-training phase, the fully connected layers and partial convolutional layers of the pre-trained VGG13 network were released. After 10 epochs of training, the model was evaluated.

3.1. All Convolutional Layers Are Frozen

In the model-training phase, only the fully connected layers of the pre-trained VGG13 network were released, so the weights of fully connected layers were updated. Since all the convolutional layers were frozen, all the weights of convolutional layers were applied to hotspot detection. After 10 epochs of training, the model was evaluated.

As shown in Table 2 and Figure 9, the results of the proposed method were compared with the results of references. Since only the recall, precision, and F1 score are given in the references, there is only the accuracy obtained by the proposed method in Figure 9a. Additionally, there are only the average performance of recall, precision, and F1 score in Ref. [25]. The proposed method shows good performance in accuracy; the accuracy is in the range of 96.1% to 99.3%. For the recall performance, the proposed method is not the best, but it is comparable to the results of references; the average recall of the proposed method reaches 97%. Compared with the results of references, there is significant improvement in the precision and F1 score of the proposed method. In the test results of five benchmarks, the precision performance of the proposed method is in the range of 72.4% to 96.8%. The average precision of the proposed method reaches 88.4%, while the results of all references are less than 48%. For the F1 score performance, the F1 score of the proposed method is in the range of 84% to 97.7%, and the average F1 score reaches 91.9%. Compared with the results of references, the proposed method obtains the much better F1 score performance. The proposed method obtains the best F1 score performance, which indicates that the proposed method has the best comprehensive performance in recall and precision. The improvement of precision and F1 score indicates that the proposed hotspot-detection method has low false-alarm rate. A low false-alarm rate will decrease the post-processing steps and decrease the turn-around time of IC manufacturing.

Table 2. The results of the proposed method and the references.

Benchmarks	Methods	Accuracy	Recall	Precision	F1 Score
Benchmark 1	Ref. [16]	-	0.951	0.358	0.520
	Ref. [26]	-	0.995	0.324	0.489
	The proposed method	0.961	0.938	0.968	0.952
Benchmark 2	Ref. [16]	-	0.988	0.216	0.354
	Ref. [26]	-	0.986	0.702	0.820
	The proposed method	0.979	0.988	0.843	0.901
Benchmark 3	Ref. [16]	-	0.975	0.199	0.331
	Ref. [26]	-	0.982	0.443	0.640
	The proposed method	0.984	0.988	0.966	0.977
Benchmark 4	Ref. [16]	-	0.938	0.157	0.269
	Ref. [26]	-	0.972	0.355	0.520
	The proposed method	0.992	0.932	0.918	0.925
Benchmark 5	Ref. [16]	-	0.927	0.181	0.303
	Ref. [26]	-	0.980	0.549	0.635
	The proposed method	0.993	1.000	0.724	0.840
Average	Ref. [16]	-	0.955	0.222	0.355
	Ref. [26]	-	0.983	0.475	0.635
	Ref. [25]	-	0.980	0.300	0.458
	The proposed method	0.982	0.970	0.884	0.919

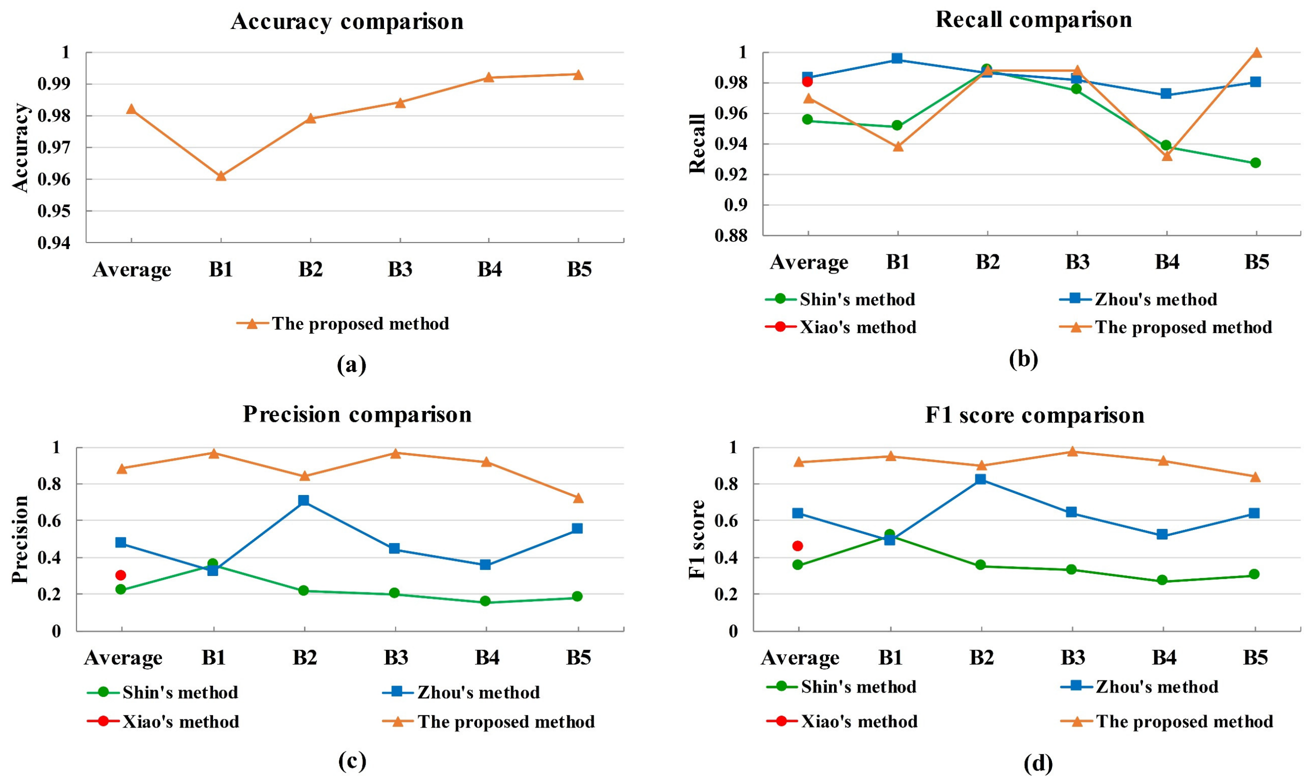

Figure 9. For the proposed method, all convolutional layers of the VGG13 network are frozen in the model-training phase. Comparison among the results of the proposed method and the references: (**a**) accuracy of the proposed method, (**b**) recall comparison, (**c**) precision comparison, and (**d**) F1 score comparison.

3.2. Partial Convolutional Layers Are Released

In the model-training phase, the fully connected layers and partial convolutional layers of the pre-trained VGG13 network were released, so that the weights of the released layers were updated. The more convolutional layers released, the more weights are updated. In order to test the effect of updating the weights of the convolutional layers on the results, partial convolutional layers were released for model training. Model training and model verification were carried out in the following situations:

Case 1: In the model-training phase, the fully connected layers and the last two convolutional layers were released. After 10 epochs of training, the model was evaluated.

Case 2: In the model-training phase, the fully connected layers and the last five convolutional layers were released. After 10 epochs of training, the model was evaluated.

The evaluation results are shown in Figures 10 and 11 and Table 3. The results are similar to the results in Section 3.1; the proposed method performs well in accuracy, recall, precision, and F1 score. For the test results of case 1, the average accuracy, recall, precision, and F1 score are 98.3%, 97.2%, 88.7%, and 92.4%. Compared with the results of freezing all convolutional layers, the results show that the performance was improved slightly. In case 2, more convolutional layers were released for model training. Compared with the results of freezing all convolutional layers, the improvement is not obvious. Comparison between the results of case 1 and case 2 shows that the evaluation results are similar. The improvement by releasing more convolutional layers for model training is not obvious.

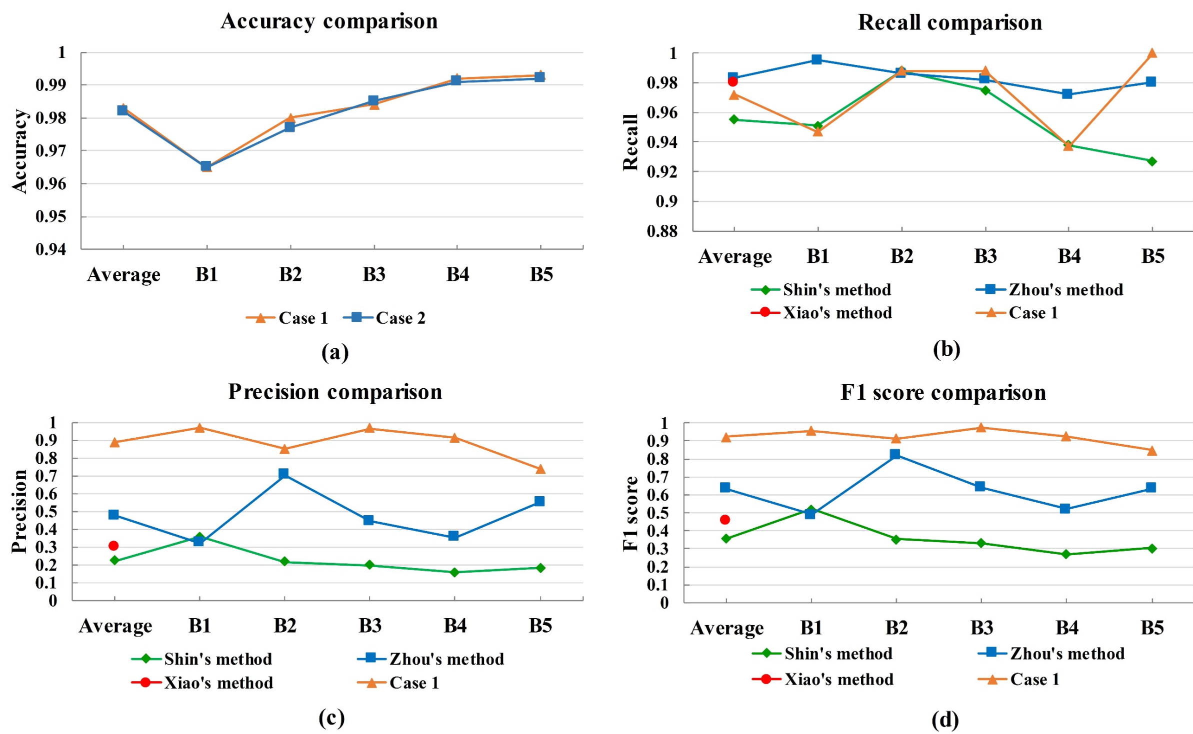

Figure 10. For the proposed method, the fully connected layers and the last two convolutional layers were released in model-training phase (case 1). Comparison among the results of case 1 and the references: (**a**) accuracy comparison, (**b**) recall comparison, (**c**) precision comparison, and (**d**) F1 score comparison.

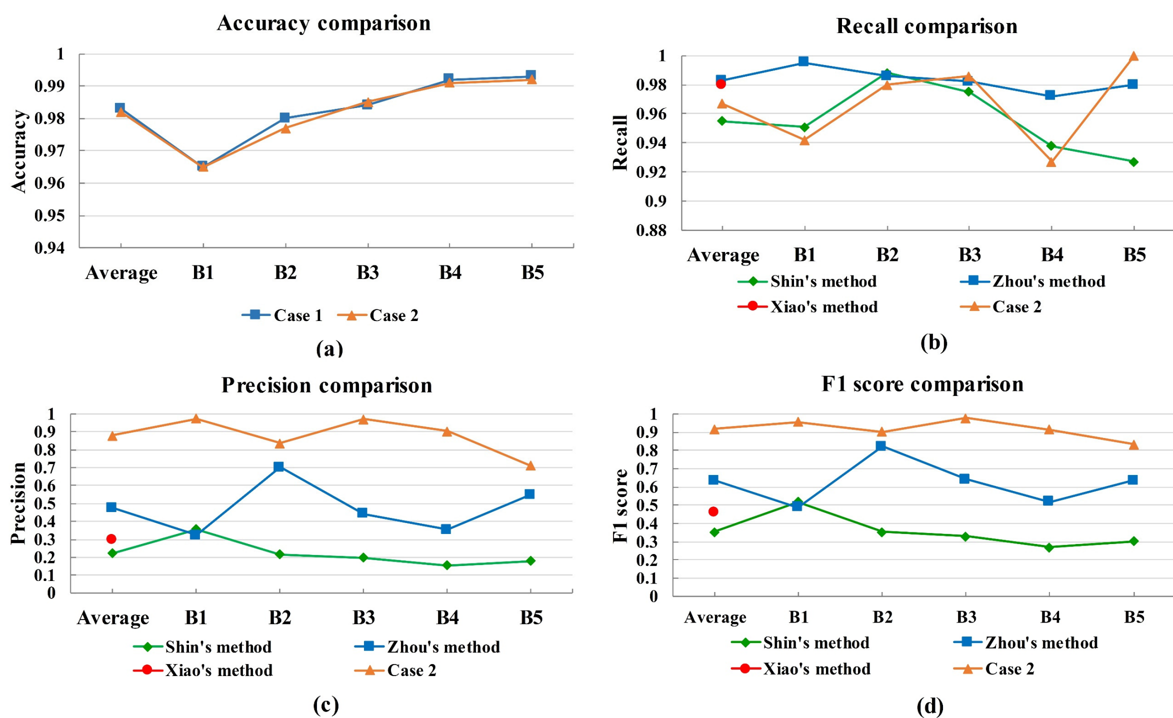

Figure 11. For the proposed method, the fully connected layers and the last five convolutional layers were released in model-training phase (case 2). Comparison among the results of case 2 and the references: (**a**) accuracy comparison, (**b**) recall comparison, (**c**) precision comparison, and (**d**) F1 score comparison.

Table 3. The results of case 1, case 2, and the references.

Benchmarks	Methods	Accuracy	Recall	Precision	F1 Score
Benchmark 1	Ref. [16]	-	0.951	0.358	0.520
	Ref. [26]	-	0.995	0.324	0.489
	Case 1	0.965	0.947	0.968	0.957
	Case 2	0.965	0.942	0.973	0.957
Benchmark 2	Ref. [16]	-	0.988	0.216	0.354
	Ref. [26]	-	0.986	0.702	0.820
	Case 1	0.980	0.988	0.850	0.914
	Case 2	0.977	0.980	0.837	0.903
Benchmark 3	Ref. [16]	-	0.975	0.199	0.331
	Ref. [26]	-	0.982	0.443	0.640
	Case 1	0.984	0.988	0.967	0.977
	Case 2	0.985	0.986	0.969	0.978
Benchmark 4	Ref. [16]	-	0.938	0.157	0.269
	Ref. [26]	-	0.972	0.355	0.520
	Case 1	0.992	0.937	0.913	0.925
	Case 2	0.991	0.927	0.903	0.915
Benchmark 5	Ref. [16]	-	0.927	0.181	0.303
	Ref. [26]	-	0.980	0.549	0.635
	Case 1	0.993	1.000	0.737	0.848
	Case 2	0.992	1.000	0.712	0.832
Average	Ref. [16]	-	0.955	0.222	0.355
	Ref. [26]	-	0.983	0.475	0.635
	Ref. [25]	-	0.980	0.300	0.458
	Case 1	0.983	0.972	0.887	0.924
	Case 2	0.982	0.967	0.879	0.917

4. Conclusions

In this paper, a lithography hotspot detection method based on transfer learning using pre-trained deep CNN has been proposed. The proposed method uses the VGG13 network trained with the ImageNet dataset as the pre-trained model. In order to obtain a model suitable for hotspot detection, the pre-trained model is trained with some down-sampled layout pattern data and takes cross entropy as the loss function. ICCAD 2012 benchmark suite is used for model training and model verification. Comparison with Samsung's hotspot-detection method based on deep CNNs and the hotspot-detection methods based on transfer learning in the past two years were carried out. The results show that the proposed method performs well in accuracy, recall, precision, and F1 score. There is significant improvement in the precision and F1 score. Compared with Samsung's deep CNN-based hotspot-detection method, the average precision and average F1 score are improved by 298% and 159%, respectively. The improvement of precision and F1 score indicates the proposed hotspot-detection method has a low false-alarm rate. A low false-alarm rate will decrease the post-processing steps and decrease the turn-around time of IC manufacturing. In order to test the effect of updating the weights of the convolutional layers on the results, partial convolutional layers were released for model training. Compared with freezing all convolutional layers, the results show that updating the weights of partial convolutional layers has little effect on the results of this method. The proposed method has the characteristics of low training cost, simple model architecture, and good performance. The proposed method is suitable for lithography hotspot detection.

Author Contributions: Conceptualization, L.L.; methodology, L.L.; software, L.L.; validation, L.L.; formal analysis, L.L.; investigation, L.L.; resources, L.L. and S.L.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, L.L., S.L., X.W., Y.C. and W.S.; visualization, L.L.; supervision and project administration, S.L. and X.W.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology Major Project of China (grant number: 2017ZX02101004-002, 2017ZX02101004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ImageNet dataset [29] of this research is available online: https://image-net.org/ (accessed on 31 December 2021). And the ICCAD 2012 benchmark suite is the dataset of ICCAD 2012 contest [30].

Acknowledgments: We would like to express our gratitude to Guodong Chen for his suggestions for revision of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Neisser, M. International Roadmap for Devices and Systems Lithography Roadmap. *J. Micro/Nanopattern. Mater. Metrol.* **2021**, *20*, 044601. [CrossRef]
2. Wikipedia: Transistor Count. Available online: https://en.wikipedia.org/wiki/Transistor_count (accessed on 31 December 2021).
3. Jochemsen, M.; Anunciado, R.; Timoshkov, V.; Hunsche, S.; Zhou, X.; Jones, C.; Callan, N. Process Window Limiting Hot Spot Monitoring for High-Volume Manufacturing. In Proceedings of the Metrology, Inspection, and Process Control for Microlithography XXX, San Jose, CA, USA, 21–25 February 2016; p. 97781R.
4. Hunsche, S.; Jochemsen, M.; Jain, V.; Zhou, X.; Chen, F.; Vellanki, V.; Spence, C.; Halder, S.; van den Heuvel, D.; Truffert, V. A New Paradigm for In-Line Detection and Control of Patterning Defects. In Proceedings of the Metrology, Inspection, and Process Control for Microlithography XXIX, San Jose, CA, USA, 22–26 February 2015; p. 94241B.
5. Nosato, H.; Sakanashi, H.; Takahashi, E.; Murakawa, M.; Matsunawa, T.; Maeda, S.; Tanaka, S.; Mimotogi, S. Hotspot Prevention and Detection Method Using An Image-Recognition Technique Based on Higher-Order Local Autocorrelation. *J. Micro/Nanolith. MEMS MOEMS* **2014**, *13*, 011007. [CrossRef]
6. Weisbuch, F.; Thaler, T.; Buttgereit, U.; Stötzel, C.; Zeuner, T. Improving ORC Methods and Hotspot Detection with The Usage of Aerial Image Metrology. In Proceedings of the Optical Microlithography XXXIII, San Jose, CA, USA, 23–27 February 2020; p. 113270D.
7. Kim, J.; Fan, M. Hotspot Detection on Post-OPC Layout Using Full-Chip Simulation-Based Verification Tool: A Case Study with Aerial Image Simulation. In Proceedings of the 23rd Annual BACUS Symposium on Photomask Technology, Monterey, CA, USA, 9–12 September 2003; pp. 919–925.
8. Gupta, P.; Kahng, A.B.; Nakagawa, S.; Shah, S.; Sharma, P. Lithography Simulation-Based Full-Chip Design Analyses. In Proceedings of the Design and Process Integration for Microelectronic Manufacturing IV, San Jose, CA, USA, 19–24 February 2006; p. 61560T.
9. Yao, H.; Sinha, S.; Chiang, C.; Hong, X.; Cai, Y. Efficient Process-Hotspot Detection Using Range Pattern Matching. In Proceedings of the 2006 IEEE/ACM International Conference on Computer Aided Design, San Jose, CA, USA, 5–9 November 2006; pp. 625–632.
10. Yang, F.; Sinha, S.; Chiang, C.C.; Zeng, X.; Zhou, D. Improved Tangent Space-Based Distance Metric for Lithographic Hotspot Classification. *IEEE Trans. Comput. -Aided Des. Integr. Circuits Syst.* **2017**, *36*, 1545–1556. [CrossRef]
11. Nagase, N.; Suzuki, K.; Takahashi, K.; Minemura, M.; Yamauchi, S.; Okada, T. Study of Hot Spot Detection Using Neural Network Judgment. In Proceedings of the Photomask and Next-Generation Lithography Mask Technology XIV, Yokohama, Japan, 17–19 April 2007; p. 66071B.
12. Ding, D.; Wu, X.; Ghosh, J.; Pan, D.Z. Machine Learning Based Lithographic Hotspot Detection with Critical-Feature Extraction and Classification. In Proceedings of the 2009 IEEE International Conference on IC Design and Technology, Austin, TX, USA, 18–20 May 2009; pp. 1–4.
13. Nakamura, S.; Matsunawa, T.; Kodama, C.; Urakami, T.; Furuta, N.; Kagaya, S.; Nojima, S.; Miyamoto, S. Clean Pattern Matching for Full Chip Verification. In Proceedings of the Design for Manufacturability through Design-Process Integration VI, San Jose, CA, USA, 14 March 2012; p. 83270T.
14. Yu, Y.-T.; Lin, G.-H.; Jiang, I.H.-R.; Chiang, C. Machine-Learning-Based Hotspot Detection Using Topological Classification and Critical Feature Extraction. In Proceedings of the 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 29 May–7 June 2013; pp. 1–6.
15. Duo, D.; Torres, J.A.; Pan, D.Z. High Performance Lithography Hotspot Detection With Successively Refined Pattern Identifications and Machine Learning. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2011**, *30*, 1621–1634. [CrossRef]
16. Shin, M.; Lee, J.-H. Accurate Lithography Hotspot Detection Using Deep Convolutional Neural Networks. *J. Micro/Nanolith. MEMS MOEMS* **2016**, *15*, 043507. [CrossRef]
17. Zhang, H.; Yang, H.; Yu, B.; Young, E.F.Y. VLSI Layout Hotspot Detection Based on Discriminative Feature Extraction. In Proceedings of the 2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Jeju, Korea, 25–28 October 2016; pp. 542–545.

18. Yu, B.; Gao, J.-R.; Ding, D.; Zeng, X.; Pan, D.Z. Accurate Lithography Hotspot Detection Based on Principal Component Analysis-Support Vector Machine Classifier with Hierarchical Data Clustering. *J. Micro/Nanolith. MEMS MOEMS* **2014**, *14*, 011003. [CrossRef]
19. Shin, M.; Lee, J.-H. CNN Based Lithography Hotspot Detection. *Int. J. Fuzzy Log. Intell. Syst.* **2016**, *16*, 208–215. [CrossRef]
20. Yang, H.; Lin, Y.; Yu, B.; Young, E.F.Y. Lithography Hotspot Detection: From Shallow to Deep Learning. In Proceedings of the 2017 30th IEEE International System-on-Chip Conference (SOCC), Munich, Germany, 5–8 September 2017; pp. 233–238.
21. Matsunawa, T.; Nojima, S.; Kotani, T. Automatic Layout Feature Extraction for Lithography Hotspot Detection Based on Deep Neural Network. In Proceedings of the Design-Process-Technology Co-optimization for Manufacturability X, San Jose, CA, USA, 16 March 2016; p. 97810H.
22. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [CrossRef]
23. Pesciullesi, G.; Schwaller, P.; Laino, T.; Reymond, J.L. Transfer Learning Enables The Molecular Transformer To Predict Regio- and Stereoselective Reactions on Carbohydrates. *Nat. Commun.* **2020**, *11*, 4874. [CrossRef] [PubMed]
24. Liu, E.; Yu, Z.; Wan, Z.; Shu, L.; Sun, K.; Gui, L.; Xu, K. Linearized Wideband and Multi-Carrier Link Based on TL-ANN. *Chin. Opt. Lett.* **2021**, *19*, 113901. [CrossRef]
25. Xiao, Y.; Huang, X. Learning Lithography Hotspot Detection from ImageNet. In Proceedings of the 2019 14th IEEE International Conference on Electronic Measurement & Instruments, Changsha, China, 1–3 November 2019; pp. 266–273.
26. Zhou, K.; Zhang, K.; Liu, J.; Liu, Y.; Liu, S.; Cao, G.; Zhu, J. An Imbalance Aware Lithography Hotspot Detection Method Based on HDAM and Pre-trained GoogLeNet. *Meas. Sci. Technol.* **2021**, *32*, 125008. [CrossRef]
27. Zhou, Z.-H. Linear Models. In *Maching Learning*, 1st ed.; Springer: Singapore, 2021; pp. 32–36.
28. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
29. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput.Vis.* **2015**, *115*, 211–252. [CrossRef]
30. Torres, J.A. ICCAD-2012 CAD Contest in Fuzzy Pattern Matching for Physical Verification and Benchmark Suite. In Proceedings of the 2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, 5–8 November 2012; pp. 349–350.
31. Yang, H.; Luo, L.; Su, J.; Lin, C.; Yu, B. Imbalance Aware Lithography Hotspot Detection: A Deep Learning Approach. *J. Micro/Nanolith. MEMS MOEMS* **2017**, *16*, 033504. [CrossRef]
32. Zhou, Z.-H. Model Selection and Evaluation. In *Maching Learning*, 1st ed.; Springer: Singapore, 2021; pp. 71–72.