applied sciences

MDPI

Article

Low-Quality Integrated Circuits Image Verification Based on Low-Rank Subspace Clustering with High-Frequency Texture Components

Guoliang Tan [1,2], Zexiao Liang [1], Yuan Chi [3,*], Qian Li [3,4,*], Bin Peng [5,*], Yuan Liu [1,2] and Jianzhong Li [1,2]

1 School of Integrated Circuits, Guangdong University of Technology, Guangzhou 510006, China
2 School of Automation, Guangdong University of Technology, Guangzhou 510006, China
3 The Fifth Electronics Research Institute of the Ministry of Industry and Information Technology of China, Guangzhou 510630, China
4 Guangdong Provincial Key Laboratory of Electronic Information Products Reliability Technology, Guangzhou 511370, China
5 Neuroeconomics Laboratory, Guangzhou Huashang College, Guangzhou 511399, China
* Correspondence: chiyuan4213@126.com (Y.C.); liqian.1234@163.com (Q.L.); 13822197681@139.com (B.P.)

Abstract: With the vigorous development of integrated circuit (IC) manufacturing, the harmfulness of defects and hardware Trojans is also rising. Therefore, chip verification becomes more and more important. At present, the accuracy of most existing chip verification methods depends on high-precision sample data of ICs. Paradoxically, it is more challenging to invent an efficient algorithm for high-precision noiseless data. Thus, we recently proposed a fusion clustering framework based on low-quality chip images named High-Frequency Low-Rank Subspace Clustering (HFLRSC), which can provide the data foundation for the verification task by effectively clustering those noisy and low-resolution partial images of multiple target ICs into the correct categories. The first step of the framework is to extract high-frequency texture components. Subsequently, the extracted texture components will be integrated into subspace learning so that the algorithm can not only learn the low-rank space but also retain high-frequency information with texture characteristics. In comparison with the benchmark and state-of-the-art method, the presented approach can more effectively process simulation low-quality IC images and achieve better performance.

Keywords: low-quality data; integrated circuits; high-frequency texture component; subspace clustering

check for updates

Citation: Tan, G.; Liang, Z.; Chi, Y.; Li, Q.; Peng, B.; Liu, Y.; Li, J. Low-Quality Integrated Circuits Image Verification Based on Low-Rank Subspace Clustering with High-Frequency Texture Components. *Appl. Sci.* **2023**, *13*, 155. https://doi.org/10.3390/app13010155

Academic Editor: Bing Li

Received: 16 November 2022
Revised: 7 December 2022
Accepted: 19 December 2022
Published: 22 December 2022

1. Introduction

Machine learning and the integrated circuit (IC) manufacturing industry have advanced rapidly in the last decade. The expansion of the IC industry especially has led to a new pursuit of efficiency in all aspects. Verification is one of the core steps in the chip design and production process, so traditional inspection methods needed to be replaced by new automated inspection means urgently. Therefore, efficient verification techniques based on computer vision or machine learning algorithms have received a lot of attention in recent years [1–4]. In the proposed process of such verification techniques, their validity is mostly derived from the high-quality ICs' images being utilized as a test set. The goal of this paper is to suggest an algorithm for the effective classification of low-quality ICs' images in order to reduce the stringent requirements on samples and provide a reliable, valid, and easily accessible database for the subsequent verification procedures.

With annual semiconductor sales exceeding USD 400 billion for the first time in 2018 [5], the need for reliability verification in the semiconductor industry has reached a new pinnacle, and there are two common types of defect objects in reliability verification, which are unintentional defects and malicious hardware Trojans (HTs). Unintentional defects can be caused by immature processes that result in low yields or poor processing environments, such as solder joint offsets and stained spots. These defects are not the original design intent, so they can be solved by improving the process and production environment. In contrast, HTs are human-implanted and can exist in two important stages of IC production, including design and manufacturing. Even more problematic is the fact that today's IC production is a global collaborative model, with various stages of the industry scattered around the globe, making it easy for HTs to break in. HTs can leak secret information, disable ICs, or cause other catastrophic consequences [6–8]. In 2009, a statistical approach was proposed with the idea of constructing a series of fingerprints based on the path delay of gold chips to verify whether the chips contain HTs by comparing the delay parameters of the chips with the fingerprints [9]. With the booming development of computer vision, vision-based techniques are one of the hot research topics in the last decade. A machine learning algorithm based on power analysis was proposed in [10], which aims to obtain the hardware Trojan infection algorithm and use machine learning to detect hardware Trojan implantation. One HT classification method is called Hardware Trojan Learning Analysis (ATLAS) [11], which identifies HT-infected circuits using a gradient boosting (GB) model on data from the gate-level netlist (GLN) stage. A method is proposed to extract the basic features of digital circuits using Haar features and augment their features with the AdaBoost learning algorithm to generate strong classifiers [12]. Two hardware Trojan detection algorithms based on support vector machine (SVM)-supervised machine learning were designed [13,14]. In [15], the intrinsic nature of many cores providing feedback to the online learning algorithm based on the core information and its behavior with incoming packets was designed for a real-time online learning method. As a latecomer, deep learning has also made a lot of achievements in this field in recent years. A deep learning (DL) class of ML was proposed based on extracting the raw circuit features and feeding them into a DL model (deep stacking autoencoder) that extracts the features which contribute to HT detection [16]. A novel, golden, reference-free HT detection method named GNN4TJ [17] was proposed in the register transfer level (RTL) based on a graph neural network (GNN). A deep learning model, GATE Net, based on graph convolutional networks (GCNs), was proposed in [18], which is trained using supervised contrast learning for labeling designs containing random insertion triggers using only the corresponding netlist. A new model-based dynamic convolution neural network (DCNN) architecture was developed in [19], which includes three convolutional and pooling layers for automatically extracting invariant and relevant features from IC images. A new graph learning (GL)-based method for detecting node HTs, called NHTD-GL, was proposed in [20].

As more and more companies join the chip manufacturing sector, the need for flow and verification of chip designs after they are completed is also on the rise. However, unlike large companies, which have the financial resources to spend large sums of money on scanning electron microscopes (SEMs) to facilitate the subsequent verification process, small- and medium-sized companies are forced to outsource this step to professional facilities, albeit at a high cost. In essence, this is because the availability of validation tools is based on high-quality IC images. This creates a contradiction: the high difficulty of obtaining high-quality images and the indispensability of noiseless high-quality IC images. This contradiction is also one of the biggest obstacles between academia and industry. A low-cost and easily achievable alternative is, therefore, worth exploring.

The aim of this paper is to propose an algorithm for the recovery of low-quality image information for subsequent detection tasks. As the approach is pragmatic, the data used in the experiments will take into account the following practices and common situations:

(1) Unlike SEM, which uses a narrowly focused high-energy electron beam to scan the sample and sweep the image elements from left to right and from top to bottom on a sample to obtain a high-precision, full-range image in a single pass, the workflow of a normal camera with a microscope is still to capture different partial views several times and finally combine them into a complete large image.

(2) For the sensor on the camera, the unstable temperature and light source of the acquisition environment can bring about changes in the signal-to-noise ratio (SNR), and images with low SNRs often contain a variety of noises.

(3) Image data may be corrupted when uploading from the camera to the server, such as due to multiplicative noise caused by unsatisfactory channels.

To cope with the images obtained in the above scenario, this paper proposes a fused representation clustering framework called High-Frequency Low-Rank Subspace Clustering (HFLRSC), where high frequency means the High-Frequency Texture Component (HFTC). In HFLRSC, A fused subspace learning framework that incorporates an HFTC and low-rank constraints is proposed. Experiments on simulated data show that the method can effectively process low-quality IC images and achieve better performance than the benchmark and existing methods.

2. Background Knowledge

This section will briefly introduce the related background knowledge on low-rank representation learning and the high-frequency texture component.

2.1. Low-Rank Representation

Due to the problem of the signal-to-noise ratio, the current data acquisition methods will inevitably cause varying degrees of noise pollution in the datasets. To reconstruct a low-rank matrix from polluted data, LRR was developed. Given a dataset $\boldsymbol{X}$ that has been polluted by errors $\boldsymbol{P}$, $\boldsymbol{X}$ can be thought of as a mixture of two parts ($\boldsymbol{X} = \boldsymbol{U} + \boldsymbol{P}$), where the raw data $\boldsymbol{U}$ should be of a low rank. In order to achieve the above objectives, the subsequent regularization rank minimization equation could be formulated as

$$\min_{\boldsymbol{U},\boldsymbol{P}} \; rank(\boldsymbol{U}) + \lambda \|\boldsymbol{P}\|_l \quad s.t. \quad \boldsymbol{X} = \boldsymbol{U} + \boldsymbol{P}, \tag{1}$$

where λ is a trade-off parameter and $\|\cdot\|_l$ also indicates a specific regularization technique for the error term.

After introducing an extra dictionary matrix, the more general rank minimization problem can be applied to more real mixed data as follows:

$$\min_{\boldsymbol{C},\boldsymbol{P}} \; rank(\boldsymbol{C}) + \lambda \|\boldsymbol{P}\|_l \quad s.t. \quad \boldsymbol{X} = \boldsymbol{U}\boldsymbol{C} + \boldsymbol{P}, \tag{2}$$

where $\boldsymbol{U}$ is a dictionary. Under the circumstances, the minimum solution of $\boldsymbol{C}$ which can be abbreviated as $\boldsymbol{C}^*$ is the "lowest rank representation" of $\boldsymbol{X}$. In the process of practical application, it is difficult to build an appropriate dictionary. Therefore, the self-representation method of using itself [21] as a dictionary came into being. At this time, $\boldsymbol{C}^*$ represents the sample's subspace knowledge and can be used for clustering.

2.2. High-Frequency Texture Component

In general, there are two important components constructing the information of the image: a texture component and a structure component. The morphological information of an image is represented by the texture component, while the structure component represents the structure. Therefore, the essential characteristic of the image can be revealed well by the texture component. The texture component knowledge can be useful for rebuilding images using compressive sensing algorithms.

The image texture component is the high-frequency information, which is easy to achieve. Among our pretests, we successfully propose an effective method to extract the morphological characteristic of human face images and utilize it to distinguish different individuals. In the IC image, the key information is the shapes, layouts, and wiring of its electronic components. The edges of elements will create the pixel change regions in the images, which can also be effectively captured and extracted by the high-pass filter, and the high-pass filter can effectively capture and extract the pixel change region generated by the edges of electronic components. Even if the resolution of IC images is low and full of noise, the features extracted by HFTC are still enough for classification and the following procedures.

3. The Proposed Approach

The motivation and architecture of HFLRSC will be discussed in detail in this section.

3.1. Proposed Framework

In just about all instances, conventional clustering approaches would consider all of the data's knowledge during the procedure, which is effective for noiseless data. However, data in the real world are often noised, and the noise may be non-sparse or of a high intensity. The noise may form some pixel blocks in the samples and cause those samples to lose their features. Thus, traditional methods would probably categorize those noised samples into the same cluster. Inspired by our pretests, the HFTC features extracted from face images can maintain valuable morphological information well and greatly weaken the noise, we believe that HFTC extraction can make better use of the key knowledge of low-quality IC images. To be specific, the valuable information of the ICs' images exists in the layout and wiring of the electronic components in the field of IC defect detection (visual detection methods). While the layout and wiring can be effectively retained by HFTC extraction. In the meantime, the impact of environmental noise and pixel loss will be significantly reduced in the extraction process, which is very conducive to the subsequent clustering tasks.

In machine learning tasks, each individual sample in the dataset is usually assembled into a column as $\boldsymbol{X} = \{\boldsymbol{x}_i | i = 1, \ldots, N\}$. Each individual sample, designated by the letter $\boldsymbol{x}_i$, must be reshaped during the extraction process to its original height and width, designated by $\boldsymbol{X}_i$. In order to facilitate the following discussion, the height and width of $\boldsymbol{X}_i$ are represented by H and W, respectively. Then, $\hat{\boldsymbol{X}}_i$ can be made to represent the expression of data $\boldsymbol{X}_i$ in the frequency domain, which is derived by using a two-dimensional Fourier transform (2D FFT):

$$\hat{\boldsymbol{X}}_i(s,t) = \sum_{m=1}^{H} \sum_{n=1}^{W} \boldsymbol{x}_i(m,n) e^{-j2\pi(\frac{ms}{H} + \frac{nt}{W})}, \tag{3}$$

where (s,t) and(m,n) denote the coordinates. Then, the high-frequency texture components of the images $\hat{\boldsymbol{h}_1}, \cdots, \hat{\boldsymbol{h}_n}$ are obtained by using a high-pass filter $\mathscr{H}$ to process frequency domain data and apply an inverse two-dimensional FFT to the filtered frequency domain data:

$$\hat{\boldsymbol{h}}_i(m,n) = \frac{1}{H}\frac{1}{W} \sum_{s=1}^{H} \sum_{t=1}^{W} \mathscr{H} \hat{\boldsymbol{X}}_i(m,n) e^{j2\pi(\frac{ms}{H} + \frac{nt}{W})}. \tag{4}$$

Then, $\hat{\boldsymbol{h}}_i$ is restructured as a column vector $\boldsymbol{h}_i$ and combined to form the matrix $H = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_N\} \in \mathbb{R}^{d \times n}$.

The high-frequency texture components retain most morphological information, which can reveal the essential details for better distinguishing individuals. When the data are noised, the traditional clustering methods may not work as usual, particularly when the noise is not sparse and has a high energy level. Using IC images as examples, when distinct circuit images are obscured by Gaussian noise or flaws, they may be classified as part of the same cluster because of the high energy of the covers. The noise described above is low-frequency information that can be considerably decreased following HFTC extraction, from which it can be concluded that using an HFTC will assist representation learning and increase subsequent clustering performance.

Rather than sharing an entire uniform space, the samples of dataset X are typically located in their distinct low-dimensional subspaces. The LRR problem [22] can be reformulated using the self-representation method:

$$\min_{\boldsymbol{C}} \ rank(\boldsymbol{C}) + \lambda \|\boldsymbol{XC} - \boldsymbol{X}\|_l \tag{5}$$

where $||\boldsymbol{XC} - \boldsymbol{X}\|_l$ represents the sparse error measurement. Using the knowledge of the high-frequency texture component matrix $\boldsymbol{H}$ and a Frobenius norm to estimate the error between the texture component and the original data, the LRR formula can be reformulated as follows:

$$\min_{\boldsymbol{C}} \ rank(\boldsymbol{C}) + \lambda_1 \|\boldsymbol{XC} - \boldsymbol{H}\|_F^2 + \lambda_2 \|\boldsymbol{XC} - \boldsymbol{X}\|_l \tag{6}$$

where λ_1 and λ_2 are the trade-off parameters. In the iterative process, the algorithm can learn the unique representation of $\boldsymbol{C}$ by integrating all aspects of the knowledge. Since the rank $(\cdot)$ operator is a non-convex problem, a convex kernel norm is typically used to relax it [23]. As a matter of convenience, $\boldsymbol{P}$ can be used to represent the data's own error term. Finally, High-Frequency Low-Rank Subspace Clustering (HFLRSC) is proposed, which is written as a minimization problem:

$$\min_{\boldsymbol{C},\boldsymbol{P}} \ \|\boldsymbol{C}\|_* + \lambda_1 \|\boldsymbol{XC} - \boldsymbol{H}\|_F^2 + \lambda_2 \|\boldsymbol{P}\|_l \quad s.t. \quad \boldsymbol{X} = \boldsymbol{XC} + \boldsymbol{P} \tag{7}$$

As previously described, λ_1 and λ_2 are the trade-off parameters. The nuclear norm and the Frobenius norm are denoted by $\|\cdot\|_*$ and $\|\cdot\|_F$, respectively. Within this framework, the $\|\cdot\|_F$ function is used to estimate the distance between the self-representation term $\boldsymbol{XC}$ and the texture component term $\boldsymbol{H}$. Subspace $\boldsymbol{C}$ will progressively acquire the information stored in the texture components during the iterative procedure of minimizing the term $\|\boldsymbol{XC} - \boldsymbol{H}\|_F^2$. Because of the sparse error and low-rank term limitations, the learning results will attain a relative balance while modifying the trade-off parameters.

When $\boldsymbol{C}$ is obtained, the category to which each sample belongs can be revealed by the subspace structure. With $\boldsymbol{C}$, the affinity matrix could be generated as follows:

$$\boldsymbol{W} = \frac{|\boldsymbol{C}| + |\boldsymbol{C}^{\boldsymbol{T}}|}{2} \tag{8}$$

where $|\cdot|$ is the absolute operator. The final clustering results can be acquired using the standard Special Clustering (SC) approach by using the generated w matrix as illustrated below:

$$\min_{\boldsymbol{F}} Tr(\boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{LF}) \quad s.t. \ \boldsymbol{F}^{\boldsymbol{T}}\boldsymbol{DF} = \boldsymbol{I} \tag{9}$$

where $Tr(\cdot)$ denotes the trace of the matrix. $\boldsymbol{F}$ denotes the cluster indicator matrix, while $\boldsymbol{I}$ denotes the identity matrix. $\boldsymbol{L}$ is the Laplacian matrix, and $\boldsymbol{D}$ is a diagonal matrix whose elements are the sum of the ith row or ith column of $\boldsymbol{W}$. Thus, Equation (9) can be effectively transformed into an eigenvalue decomposition problem, and the clustering index of the data points can be directly expressed by a matrix $\boldsymbol{F}$.

3.2. Solution to HFLRSC

This section will provide a brief overview of the HFLRSC solution and essential steps. The optimization problem in Equation (7) is convex and can be solved by various methods. For efficiency, we adopt in this paper the Augmented Lagrange Multiplier (ALM) [24] method. We first convert Equation (7) to the following equivalent problem:

$$\min_{\boldsymbol{C},\boldsymbol{R},\boldsymbol{P}} \ \|\boldsymbol{C}\|_* + \lambda_1 \|\boldsymbol{XR} - \boldsymbol{H}\|_F^2 + \lambda_2 \boldsymbol{\Omega}(\boldsymbol{P}) \quad s.t. \quad \boldsymbol{X} = \boldsymbol{XR} + \boldsymbol{P}, \boldsymbol{R} = \boldsymbol{C} \tag{10}$$

The problem can be solved by the ALM method, and the objective function using the ALM method can be written as

$$\begin{aligned} \boldsymbol{L} = & \|\boldsymbol{C}\|_* + \lambda_1 \|\boldsymbol{XR} - \boldsymbol{H}\|_F^2 + \lambda_2 \boldsymbol{\Omega}(\boldsymbol{P}) + \\ & \frac{\mu}{2}\left(\|\boldsymbol{XR} + \boldsymbol{P} - \boldsymbol{X} + \frac{\beta_1}{\mu}\|_F^2 + \|\boldsymbol{C} - \boldsymbol{R} + \frac{\beta_2}{\mu}\|_F^2 \right) \end{aligned} \tag{11}$$

The above problem is unconstrained. Thus, it can be minimized with respect to $\boldsymbol{C}$, $\boldsymbol{R}$, and $\boldsymbol{P}$ by fixing the other variables and then updating the Lagrange multipliers β_1 and β_2, where $\mu > 0$ is a penalty parameter. The inexact ALM method, also called the alternating direction method, is used in HFLRSC and is outlined in Algorithm 1.

In Algorithm 1, the subscript t indicates the index of iteration. $\Theta(a,b) = \boldsymbol{U}\mathcal{S}_b(\Sigma_a)\boldsymbol{V}^T$ is the singular value thresholding (SVT) operator, in which $\boldsymbol{U}\Sigma_a\boldsymbol{V}^T$ is the singular value decomposition (SVD) of a and $\mathcal{S}_b(\boldsymbol{X}) = Sign(\boldsymbol{X}) * max(|\boldsymbol{X}| - q, 0)$ is the soft threshold operator.

Algorithm 1: The algorithm of HFLRSC.

Input: Data $\boldsymbol{X} = \{\boldsymbol{x}_i | i = 1, \ldots, N\}$, trade-off parameters λ_1 and λ_2, Butterworth filter order n_1 and cut-off frequency D_0.

Output: Self-representation matrix $\boldsymbol{C}$

Initialize filter by parameters n_1 and D_0 ;

Extract the high-frequency texture components of datasets by (4);

while *not convergence* **do**

Update $\boldsymbol{C}$ by $\boldsymbol{C}_{t+1} = \Theta((\boldsymbol{R} - \frac{\beta_2}{\mu})t, \frac{1}{\mu})$;

Update $\boldsymbol{R}$ by $\boldsymbol{R}^*_{t+1} =$

$((2\lambda_1 + \mu)\boldsymbol{X}^T\boldsymbol{X} + \mu * I)^{-1}(2\lambda_1\boldsymbol{X}^T\boldsymbol{H} + \mu\boldsymbol{X}^T(\boldsymbol{P} - \boldsymbol{X} - \frac{\beta_1}{\mu})t + \mu(\boldsymbol{C} + \frac{\beta_2}{\mu}t))$;

Update $\boldsymbol{P}$ by $\boldsymbol{P}_{t+1} = \mathcal{S}_{\frac{\lambda_2}{\mu}}((-\boldsymbol{X}\boldsymbol{R} + \boldsymbol{X} - \frac{\beta_1}{\mu})t)$;

Update parameters $\begin{cases} \beta_1 = \beta_1 + \mu(\boldsymbol{X}\boldsymbol{R} + \boldsymbol{P} - \boldsymbol{X}) \\ \beta_2 = \beta_2 + \mu(\boldsymbol{C} - \boldsymbol{R}) \\ \mu = \mu * \rho \end{cases}$

end

4. Experimental Study

In this section, we conduct extensive experiments on simulation datasets to verify the effectiveness of the proposed approach. All experimental data were obtained from a real-world scanning electron microscope at different settings.

4.1. Data Preparation

In the implementation, we chose to use simulated samples obtained by processing high-quality images from the real world due to several reasons. First of all, it is hard to access sufficient quantities of the desired low-resolution equipment. Secondly, the high-quality image used in this experiment was taken by a scanning electron microscope, meaning that it should be able to represent the highest-quality chip image. Therefore, the down-sampling of high-quality images in different proportions is effective for simulating different conditions for low-resolution devices. In addition, the simultaneous interpreting of different sizes and intensities of the image is used to simulate the sizes of different sensors, because the size of the sensor is positively correlated with the signal-to-noise ratio (SNR). Based on the above discussion, we entrusted an authoritative research institution, named the Science and Technology on Reliability Physics and Application of Electronic Component Laboratory, with the high-quality chip images scanned by an electron microscope. Based on the principle of confidentiality of information and the frequency with which the chip was attacked, the research institution provided a real image of the chip. Figure 1 shows a general view of this chip.

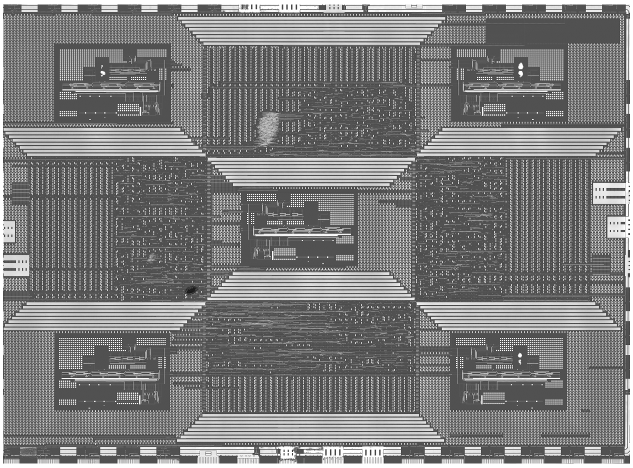
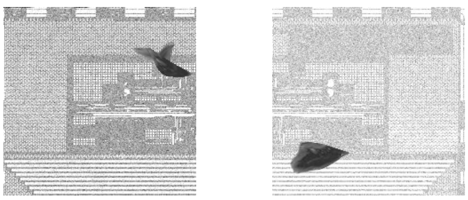


Figure 1. A scanning electron microscope image of the chip with a resolution of 6960 × 5084.

There was one more major processing step before we proceeded with the clustering experiment, which was histogram equalization. This method is generally utilized to increase the global contrast of images, especially when the contrast of the valid data of the image is quite close. Through this method, the brightness can be better distributed on the histogram. In the actual production process, it is difficult to obtain timely and high-quality chip images to analyze whether there are hardware Trojans and defects, which are caused by equipment accuracy, environmental interference, and operation error. In order to solve the above shackles, different partial views of the chip can be scanned multiple times through a low-precision lens, and all partial views can be reassembled into a complete chip image.

When obtaining experimental data, the low-quality image would be simulated through the following three steps. First, we divided the full-view pictures into several parts and further controlled the resolution by reducing the sampling proportionally. Secondly, the defects of different sizes and numbers extracted from real chip images were covered on the simulated data samples. Lastly, we added different strengths of Gaussian noise, salt-and-pepper noise, and multiplicative noise to simulate environmental interference. Figure 2 presents a part of the simulated low-quality images of ICs. In this experiment, a complete chip image would be divided into 12 parts, and each part would be divided into 10 categories by adding defective blocks at random locations and the same noise intensity. Therefore, the dataset size of this experiment was 10 categories and 120 samples in total.



(**a**) (**b**)

Figure 2. Data samples corrupted by Gaussian noise with two different types of intensities and simulated defects. (**a**) sample a; (**b**) sample b.

Next is the explanation of the noise parameters. "Gaussian 0.1" means adding Gaussian white noise, where the mean value is 0.1 and the variance is 0.01 by default. "Multiplicative noise 0.1" means adding multiplicative noise using equation $J = I + n * I$, where I is the original image and N is the uniformly distributed random noise with a mean value of 0 and variance of 0.1 by default. The range of the mean and variance parameters of Gaussian noise and multiplicative noise types is [0, 1]. "Salt-and-Pepper 0.1" means adding salt and pepper noise, where 0.1 is the noise density, which means it will affect about $0.1 * numel(I)$ pixels.

4.2. Comparison Methods

In this experiment, we chose K-means and spectral clustering with normalized cut (Ncut) for the baseline. Classical representation learning algorithms including Sparse Subspace Clustering (SSC) [25] and Low-Rank Representation (LRR) [23] were also adopted for comparison. We also added two recent clustering algorithms for comparison: SC-SRGF [26] and USENC [27].

Next, the five general performance indicators selected in this section are introduced.The accuracy (ACC) was calculated using the Hungarian algorithm to best map among the ground true labels and cluster results. The purity and accuracy in classification problems are similar. It takes the most classes in each cluster to be the class represented by this cluster, calculates the number of correctly allocated classes, and then divides this number by the total to obtain a value. The normalized mutual information (NMI) is used to measure the correlation between data distribution. The higher the mutual information, the higher the correlation. The F-measure (F) is a combination of precision and recall in information retrieval for clustering evaluation. The adjusted Rand index (ARI) indicates the similarity between the two clusters.

4.3. Results and Analyses

In this section, we will conduct clustering of the simulated datasets with different degrees and types of noise under the condition of the same pixel loss. In particular, we fixed the number of defects at 10, which was to simulate the chip with a small number of defects in the general scenario, as too many or too little would appear impractical.

Table 1 lists the set-ups for the objective datasets and the indexes of those set-ups. Table 2 presents the clustering results with the five performance indicators mentioned above. The results in the table clearly show that HFLRSC not only achieved the best results in all datasets but also achieved 100% excellent performance in five indicators in half of the datasets. With the increase in noise intensity, the performance of the general methods would fall strongly. Specifically, in the process of Gaussian noise rising from 0.1 to 0.6, the accuracy of NCUT and K-means decreased significantly by 64.87% and 53.48%, respectively. The accuracy of HFLRSC was very stable, and even the lowest could reach a 92.62% result, which shows that our method has a very obvious effect on noise suppression and can effectively extract valuable information from low-quality IC images.

In order to make this conclusion more intuitive, we drew a broken line graph as shown in Figure 3, in which the accuracy of each algorithm changes with the transformation of Gaussian noise.

Table 1. Simulated datasets with different settings.

Index	Noise	Defects
1	Gaussian 0.1	10
2	Gaussian 0.2	10
3	Gaussian 0.4	10
4	Gaussian 0.6	10
5	Salt-and-Pepper 0.1	10
6	Salt-and-Pepper 0.2	10
7	Salt-and-Pepper 0.4	10
8	Salt-and-Pepper 0.6	10
9	Multiplicative 0.01	10
10	Multiplicative 0.1	10
11	Multiplicative 0.15	10
12	Multiplicative 0.2	10

Table 2. The clustering performances of the proposed approach and the comparison methods for the simulated datasets (1–6). The bold denotes the best results.

Datasets	Methods	ACC	NMI	Purity	F	ARI
1	HFLRSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	NCut	88.20	94.31	90.17	88.11	87.01
	K-means	73.94	85.24	76.54	73.46	70.79
	LRR	97.97	99.16	98.50	98.11	97.95
	SSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	SC-SRGF	92.22	96.58	94.02	92.54	91.89
	USENC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
2	HFLRSC	**98.33**	**97.75**	**98.33**	**96.49**	**96.20**
	NCut	87.03	93.32	89.50	86.69	85.49
	K-means	68.03	79.23	70.63	64.85	61.20
	LRR	97.42	98.98	98.17	97.65	97.45
	SSC	93.33	92.33	93.33	86.47	85.34
	SC-SRGF	88.33	95.35	91.67	89.27	88.33
	USENC	93.13	95.95	93.88	92.33	91.64
3	HFLRSC	**93.40**	**91.37**	**93.40**	**86.47**	**85.36**
	NCut	43.05	47.87	44.77	28.59	22.31
	K-means	36.75	41.23	38.78	26.42	18.13
	LRR	80.81	79.46	81.13	66.37	63.50
	SSC	73.50	67.50	73.92	48.18	43.20
	SC-SRGF	64.15	63.65	65.97	46.61	42.06
	USENC	49.17	53.75	50.58	35.59	29.90
4	HFLRSC	**92.62**	**90.72**	**92.62**	**83.74**	**82.37**
	NCut	23.42	25.77	24.60	12.17	2.75
	K-means	20.46	19.53	21.45	11.62	0.00
	LRR	37.12	38.93	38.85	19.50	12.31
	SSC	27.17	32.20	28.83	13.44	5.53
	SC-SRGF	38.90	39.82	40.83	22.05	15.26
	USENC	26.46	28.86	27.58	14.47	5.43
5	HFLRSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	NCut	90.92	95.91	92.67	91.37	90.59
	K-means	71.10	82.77	73.69	69.34	66.18
	LRR	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	SSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	SC-SRGF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	USENC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
6	HFLRSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	NCut	88.65	94.13	90.45	88.40	87.35
	K-means	68.06	78.75	70.55	64.68	61.15
	LRR	95.67	98.23	96.83	96.01	95.66
	SSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	SC-SRGF	94.72	94.92	95.02	91.77	91.08
	USENC	96.63	98.22	97.04	96.40	96.08
7	HFLRSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	NCut	78.98	86.87	81.40	76.16	74.02
	K-means	59.77	64.94	62.67	49.17	44.34
	LRR	92.08	95.83	93.70	91.57	90.83
	SSC	96.58	96.41	96.58	93.40	92.86
	SC-SRGF	90.42	92.94	91.55	87.14	86.05
	USENC	90.29	93.79	91.25	88.48	87.48

Table 2. *Cont.*

Datasets	Methods	ACC	NMI	Purity	F	ARI
8	HFLRSC	**95.23**	**93.75**	**95.23**	**89.61**	**88.75**
	NCut	59.30	66.47	61.17	49.10	44.68
	K-means	48.69	50.90	51.68	32.89	26.73
	LRR	85.78	86.94	85.83	76.80	74.85
	SSC	75.50	73.28	75.50	52.70	48.12
	SC-SRGF	71.47	71.77	72.40	55.89	52.17
	USENC	54.67	57.16	56.46	37.81	32.37
9	HFLRSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	NCut	91.37	95.97	92.83	91.63	90.87
	K-means	71.67	84.18	74.62	71.70	68.83
	LRR	99.12	99.63	99.33	99.17	99.09
	SSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	SC-SRGF	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	USENC	98.46	98.94	98.67	98.02	97.84
10	HFLRSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	NCut	82.82	89.03	85.10	80.15	78.36
	K-means	57.20	63.51	61.38	44.61	38.83
	LRR	98.90	99.54	99.17	98.95	98.86
	SSC	95.83	95.31	95.83	91.69	91.00
	SC-SRGF	89.93	89.72	89.95	83.10	81.70
	USENC	88.79	89.46	89.63	82.50	80.99
11	HFLRSC	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	NCut	71.45	76.92	72.77	63.55	60.35
	K-means	52.20	55.65	55.89	36.86	30.48
	LRR	89.37	89.41	90.38	80.02	78.26
	SSC	87.83	85.71	87.83	76.05	73.98
	SC-SRGF	83.50	84.49	84.73	75.85	73.83
	USENC	81.46	81.76	82.79	70.22	67.62
12	HFLRSC	**99.17**	**98.88**	**99.17**	**98.24**	**98.10**
	NCut	58.32	63.05	60.10	45.04	40.29
	K-means	46.56	48.47	50.24	29.39	22.36
	LRR	83.68	83.71	83.92	69.75	67.04
	SSC	85.58	82.37	85.58	70.23	67.62
	SC-SRGF	75.73	73.50	75.98	59.88	56.50
	USENC	46.96	48.12	48.54	27.86	21.27

It should be noted that when extracting the HFTC, the experimental images needed to be filtered with a Butterworth filter, which has two hyperparameters: filter order and cut-off frequency. Therefore, we designed a group of face images through a Butterworth filter to intuitively express the impact of these two production parameters on the images in Figure 4. Although we fully grasped the physical meanings of these two hyperparameters, different datasets often need different combinations of two parameters, which still need the guidance of human prior knowledge. Therefore, it would lead to great progress to propose a filter to automatically find parameters.

Aside from that, the trade-off parameters of HFLRSC play roles in balancing the low rank part of the model. Because of the efficiency of HFLRSC, the values of these parameters are often set within a large range, and the best parameter value with the best result is selected by cycling all the parameter values.

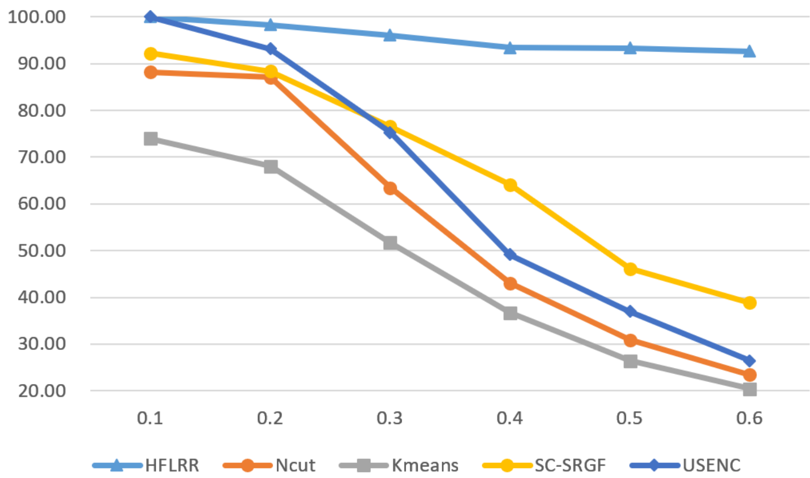
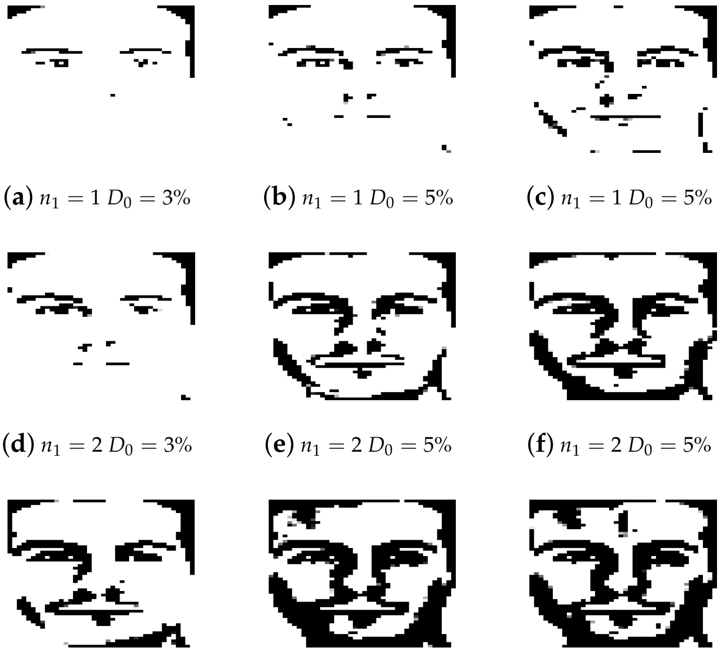


Figure 3. The clustering accuracy varies with the level of Gaussian noise.



(**a**) $n_1 = 1$ $D_0 = 3\%$ (**b**) $n_1 = 1$ $D_0 = 5\%$ (**c**) $n_1 = 1$ $D_0 = 5\%$

(**d**) $n_1 = 2$ $D_0 = 3\%$ (**e**) $n_1 = 2$ $D_0 = 5\%$ (**f**) $n_1 = 2$ $D_0 = 5\%$

(**g**) $n_1 = 5$ $D_0 = 3\%$ (**h**) $n_1 = 5$ $D_0 = 4\%$ (**i**) $n_1 = 5$ $D_0 = 5\%$

Figure 4. Figures from (**a**–**i**) present the change caused by the filter's parameters, where n_1 is the order of the filter and D_0 denotes the cut-off frequency by the percentage of the image width. As can be seen, the features of the high-frequency texture components were related to the cut-off frequency, while the thickness of the texture depended on the order of the filter.

5. Conclusions

In this work, we discuss how to effectively process a low-quality image of an integrated circuit. At this stage, many hardware Trojans and defect detection methods have been invented. However, most of these methods require high-quality data images. Limited by cost, high-quality images are often difficult to obtain. In addition, due to mechanical errors or processes, interference is inevitable, which may cause data corruption. Therefore, this study proposes an innovative framework called HFLRSC that attempts to develop fusion subspace clustering using low-rank constraints and high-frequency texture components for processing low-quality images in order to offer a data foundation for the subsequent verification tasks. The first step of this algorithm is using a two-dimensional Fourier transform and Butterworth filter to extract high-frequency texture components, which would be fused into the subspace clustering. The presented method achieved high clustering precision and a stable property to the level of data corruption, according to experimental results for the simulated datasets.

Author Contributions: Conceptualization, G.T. and Z.L.; Methodology, G.T.; Software, G.T.; Validation, G.T.; Formal analysis, Z.L.; Investigation, G.T.; Data curation, G.T.; Writing—original draft, G.T.; Writing—review & editing, Y.L. and J.L.; Visualization, G.T.; Supervision, Z.L., Y.C., Q.L., B.P. and J.L.; Project administration, Y.L. and J.L.; Funding acquisition, Y.C., Q.L. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key-Area Research & Development Program of Guangdong Province under Grants 2020B010173001 and 2020B010170002, the Open Foundation of the Guangdong Provincial Key Laboratory of Electronic Information Products Reliability Technology, and the Neuroeconomics Laboratory of Guangzhou Huashang College (2021WSYS002).

Institutional Review Board Statement: This study does not require ethical approval, and the authors chose to exclude this statement.

Informed Consent Statement: This study does not involve humans, and the authors choose to exclude this statement.

Data Availability Statement: Not applicable.

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

References

1. Hao, Q.; Zhang, Z.; Xu, D.; Wang, J.; Liu, J.; Zhang, J.; Ma, J.; Wang, X. A hardware security-monitoring architecture based on data integrity and control flow integrity for embedded systems. *Appl. Sci.* **2022**, *12*, 7750. [CrossRef]
2. Lu, H.; Capecci, D.E.; Ghosh, P.; Forte, D.; Woodard, D.L. Computer vision for hardware security. In *Emerging Topics in Hardware Security*; Springer: Cham, Switzerland, 2021; pp. 493–525.
3. Wang, X.; Zhang, Z.; Xu, Y.; Zhang, L.; Yan, R.; Chen, X. Real-time minor defect recognition of pseudo-terahertz images via the improved yolo network. In Proceedings of the 2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD), Nanjing, China, 21–23 October 2021; pp. 1–6.
4. Zhou, B.; Aksoylar, A.; Vigil, K.; Adato, R.; Tan, J.; Goldberg, B.; Ünlü, M.S.; Joshi, A. Hardware trojan detection using backside optical imaging. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *40*, 24–37. [CrossRef]
5. Rosso, D. *Annual Semiconductor Sales Increase 21.6 Percent, Top 400 Billion for First Time*; Semiconductor Industry Association: Washington, DC, USA, 2018; Volume 1.
6. Xue, M.; Bian, R.; Wang, J.; Liu, W. Building an accurate hardware trojan detection technique from inaccurate simulation models and unlabelled ics. *IET Comput. Digit. Tech.* **2019**, *13*, 348–359. [CrossRef]
7. Ye, M.; Zhang, D.; Yap, Y.K. Recent advances in electronic and optoelectronic devices based on two-dimensional transition metal dichalcogenides. *Electronics* **2017**, *6*, 43. [CrossRef]
8. Kaintura, A.; Dhaene, T.; Spina, D. Review of polynomial chaos-based methods for uncertainty quantification in modern integrated circuits. *Electronics* **2018**, *7*, 30. [CrossRef]
9. Jin, Y.; Makris, Y. Hardware trojan detection using path delay fingerprint. In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, USA, 9 June 2008; pp. 51–57.
10. Gayatri, R.; Gayatri, Y.; Mitra, C.; Mekala, S.; Priyatharishini, M. System level hardware trojan detection using side-channel power analysis and machine learning. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 650–654.
11. Liakos, K.G.; Georgakilas, G.K.; Plessas, F.C. Hardware trojan classification at gate-level netlists based on area and power machine learning analysis. In Proceedings of the 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Tampa, FL, USA, 7–9 July 2021; pp. 412–417.
12. Nasr, A.A.; Abdulmageed, M.Z. Automatic feature selection of hardware layout: A step toward robust hardware trojan detection. *J. Electron. Test.* **2016**, *32*, 357–367. [CrossRef]
13. Kulkarni, A.; Pino, Y.; Mohsenin, T. Svm-based real-time hardware trojan detection for many-core platform. In Proceedings of the 2016 17th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 15–16 March 2016; pp. 362–367.
14. Hasegawa, K.; Oya, M.; Yanagisawa, M.; Togawa, N. Hardware trojans classification for gate-level netlists based on machine learning. In Proceedings of the 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), Sant Feliu de Guixols, Spain, 4–6 July 2016; pp. 203–206.

15. Kulkarni, A.; Pino, Y.; Mohsenin, T. Adaptive real-time trojan detection framework through machine learning. In Proceedings of the 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), McLean, VA, USA, 3–5 May 2016; pp. 120–123.
16. Vishnupriya, R.; Devi, M.N. Hardware trojan detection using deep learning-deep stacked auto encoder. In Proceedings of the International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications, Hyderabad, India, 28–29 March 2021; Springer: Singapore, 2021; pp. 345–353.
17. Yasaei, R.; Yu, S.-Y.; Faruque, M.A.A. Gnn4tj: Graph neural networks for hardware trojan detection at register transfer level. In Proceedings of the 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 1–5 February 2021; pp. 1504–1509.
18. Muralidhar, N.; Zubair, A.; Weidler, N.; Gerdes, R.; Ramakrishnan, N. Contrastive graph convolutional networks for hardware trojan detection in third party ip cores. In Proceedings of the 2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Tysons Corner, VA, USA, 12–15 December 2021.
19. Sharma, R.; Sharma, G.; Pattanaik, M. A few shot learning based approach for hardware trojan detection using deep siamese cnn. In Proceedings of the 2021 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID), Guwahati, India, 20–24 February 2021; pp. 163–168.
20. Hasegawa, K.; Yamashita, K.; Hidano, S.; Fukushima, K.; Hashimoto, K.; Togawa, N. Node-wise hardware trojan detection based on graph learning. *arXiv* **2021**, arXiv:2112.02213.
21. Wei, L.; Wang, X.; Wu, A.; Zhou, R.; Zhu, C. Robust subspace segmentation by self-representation constrained low-rank representation. *Neural Process. Lett.* **2018**, *48*, 1671–1691. [CrossRef]
22. Göçeri, E. Impact of deep learning and smartphone technologies in dermatology: Automated diagnosis. In Proceedings of the 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA), Paris, France, 9–12 November 2020; pp. 1–6.
23. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 171–184. [CrossRef] [PubMed]
24. Bertsekas, D.P. On penalty and multiplier methods for constrained minimization. *SIAM J. Control. Optim.* **1976**, *14*, 216–235. [CrossRef]
25. Elhamifar, E.; Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2765–2781. [CrossRef] [PubMed]
26. Cai, X.; Huang, D.; Wang, C.-D.; Kwoh, C.-K. Spectral clustering by subspace randomization and graph fusion for high-dimensional data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Cham, Switzerland, 2020; pp. 330–342.
27. Huang, D.; Wang, C.-D.; Wu, J.-S.; Lai, J.-H.; Kwoh, C.-K. Ultra-scalable spectral clustering and ensemble clustering. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 1212–1226. [CrossRef]