REFICS: A Step Towards Linking Vision with Hardware Assurance

Ronald Wilson, Hangwei Lu, Mengdi Zhu, Domenic Forte and Damon L. Woodard
Florida Institute for Cybersecurity Research (FICS), University of Florida
Gainesville, FL 32601, USA
ronaldwilson@ufl.edu

Abstract

Hardware assurance is a key process in ensuring the integrity, security and functionality of a hardware device. Its heavy reliance on images, especially on Scanning Electron Microscopy images, makes it an excellent candidate for the vision community. The goal of this paper is to provide a pathway for inter-community collaboration by introducing the existing challenges for hardware assurance on integrated circuits in the context of computer vision and support further development using a large-scale dataset with 800,000 images. A detailed benchmark of existing vision approaches in hardware assurance on the dataset is also included for quantitative insights into the problem.

1. Introduction

The awe-inspiring capabilities of contemporary electronic devices stems from their extensive use of Integrated Circuits (IC) and Printed Circuits Boards (PCB). The utility of ICs is as diverse as the avenues in which its used - ranging from power efficient Internet-of-Things devices to high-performance computing clusters. However, to meet increasing demands for performance, functionality and energy efficiency, the complexity of these devices were scaled up significantly. This is especially true for ICs built using nanoscale structures. Their inherent complexity, generated by integrating several billion transistors in a tiny space, provides ample opportunities for adversaries to hide malicious modifications within the IC. These hidden modifications, called hardware Trojans, may impact the expected lifetime of the device or make it vulnerable to adversarial attacks i.e. compromise the integrity of the device (see Figure 1(a)).

With the real world consequences of using compromised devices ranging from exposure of sensitive data to failure of mission critical systems, ensuring the integrity of these devices becomes a priority. *Hardware assurance* refers to the process of verifying the design of hardware devices to ensure that there are no malicious modification present in the device. In computing, this process is akin to ensuring that a computer system is free from spyware or bloatware that may cause premature system failure or compromises the system to outside threats. In its infancy, the assurance measures for ICs were performed manually on optical images in a time and resource intensive manner [44]. However, the advent of modern ICs, with nanoscale features not resolvable under optical imaging modalities, this approach was rendered obsolete and ineffective. This prompted the transition from optical imaging to electron microscopy techniques, such as the Scanning Electron Microscopy (SEM), and the adoption of automated image analysis techniques into the hardware assurance process.

The ICs are manufactured by sandwiching multiple layers with unique properties into a complex three dimensional structure (see Figure 1(b)). For assuring trust in the device, structures in every layer of the IC needs to be verified. With the fragile nature of the IC, complicated techniques, like Reverse Engineering (RE), is required to access every layer of the IC and image them. A detailed workflow for RE can be found in a recent survey [5]. In simple terms, the RE workflow destructively removes the topmost exposed layer of the IC and acquires images of the exposed region in an iterative loop till all the layers are processed. As expected, RE is a complicated process and introduces several undesirable artefacts to the acquired images affecting the efficacy of the hardware assurance process. Some of these challenges, akin to uneven lighting and shot noise interference, were effectively handled by the computer vision community. However, there are more challenges and open research questions that needs to be addressed.

There are several instances, like with X-ray and Magnetic Resonance Imaging (MRI), where the computer vision community has assisted in development of critical algorithmic infrastructure to address issues in the medical community. In this paper our goal is to introduce a large-scale SEM image dataset, called REFICS[1], to provide a pathway for the computer vision community to assist in addressing challenges in the field of hardware assurance. Although

[1]Published under Creative Commons Attribution (CC-BY-4.0). Hosted on Trust-hub. Link: https://trust-hub.org

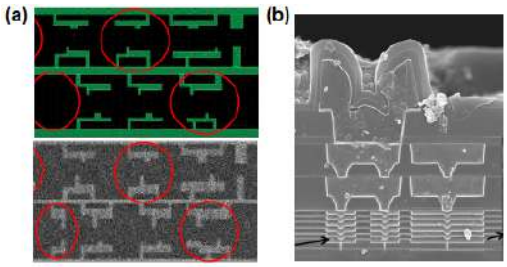

Figure 1. (a) An example of a hardware Trojan [47]. The original layout is on the top and the SEM image of the corresponding location on the IC is on the bottom. (b) Cross-section of an IC captured using SEM imaging indicating multiple layers in its makeup [32].

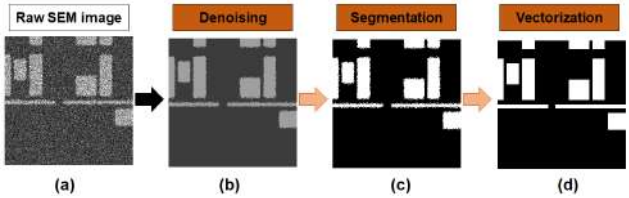

Figure 2. The image processing pipeline for hardware assurance

there are several datasets available for hardware assurance on PCBs [33, 38, 34, 22, 6], REFICS is the only dataset to introduce both the SEM imaging modality and hardware assurance problems for ICs into the vision community. The rest of the paper is organized as follows: Section 2 elaborates on the challenges in hardware assurance for ICs and presents them in the context of computer vision for easier understanding and abridging the distance between both the communities. Section 3 expands on the dataset generation process and benchmarks existing image analysis/computer vision approaches in the hardware assurance community. Section 3 further includes the insights that can be leveraged for developing effective vision algorithms. The work is concluded in Section 4 along with information on planned future expansions for the dataset.

2. Relevance to Computer Vision

Hardware assurance is a relatively unknown domain for the computer vision community. The importance of the process in preserving privacy and security of ICs was described earlier. However, these facts do not highlight the difficulty and challenges associated with the process or the long term viability of the domain for computer vision studies. The goal in this section is to emphasize the challenges associated with the process and highlight the short-term and long-term research prospects for the community. The problems can be classified based on their scope and complexity involved in addressing them. They are low-level vision, high-level vision, data assessment and data manipulation problems.

The **low-level vision** problems is based on the limitations of the existing image processing pipeline in hardware assurance. The pipeline is shown in Figure 2. These are common problems in computer vision dealt with a wide variety of approaches. However, there are certain factors that make this problem hard. An exemplary situation is shown in Figure 3. The scale of features in IC SEM images may only span a few pixels. With noise and other factors affecting the quality of the image, the image processing pipeline may detect non-existent hardware Trojans and/or have ambiguous detection. With the images collected from a single IC aggregating to several hundred thousand images, resolving each ambiguity would lead to several hundred hours of manual verification [26]. The obvious solution to the problem is to limit the imaging modality to high quality images -both in terms of feature sizes (magnification) and noise characteristics. But this leads to the inflation of the image acquisition time frame by orders of magnitudes [47], thereby, making the process infeasible for everyday application. This requires the image processing pipelines to provide accurate results (as close to the ground truth as possible) without increasing image quality. The existing image processing approaches in hardware assurance as well as their efficacy on the dataset is given in Section 3.2. A major bottleneck in the hardware assurance process is considered to be the image processing pipeline. Hence, time-efficient, reliable and robust algorithms needs to be developed and integrated into the existing pipeline.

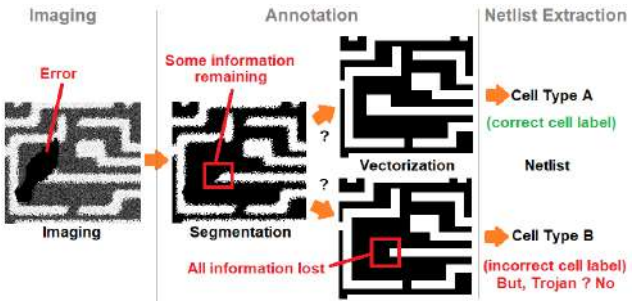

Figure 3. Exemplary case depicting accumulation of errors in each successive step of the hardware assurance process for improperly processed/acquired data.

The **high-level vision** problems builds on the information provided by the image processing pipelines and interprets the data on a contextual level. Assuming the integrity of the acquired data, the high-level vision problems focuses on two tasks: *assembling the fragmented data into a contextually-valid human-interpretable form* and *learning from the data for improving the low-level vision pipeline*. The first task has its origins in the field-of-view of the imaging modality. The scale of the features ensures that multiple images of the IC needs to be taken to have feature resolvability. This, in turn, requires the images to be merged together to form a larger view of the IC. With existing approaches in the hardware assurance community relying on

Problem Type	Hardware Assurance task	Computer Vision counterparts
Low-level vision	Denoising, Segmentation, Vectorization	Segmentation, Efficient training and inferencing
High-level vision	Cross-node generalization Stitching Alignment	Transfer learning Representation learning Detection and localization in 2D and 3D
Data manipulation	Synthetic image generation, Obfuscation	Neural generative models, Adversarial learning
Data assessment	Quality Metric/Anomaly Detection Missing/Damaged features	Explainable AI Image reconstruction

Table 1. Summary representation of hardware assurance problems and their counterparts in computer vision

simple approaches like cross-correlation to merge images, several artefacts, such as stitching errors, are brought into existence. This problem also extends into 3D when the individual layers of the IC need to be aligned. With the core of hardware assurance problems requiring the matching between the acquired IC SEM layout and a known Trojan-free layout, feature representations immune to stitching errors and misalignment needs to be developed and studied.

The latter task is based on the existence of various IC vendors and node technologies. Every IC vendor has their own type of unique features for each layers. These features are taken from a private library and constrained by a set of unknown design rules. This issue requires that the approaches designed for a specific IC to be generalizable to other ICs as well. In Section 3.3, a brief study of the generalization of supervised machine learning methods is provided to demonstrate the magnitude of the problem. Further, obtaining ground-truth labels for each pixel in the image for every new IC vendor and node technology encountered is an infeasible and arduous task. Hence, the number of samples to be manually labelled needs to be reduced or eliminated by the use of transfer/zero or few-shot learning approaches.

The semiconductor industry relies heavily on confidentiality to protect their design rules and libraries from public exposure. With the consequence of such exposure being compromised devices and stolen intellectual property, their concern and safeguard practices are well warranted. However, this severely limits the possibility of data sharing/collection and possibilities for further advancements of the research field. This is the prime motivating reason behind using a synthetic workflow to generate images for our dataset. Hence, the primary goal of the **Data manipulation** problem is to develop necessary algorithmic infrastructure, such as privacy preserving transforms [40, 15, 1], to obfuscate/transform the design data so that the original data can be hidden but the characteristics of the layout can be learned and the knowledge can be utilized in resolving issues with the image processing pipeline. This can also be utilized in generating more diverse synthetic images.

The **Data assessment** problem reflects on the lack of supporting infrastructure in hardware assurance for evaluating the efficacy of the image processing pipeline. Currently, image quality metrics such as intersection-over-union (IoU), structural similarity index measure (SSIM), mean squared error (MSE) and peak signal-to-noise ratio (PSNR) are being used to evaluate the efficacy of the process. These borrowed metrics from computer vision do not fully incorporate the contextual quality of the images into account. For instance, an over-segmented component in the image could mean an open-circuit in the recovered layout. Similarly, an under-segmented component could mean a short-circuit in the design. Moreover, there are situations where these metrics do not agree on the efficacy of an algorithm and provide conflicting results. This aspect is discussed in detail in Section 3.3.

Finally, most of the existing knowledge in hardware assurance, especially based on images, come from experience reported in case studies. There are several instances where the study was interrupted by unknown issues resulting in anomalous data and required human intervention to resolve. With the image acquisition process being repetitive and time consuming, the process is automated. However, these anomalous data does not get handled at the imaging phase and creeps into the pipeline causing more errors to accumulate at later stages. Error resolution by a subject matter expert at later stages is labor intensive and extremely time consuming. Some case studies report error resolution time frames as more than the time period required for image acquisition [26]. Hence, detecting anomalous data in the image acquisition phase is of paramount importance. Similarly, with the majority of the layout design generated from repeating patterns, the anomalous/corrupted data can be reconstructed using learned features from other images of the IC with techniques such as image inpainting.

A brief summary of the challenges along with their computer vision counterparts are shown in Table 1. Addressing these challenges will require the development of key algorithmic infrastructure and the widespread adoption and integration of algorithms from the computer vision community. Although it is a significant undertaking, improving the efficacy and robustness of the hardware assurance process is critical in ensuring a safe and secure cyberspace.

3. The Dataset

The SEM produces a single channel grayscale image with the intensity of each pixel representing the properties

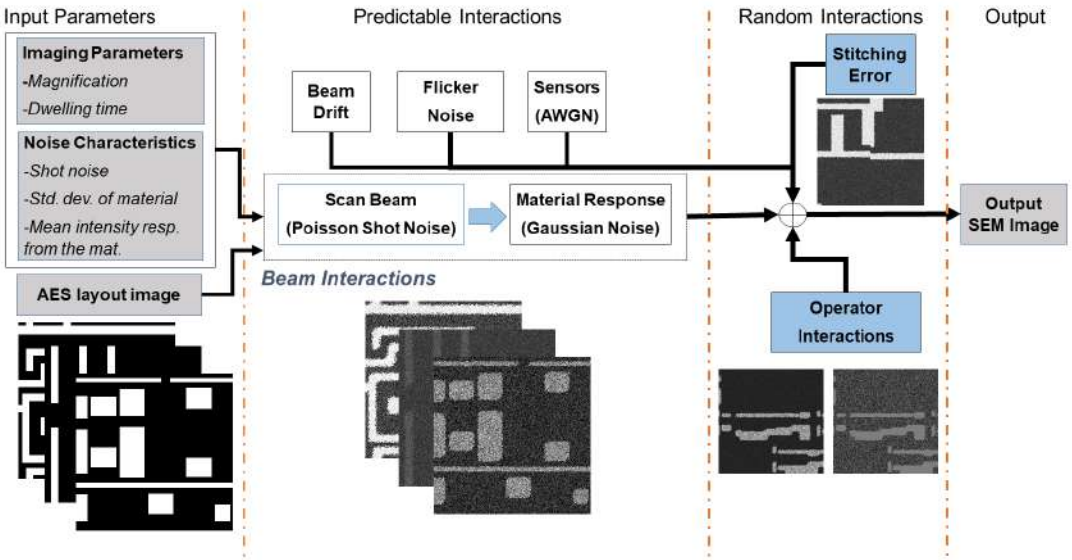

Figure 4. Workflow for generating a synthetic SEM image for the REFICS dataset.

of the source material. The SEM directs a scanning beam towards the sample under study and the sample, in response, releases electrons towards the SEM detector. The pixel intensity value is scaled in response to the number of electrons received in response to the scanning beam. The image formation process, similar to a regular camera, is relatively simple and can be simulated. Existing studies in electron microscopy and IC fault analysis also support this statement [11, 12, 7]. The electron microscopy community introduced an SEM image simulator called ARTIMAGEN, an initiative supported by the National Institute for Standards and Technology (NIST) [11, 12], capable of generating images with varying influences of drift, vibration, thermal expansion, and noise profiles (Gaussian/Poisson). However, the selection of materials, noise profiles, and shape contours are limited and not suitable for IC hardware assurance. Methods in IC fault analysis also use simulated images using a similar process. For instance, in benchmarking Line Edge/Width Roughness (LER/LWR) algorithms [7]. A recent deep learning (DL) approach generated synthetic SEM images based on layout data for mask optimization and virtual meteorology [39]. With precedent established in substituting real SEM images with a synthetic proxy, an SEM image generator, with the limitations of the existing works addressed, can be a viable source of diverse on-demand data for hardware assurance.

3.1. Generating the Dataset

The initial requirement for generating a synthetic SEM image is to have the correct context. In this case, the context is the layout-level design file synthesized using standard cell libraries. Approximately 10,000 standard cells from two standard libraries, 32/28nm and 90nm, were used to generate the four cardinal layers of an IC, namely, doping, polysilicon, contacts and the metal layer [19, 18]. The difference in each layer, visually, is in its contrast (material characteristics) and shapes (library-dependent). The layout files were split into 250×250 patches and fed into the image synthesis workflow, as described in Figure 4, along with the image synthesis parameters.

There are two sets of input parameters for image synthesis. The first set corresponds to the imaging settings in the SEM: the Field-of-View/Magnification (1×, 2×, 3× and 4× the original standard cell dimensions) and the dwelling time per pixel (3.2 μsec/pixel and 10 μsec/pixel). The second set of parameters corresponds to the noise characteristics: the shot noise parameter for the scanning beam along with the expected mean pixel intensity and standard deviation of the material under study. The shot noise distorts the scanning beam intensity at 2%, 5%, 10% and 20%. Every μsec spent on a pixel is equivalent to 1000 samples acquired from the simulation. So, a single pixel acquired at 10 μsec/pixel setting would simulate 10,000 samples from a Monte-Carlo simulation using the beam interaction model. A Poisson-Gaussian model was used to model the beam interactions in the workflow. The mean pixel intensity response for each material was acquired from real SEM images of that layer. This concludes the simulation of the image formation process.

The noise profile for hardware assurance using SEM comes with a few additional predictable and random noise sources. The predictable noise sources, such as beam drift and sensor noise, were adopted from earlier works. The beam drift was simulated using a 5×5 kernel where the beam drift probabilities from the center of the kernel to the periphery was determined by a Gaussian distribution. Sen-

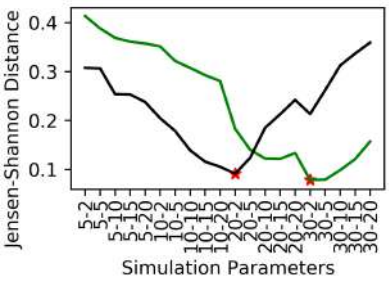

Figure 5. Plot indicating the similarity between the real and synthetic SEM image. Simulation parameter format: Standard deviation of the material-Shot noise parameter. The black and green trends indicate 10 μsec/pixel and 3.2 μsec/pixel dwelling times with the real parameters estimated at 22-2 and 38-2 respectively.

sor noise was modelled as Additive White Gaussian Noise. The random interactions are added to the image after the SEM image is generated. Currently, the synthesis workflow applies two random interaction to the SEM image: stitching errors and operator interactions. Stitching errors are applied randomly on either the vertical, horizontal or both axes at the same time. The operator interactions are limited to contrast adjustments in the image. This modification is applied by introducing random variation to the material's mean pixel intensity response. For instance, the mean pixel intensity response for the doped region is 160. Operator interaction randomly samples a Gaussian distribution with the mean set at 160 and a standard deviation of 15 to change the mean pixel intensity response from the materials. This modification can increase or decrease contrast in the image. Finally, the corners in the original layout are converted to simple curves in the SEM images to capture the variation introduced by the mask generation process for etching the IC layout onto the wafer during manufacturing [31]. The radius of the curve is randomly sampled from one to five. The chosen magnification parameter also scales the radius. The final SEM image is generated at the end of this stage. This SEM image synthesis workflow constitutes the closest substitute to a real SEM image obtained from an IC through complex resource-intensive processes like RE.

The claim for the closest substitute to a real SEM image can be verified through extensive experimentation. The experiment protocol applied for verification consists of acquiring a large number of SEM images of an IC at a fixed set of imaging parameters and comparing them against synthetic SEM images generated using the same parameters through Monte Carlo simulations of the beam interactions and the other predictable interactions. If the model is valid, then the statistical similarity of the synthetic and the real SEM image should be highest at the same imaging parameters. The imaging parameters used in the simulation includes dwelling time per pixel, shot noise and the standard deviation of the material. The simulation generated 64,000 pixels for every possible combination of the parameters listed earlier. The comparison between the real and synthetic SEM image was done by using the pixel intensity histogram of the respective images and comparing their similarity using Jensen-Shannon divergence. A divergence value of zero indicates that both the histograms are the same. Similarly, texture-based similarity was assessed in the Fourier domain by taking the cosine distance between the magnitude spectrum of the images. Both experiments produced identical results, shown in Figure 5, suggesting that the real and synthetic SEM images are very similar. For additional information on model robustness and validation, especially in the context of real-world noise sources, refer supplementary material Sections 1 and 2.

Finally, the REFICS dataset was generated. It currently consists of 800,000 synthetic SEM images with 100,000 SEM image for each layer per node technology. Every SEM image has a corresponding segmented ground truth (GT) and a layout-level mask. In addition to assisting in resolving the challenges enumerated in Table 1, the dataset can also be used for bench-marking novel image processing algorithms and developing hardware assurance specific neural network architectures. For approaches that involve complex DL strategies or a directed purpose like handling stitching errors, a tool[2] is made available for generating more SEM image samples. The tool assists in generating more SEM images and can be modified to generate SEM images with a particular error or set of errors depending on the user's intended application.

3.2. Existing Literature in Hardware Assurance

A detailed understanding of existing approaches in hardware assurance is necessary to identify shortcomings and contribute effectively. To facilitate this, a summary of existing works is provided below.

Denoising: There are several approaches in literature for processing noisy SEM images. They include spatial filtering approaches, including Gaussian, median, curvature, anisotropic diffusion, wavelet, adaptive wiener filter, and hysteresis smoothing [35, 3, 46, 36]. Simple high-frequency filtering and DL-based denoising approaches have also been used on SEM images [17]. ML-based denoising approaches, such as image inpainting, super-resolution and dictionary-based sparse reconstruction, have also been explored for SEM images [28, 45, 7, 30]. SEM image quality is assessed using PSNR and SSIM [41, 43, 42, 24].

Segmentation: Segmentation algorithms for SEM images can be supervised, unsupervised or interactive. Supervised segmentation approaches based on Support Vector Machines (SVM) and Convolutional Neural Network (CNN) were explored [21, 9]. The unsupervised approaches are

[2]Also made available with the dataset

based on generalizable features that can be found in the same IC or across ICs. For instance, the technique developed by [8, 9, 10] relies on the fact that polysilicon structures and metal layer traces can be generated by simple Manhattan geometry contours. Interactive approaches, such as [13], require the operator to guide the segmentation. K-means and Fuzzy C-means are some simpler unsupervised segmentation approaches [8]. LASRE is another unsupervised technique that relies on using frequency-based texture signatures for different materials to segment out IC structure across multiple layers [49]. Simple image processing techniques, such as Otsu's binarization, have also been explored for segmenting SEM images [14, 48, 29, 27, 47]. The segmentation protocol can be conducted in two ways: the raw SEM image is segmented directly (Figure 2(a)→(c)) and the raw SEM image is denoised before it is segmented (Figure 2(a)→(b)→(c)). Denoising SEM images before segmentation typically yields better results. Segmentation accuracy is measured in MSE, SSIM, F-measure and IoU.

Under-segmented and over-segmented shapes in the SEM image typically translates to a short-circuit and open-circuit in the design. Since this quality is not measured by existing metrics, a 4-connected components analysis was used to find the ratio of short-circuited and open-circuited components to all the components present in the segmented image. These are referred to as CC-US and CC-OS (refer supplementary material Section 3). A value of zero indicates perfect segmentation in terms of electrical connectivity. In unison with the existing measures/metrics, these two additional metrics provides a better understanding of the true quality of the segmented results.

Vectorization: This process is used to recover the design files as close to the original layout as possible by converting the segmented image into a bunch of polygons. It also serves in suppressing edge noise between materials and compressing the amount of data in the image. Simple edge following algorithms have been explored in this context [4].

Deep Learning: Although the use of DL for image processing and computer vision is mainstream, its adoption into SEM images, especially for hardware assurance, is in its infancy. These existing approaches can be classed into *image-to-image translation* and *blind denoising*. The first class of approaches are used to convert an image from one representation into another [23]. In REFICS, the raw SEM image and its corresponding GT can be considered as two representations of one image. Under this assumption, the pix2pix network was used for SEM image quality enhancement [37], and CycleGAN was used to transform SEM images into corner-deformed GT for further image comparison [39]. The latter class of approaches are used to remove real-world image noises from photographs. DnCNN is the most used architecture in SEM related applications. It has been leveraged for EWR/LWR estimation on images with unknown levels of Gaussian-Poisson mixture noise [7, 17]. However, DnCNN is often criticized for easily over-fitting to a specific noise model. Networks architectures, proposed recently, have addressed this issue on generalizability by reusing features with long and short skip connections [20, 2, 25]. However, these advancements were not validated in hardware assurance. The CBDNet was suggested to be effective in preserving sharp edges -a highly desirable characteristic for hardware assurance applications. Most DL models use an end-to-end architectures by-passing the individual steps of the image processing pipeline currently used in the field. The input to the networks are the raw SEM images and the outputs are expected to be the original segmented GT images (Figure 2(a)→(d)). The outputs are evaluated using the segmentation metrics.

3.3. Results and Discussion

In this section, the performance of the image processing and ML methods are evaluated using the metrics discussed earlier. The results are presented in Table 2. The key characteristic of a good algorithm for use in hardware assurance is in its ability to score high on the chosen metrics and maintain stable scores across different layers and node technologies. This characteristic, called cross-generalizability, is critical for supervised approaches that maintain heuristics on the layout design data from the labelled GT. Consequently, the cross-generalizability of DL methods between layers and nodes is also investigated and discussed in detail.

Denoising: To obtain the ground truth denoised image in the REFICS dataset, apply the mean intensity response of the materials in the image to the segmented ground truth. The key observation from the presented data is that the denoising performance reduces in the order: Polysilicon $>$ Doping $>$ Metal layer. In most cases, the metal layer shows reduction in image quality after denoising. This can be attributed to the fact that the contrast in the metal layer is much higher than those of other layers. The contrast is the lowest in the polysilicon layer and, hence, benefits the most from denoising. Anisotropic diffusion filters performs the best. This filter smooths the image while preserving the edges. With the hardware design layout being produced by straight edges, the performance metrics behind this filter can be intuitively understood. The Gaussian filter and Median filter performed relatively well. The ML-based denoising approaches performed poorly as compared to regular methods. Note that, Gaussian filter and BM3D performed consistently across all layers and node technologies.

Segmentation: The results were obtained by comparing the original and segmented SEM images (Figure 2 (a)→(c)). Denoising was not performed on the raw SEM image before segmentation. The key observation from the table is that the results are similar to the observations from the denoising experiments. A simple image binarization method

Algorithm	Metal Layer		Doping Layer		Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node	32nm node	90nm node
Denoising Algorithms (Improvement in % for PSNR (↑) / SSIM (↑) over raw SEM image)						
Gaussian fil	8.11 / 22.53	9.46 / 22.24	15.50 / 30.58	15.93 / 32.52	15.84 / 27.99	17.27 / 36.75
Aniso. diff. fil.	1.60 / 45.67	6.37 / 44.79	**26.08** / 72.73	28.64 / 79.16	**29.44 / 62.28**	37.82 / 91.62
Curvature fil.	-28.56 / 29.55	-23.27 / 29.93	-14.65 / 50.02	1.36 / 67.16	18.41 / 52.47	32.46 / 84.95
Median fil.	0.30 / **48.73**	7.01 / 46.83	25.83 / **75.55**	**30.02 / 82.41**	26.82 / 59.91	**42.06 / 94.74**
Adap. Weiner	-27.20 / -29.05	-21.09 / -12.42	-9.73 / 12.05	3.84 / 33.63	9.69 / 30.03	22.81 / 83.45
BM3D	10.20 / 17.99	12.86 / 18.14	11.21 / 22.56	13.54 / 21.22	7.5 / 14.25	12.93 / 19.66
K-SVD	**12.52** / 37.95	**15.32 / 64.73**	23.07 / 49.35	16.00 / 64.27	22.47 / -9.65	23.73 / 65.88
Segmentation Algorithms (SSIM (↑) / IoU (↑) / CC-US (↓) / CC-OS (↓))						
Otsu's thresh.	0.77 / **0.88** / **0.11** / 0.91	**0.79 / 0.91 / 0.13** / 0.69	0.55 / 0.73 / 0.38 / 0.77	0.27 / 0.49 / 0.29 / 0.61	0.40 / 0.52 / 0.64 / 0.69	0.12 / 0.29 / 0.80 / 0.53
Fuzzy C-means	0.75 / 0.86 / **0.11** / 0.91	0.78 / 0.90 / 0.14 / 0.68	0.53 / 0.72 / 0.38 / 0.77	0.27 / 0.49 / 0.30 / 0.60	0.39 / 0.51 / 0.65 / 0.68	0.11 / 0.28 / 0.83 / 0.52
K-means	0.77 / **0.88** / **0.11** / 0.91	**0.79 / 0.91 / 0.13** / 0.69	0.55 / 0.73 / 0.38 / 0.77	0.27 / 0.49 / 0.29 / 0.60	0.40 / 0.52 / 0.64 / 0.69	0.12 / 0.29 / 0.81 / 0.53
HAS	**0.85** / 0.78 / 0.41 / 0.70	0.76 / 0.82 / 0.36 / 0.17	**0.85 / 0.81** / 0.43 / 0.78	0.81 / 0.80 / 0.17 / 0.18	**0.67** / 0.52 / 0.52 / 0.76	**0.56 / 0.46 / 0.33** / 0.60
LASRE	0.75 / 0.70 / 0.15 / **0.14**	0.72 / 0.76 / 0.28 / 0.22	0.78 / 0.79 / **0.09 / 0.20**	0.72 / 0.73 / 0.12 / 0.28	0.46 / **0.58 / 0.30 / 0.38**	0.22 / 0.44 / 0.39 / **0.42**
SVM-10	0.76 / 0.73 / 0.26 / 0.78	0.67 / 0.79 / 0.20 / **0.15**	0.74 / 0.78 / 0.27 / 0.86	**0.85 / 0.85 / 0.05 / 0.11**	0.34 / 0.44 / 0.60 / 0.78	0.32 / 0.37 / 0.61 / 0.47
Deep Learning Algorithms (SSIM (↑) / IoU (↑) / CC-US (↓) / CC-OS (↓))						
DnCNN	0.94 / 0.90 / **0.00** / 0.03	0.92 / 0.92 / **0.00** / 0.10	0.96 / 0.95 / **0.00** / 0.02	0.94 / 0.91 / **0.00** / 0.07	0.83 / 0.67 / **0.00** / 0.48	0.88 / 0.63 / 0.02 / 0.17
CBDNet	**0.96 / 0.94 / 0.00** / 0.03	**0.96 / 0.95** / 0.01 / 0.04	**0.98 / 0.97 / 0.00 / 0.00**	**0.98 / 0.96 / 0.00 / 0.01**	**0.95 / 0.93 / 0.00 / 0.02**	**0.96 / 0.87 / 0.00 / 0.03**
Pix2pix	0.88 / 0.85 / **0.00 / 0.01**	0.72 / 0.70 / 0.01 / 0.04	0.86 / 0.84 / **0.00** / 0.03	0.76 / 0.71 / **0.00 / 0.01**	0.90 / 0.85 / **0.00** / 0.07	0.67 / 0.74 / 0.01 / 0.04
CycleGAN	0.93 / 0.89 / **0.00 / 0.01**	0.78 / 0.74 / 0.01 / **0.02**	0.96 / 0.90 / **0.00** / 0.03	0.95 / 0.72 / **0.00 / 0.01**	0.90 / 0.87 / **0.00** / 0.07	0.91 / 0.62 / 0.02 / 0.04

Table 2. Benchmark of image processing algorithms used on SEM images in hardware assurance. The negative values reported for denoising algorithms indicate degradation in image quality after denoising. For segmentation algorithms, apart from SVM, all other methods are unsupervised. K-means, Fuzzy C-means and HAS use a 5×5 kernel and SVM uses a 10×10 kernel. The highest improvement in metrics for each layer and node technology is highlighted in bold.

Networks	Metal Layer		Doping Layer		Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node	32nm node	90nm node
DnCNN	**0.93 / 0.90 / 0.00** / 0.04	0.91 / 0.91 / **0.00** / 0.09	0.92 / 0.91 / 0.02 / 0.06	**0.96 / 0.93 / 0.00 / 0.02**	0.80 / 0.75 / 0.04 / 0.06	0.83 / 0.41 / **0.00** / 0.53
CBDNet	0.90 / 0.87 / 0.01 / 0.05	**0.94 / 0.94 / 0.00** / 0.05	**0.95 / 0.94** / 0.01 / **0.02**	0.95 / 0.92 / **0.00** / 0.06	**0.83 / 0.80 / 0.01** / 0.03	**0.86 / 0.57** / 0.01 / **0.03**
Pix2pix	0.73 / 0.70 / 0.04 / **0.03**	0.75 / 0.70 / **0.00** / 0.05	0.86 / 0.84 / **0.00** / 0.04	0.71 / 0.65 / **0.00** / 0.06	0.66 / 0.61 / 0.03 / 0.25	0.66 / 0.41 / 0.03 / 0.30
CycleGAN	0.88 / 0.83 / 0.02 / **0.03**	0.79 / 0.72 / **0.00 / 0.02**	0.90 / 0.82 / **0.00** / 0.05	0.91 / 0.70 / 0.01 / 0.09	**0.83** / 0.76 / **0.01 / 0.02**	0.59 / 0.41 / 0.05 / 0.28

Table 3. Cross-node generalizability results. The listed node technology represents the test set with the network trained on the other node. The results are represented as SSIM / IoU / CC-US / CC-OS scores. The highest improvement in metrics is highlighted in bold.

Trained on Metal Layer				
Networks	Tested on Doping Layer		Tested on Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node
DnCNN	0.91 / 0.82 / **0.00 / 0.01**	0.94 / 0.89 / **0.00** / 0.03	0.66 / 0.07 / **0.00** / 0.22	0.76 / 0.01 / **0.00** / 0.02
CBDNet	0.87 / 0.72/ **0.00** / 0.15	**0.95 / 0.90 / 0.00** / 0.05	0.66 / 0.06 / 0.00 / 0.09	0.76 / 0.01 / 0.00 / 0.04
Pix2pix	0.85 / 0.83 / **0.00** / 0.09	0.68 / 0.63 / **0.00** / 0.02	0.51 / 0.53 / 0.12 / 0.18	0.26 / 0.22 / 0.20 / 0.29
CycleGAN	**0.92 / 0.89 / 0.00 / 0.01**	0.78 / 0.73 / **0.00 / 0.01**	**0.75 / 0.69 / 0.09 / 0.17**	0.55 / 0.37 / 0.09 / 0.18
Trained on Doping Layer				
Networks	Tested on Metal Layer		Tested on Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node
DnCNN	**0.91 / 0.89 / 0.01 / 0.04**	0.85 / 0.87 / 0.03 / 0.32	0.78 / 0.47 / 0.00 / 0.38	0.76 / 0.05 / 0.00 / 0.09
CBDNet	0.88 / 0.82 / **0.01** / 0.13	**0.91 / 0.91 / 0.02** / 0.08	**0.76 / 0.60 / 0.00 / 0.17**	0.78 / 0.17 / 0.00 / 0.14
Pix2pix	0.65 / 0.63 / 0.06 / 0.14	0.69 / 0.67 / 0.04 / **0.06**	0.34 / 0.40 / 0.20 / 0.26	0.31 / 0.23 / 0.19 / 0.25
CycleGAN	0.81 / 0.70 / 0.04 / 0.24	0.79 / 0.57 / **0.02** / 0.20	0.43 / 0.32 / 0.31 / 0.15	0.77 / 0.23 / 0.06 / 0.17
Trained on Polysilicon Layer				
Networks	Tested on Metal Layer		Tested on Doping Layer	
	32nm node	90nm node	32nm node	90nm node
DnCNN	**0.85 / 0.82** / 0.08 / **0.09**	0.83 / 0.79 / 0.08 / 0.12	**0.89 / 0.87** / 0.08 / **0.03**	0.88 / 0.80 / 0.03 / 0.03
CBDNet	0.82 / 0.67 / **0.01** / 0.49	0.90 / **0.87** / 0.04 / 0.12	0.80 / 0.63 / **0.00** / 0.24	**0.95 / 0.87** / 0.04 / 0.12
Pix2pix	0.75 / 0.55 / 0.05 / 0.52	0.64 / 0.55 / 0.05 / 0.24	0.68 / 0.46 / 0.01 / 0.60	0.72 / 0.58 / **0.00** / 0.09
CycleGAN	0.76 / 0.59 / 0.03 / 0.44	**0.91** / 0.72 / **0.02 / 0.04**	0.74 / 0.65 / 0.02 / 0.36	0.93 / 0.69 / **0.00 / 0.02**

Table 4. Cross-layer generalizability results. The results are represented as SSIM / IoU / CC-US / CC-OS scores. The highest improvement in metrics is highlighted in bold.

like Otsu's thresholding has performance equivalent to that of ML approaches in the metal layer. Otsu's thresholding along with K-means and Fuzzy C-means also demonstrate stable performance across all layers and node technologies. HAS conserves more of the shape information in the segmented image while losing connectivity information [48]. LASRE, on the other hand, preserves more connectivity information over shape information. The interesting observation in the table is that a supervised segmentation approach based on SVM performs similar to unsupervised methods despite having access to labelled ground truth data. The SVM was trained on 90,000 images and tested on 10,000 images of a single layer and node technology. Since one GT in the dataset may correspond to a couple of noisy raw SEM images, this splitting is chosen to guarantee the test set is independent from the trained models. The parameter for the SVM classifier was obtained from an earlier work [9]. As suggested by the author, cascading different classi-

fiers or using a committee of classifiers will possibly yield better results than using an individual classifier.

Deep Learning: The end-to-end baseline performance results are obtained by training and testing on the same subset of one node technology and one layer in a 9:1 split ratio as done in the case of the SVM. The training parameters for each network are adjusted for the best performance. A threshold value of 127 was used to binarize the output images. The key observation from the baseline experiments is that most deep neural networks perform consistently on different layers. CBDNet performs the best. This may be attributed to multiple losses and the Unet architecture, which was designed for semantic segmentation. The two image translation networks perform consistently, however, they also show lower IoU scores. This may be due to missing pixels on small features or the stitching errors carrying over to reconstructed images. DnCNN shows decreased performance on the polysilicon layer, which has lower contrast comparing to other layers. Since DnCNN solely follows the residual learning schematic, low contrast images having noise pixels similar to the pixels representing clean images could cause denoising difficulties for DnCNN. Inaccurate reconstruction cases observed for these networks are of two main types. The first type is caused by low contrast, where the patterns in the noisy image can barely be seen. The second type is limited to stitching error. End-to-end models does not seem to account for stitching errors accurately. Exemplary cases are presented in Section 5 of the supplementary material. It is also observed that these four networks outperform conventional methods. However, the results reported in Table 3 and Table 4 suggests that the networks do not generalize well across different technology nodes and IC layers. Exemplary labelled data for each target layer and node technology will be required to boost the efficacy up to necessary levels required for hardware assurance.

Overall Insights: The benchmarks serve as a quantitative reminder over the type of algorithms that can be chosen to resolve any directed image processing task in hardware assurance. However, there are some key observations from the presented results that can be leveraged for the development of better algorithms and smoother integration of data-driven paradigms into image processing.

The metrics commonly used in evaluating image quality and segmentation accuracy are not stable. There are several instances where the highest score in two metrics evaluating segmentation accuracy, in terms of shape for instance, goes to two different algorithms. Similarly, the methods that can achieve a high score on shape similarity measurement may not perform the same in terms of electrical connectivity. For example, in the metal layer, similar SSIM and IoU values are observed across nodes, while CC exhibits significant differences. To truly evaluate image quality, multiple metrics maybe necessary or a novel metric, specifically designed for hardware assurance tasks, has to be developed.

A very interesting observation from the result is that most approaches show a lack of stability across node technologies and IC layers. Realizing the fact that the images are generated using the same beam interaction models with varying layouts, it is counter-intuitive for the algorithms to have variations in performance. The effect is compounded for the polysilicon layer. Even supervised methods with access to large quantities of high-quality labelled data show this trend. This suggest that the approaches are not able to detect the edges in the original layout effectively. This effect can be clearly seen in the performance metrics reported, especially for the polysilicon layer with the lowest contrast among all three layers.

Another noteworthy observation is in Table 4. DL networks trained on the metal layer, a high-contrast image with relatively simple geometry, performed well on other layers especially in terms of separation between structures, i.e. the CC metric. However, when trained on the other two layers with structures having complex geometry, they performed better in terms of conserving the shape of the structures, i.e. SSIM and IoU metrics. Although this doesn't affect the state-of-the-art performances provided by the DL models significantly, this does underline the fact that model architectures that are capable of resolving the edges between different materials under low contrast need to be developed. Off-the-shelf complex neural architectures may not be enough for hardware assurance applications. Supporting evidence can be found in a critical work that suggests that neural networks, especially those that work on images, are influenced more by the texture of the image than by the edges themselves [16]. Hence, more directed research is necessary for the development of effective neural network models. This observation also provides credence to the efficacy of template-based segmentation approaches [10].

4. Conclusion

In this paper, a brief overview on the role of computer vision in hardware assurance process for ICs was provided along with the associated challenges. These challenges, both low-level vision and high-level contextual problems, could not be resolved using existing algorithmic infrastructure, including DL approaches, due to its inherent complexity. Consequently, a dataset is introduced along with necessary benchmarks, to indicate areas requiring significant improvement, for further research and to establish a pathway for both short-term and long-term inter-community collaboration. Furthermore, the dataset will be diversified in the future using the provided software tool to include more IC layouts and noise profiles, and, a standalone real SEM image sub-dataset, collected by executing the RE workflow on specially designed ICs, for in-depth study into the problem and promoting constraint-free community research.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 439–450, 2000.

[2] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3155–3164, 2019.

[3] N. Arazm, A. Sahab, and M. F. Kazemi. Noise reduction of sem images using adaptive wiener filter. In *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, pages 50–55, 2017.

[4] Simon Blythe, Beatrice Fraboni, Sanjay Lall, Haroon Ahmed, and Ugo de Riu. Layout reconstruction of complex silicon chips. *IEEE journal of solid-state circuits*, 28(2):138–145, 1993.

[5] Ulbert J Botero, Ronald Wilson, Hangwei Lu, Mir Tanjidur Rahman, Mukhil A Mallaiyan, Fatemeh Ganji, Navid Asadizanjani, Mark M Tehranipoor, Damon L Woodard, and Domenic Forte. Hardware trust and assurance through reverse engineering: A survey and outlook from image analysis and machine learning perspectives. *arXiv preprint arXiv:2002.04210*, 2020.

[6] Adam Byerly, Tatiana Kalganova, and Anthony J Grichnik. On the importance of capturing a sufficient diversity of perspective for the classification of micro-pcbs. *arXiv preprint arXiv:2101.11164*, 2021.

[7] Narendra Chaudhary, Serap A Savari, and Sai S Yeddulapalli. Line roughness estimation and poisson denoising in scanning electron microscope images using deep learning. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 18(2):024001, 2019.

[8] Deruo Cheng, Yiqiong Shi, Bah-Hwee Gwee, Kar-Ann Toh, and Tong Lin. A hierarchical multiclassifier system for automated analysis of delayered ic images. *IEEE Intelligent Systems*, 34(2):36–43, 2018.

[9] Deruo Cheng, Yiqiong Shi, Tong Lin, Bah-Hwee Gwee, and Kar-Ann Toh. Hybrid k-means clustering and support vector machine method for via and metal line detections in delayered ic images. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(12):1849–1853, 2018.

[10] Deruo Cheng, Yiqiong Shi, Tong Lin, Bah-Hwee Gwee, and Kar-Ann Toh. Global template projection and matching method for training-free analysis of delayered ic images. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2019.

[11] Petr Cizmar, András E Vladár, Bin Ming, Michael T Postek, National Institute of Standards, and Technology. Simulated sem images for resolution measurement. *Scanning*, 30(5):381–391, 2008.

[12] Petr Cizmar, András E Vladár, and Michael T Postek. Optimization of accurate sem imaging by use of artificial images. In *Scanning Microscopy 2009*, volume 7378, page 737815. International Society for Optics and Photonics, 2009.

[13] Piali Das, Olga Veksler, Vyacheslav Zavadsky, and Yuri Boykov. Semiautomatic segmentation with compact shape prior. *Image and Vision Computing*, 27(1-2):206–219, 2009.

[14] Alexander Doudkin, Alexander Inyutin, and Maksim Vatkin. Objects identification on the color layout images of the integrated circuit layers. In *2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 610–614. IEEE, 2005.

[15] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *International symposium on privacy enhancing technologies symposium*, pages 235–253. Springer, 2009.

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[17] E Giannatou, G Papavieros, V Constantoudis, H Papageorgiou, and E Gogolides. Deep learning denoising of sem images towards noise-reduced ler measurements. *Microelectronic Engineering*, 216:111051, 2019.

[18] Richard Goldman, Karen Bartleson, Troy Wood, Kevin Kranen, C Cao, Vazgen Melikyan, and Gayane Markosyan. Synopsys' open educational design kit: capabilities, deployment and future. In *2009 IEEE International Conference on Microelectronic Systems Education*, pages 20–24. IEEE, 2009.

[19] R Goldman, K Bartleson, T Wood, K Kranen, V Melikyan, and E Babayan. 32/28nm educational design kit: Capabilities, deployment and future. In *2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, pages 284–288. IEEE, 2013.

[20] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[21] Xuenong Hong, Deruo Cheng, Yiqiong Shi, Tong Lin, and Bah Hwee Gwee. Deep learning for automatic ic image analysis. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pages 1–5. IEEE, 2018.

[22] Weibo Huang, Peng Wei, Manhua Zhang, and Hong Liu. Hripcb: a challenging dataset for pcb defects detection and classification. *The Journal of Engineering*, 2020(13):303–309, 2020.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[24] Nidal S Kamel and KS Sim. Image signal-to-noise ratio and noise variance estimation using autoregressive model. *Scanning: The Journal of Scanning Microscopies*, 26(6):277–281, 2004.

[25] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3482–3492, 2020.

[26] Adam Kimura, Jon Scholl, James Schaffranek, Matthew Sutter, Andrew Elliott, Mike Strizich, and Glen David Via. A decomposition workflow for integrated circuit verification and validation. *Journal of Hardware and Systems Security*, pages 1–10, 2020.

[27] Dmitry Lagunovsky, Sergey Ablameyko, and M Kutas. Recognition of integrated circuit images in reverse engineering. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, volume 2, pages 1640–1642. IEEE, 1998.

[28] A Lazar and Petru S Fodor. Sparsity based noise removal from low dose scanning electron microscopy images. In *Computational Imaging XIII*, volume 9401, page 940105. International Society for Optics and Photonics, 2015.

[29] Jang Hee Lee and Suk In Yoo. An effective image segmentation technique for the sem image. In *2008 IEEE international conference on industrial technology*, pages 1–5. IEEE, 2008.

[30] Myungjun Lee, Jason Cantone, Ji Xu, Lei Sun, and Ryounghan Kim. Improving sem image quality using pixel super resolution technique. In *Metrology, Inspection, and Process Control for Microlithography XXVIII*, volume 9050, page 90500U. International Society for Optics and Photonics, 2014.

[31] Bernhard Lippmann, Niklas Unverricht, Aayush Singla, Matthias Ludwig, Michael Werner, Peter Egger, Anja Duebotzky, Helmut Graeb, Horst Gieser, Martin Rasche, et al. Verification of physical designs using an integrated reverse engineering flow for nanoscale technologies. *Integration*, 71:11–29, 2020.

[32] Bernhard Lippmann, Michael Werner, Niklas Unverricht, Aayush Singla, Peter Egger, Anja Dübotzky, Horst Gieser, Martin Rasche, Oliver Kellermann, and Helmut Graeb. Integrated flow for reverse engineering of nanoscale technologies. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 82–89. ACM, 2019.

[33] Hangwei Lu, Dhwani Mehta, Olivia P Paradis, Navid Asadizanjani, Mark Tehranipoor, and Damon L Woodard. Fics-pcb: A multi-modal image dataset for automated printed circuit board visual inspection. *IACR Cryptol. ePrint Arch.*, 2020:366, 2020.

[34] Gayathri Mahalingam, Kevin Marshall Gay, and Karl Ricanek. Pcb-metal: A pcb image dataset for advanced computer vision machine learning component analysis. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–5. IEEE, 2019.

[35] G Masalskis et al. Reverse engineering of cmos integrated circuits. *Elektronika ir elektrotechnika*, 88(8):25–28, 2008.

[36] Mohadeseh Mazhari and Reza PR Hasanzadeh. Suppression of noise in sem images using weighted local hysteresis smoothing filter. *Scanning*, 38(6):634–643, 2016.

[37] Yoshihiro Midoh and Koji Nakamae. Image quality enhancement of a cd-sem image using conditional generative adversarial networks. In *Metrology, Inspection, and Process Control for Microlithography XXXIII*, volume 10959, page 109590B. International Society for Optics and Photonics, 2019.

[38] Christopher Pramerdorfer and Martin Kampel. A dataset for computer-vision-based pcb analysis. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 378–381. IEEE, 2015.

[39] Hao-Chiang Shao, Chao-Yi Peng, Jun-Rei Wu, Chia-Wen Lin, Shao-Yun Fang, Pin-Yen Tsai, and Yan-Hsiu Liu. From ic layout to die photo: A cnn-based data-driven approach. *arXiv preprint arXiv:2002.04967*, 2020.

[40] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[41] KS Sim, ME Nia, and CP Tso. Image noise cross-correlation for signal-to-noise ratio estimation in scanning electron microscope images. *Scanning*, 33(2):82–93, 2011.

[42] KS Sim, ME Nia, and Chih Ping Tso. Noise variance estimation using image noise cross-correlation model on sem images. *Scanning*, 35(3):205–212, 2013.

[43] JTL Thong, KS Sim, and JCH Phang. Single-image signal-to-noise ratio estimation. *Scanning*, 23(5):328–336, 2001.

[44] Randy Torrance and Dick James. The state-of-the-art in ic reverse engineering. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 363–381. Springer, 2009.

[45] Patrick Trampert, Faysal Bourghorbel, Pavel Potocek, Maurice Peemen, Christian Schlinkmann, Tim Dahmen, and Philipp Slusallek. How should a fixed budget of dwell time be spent in scanning electron microscopy to optimize image quality? *Ultramicroscopy*, 191:11–17, 2018.

[46] Bruno Machado Trindade, Eranga Ukwatta, Mike Spence, and Chris Pawlowicz. Segmentation of integrated circuit layouts from scan electron microscopy images. In *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, pages 1–4. IEEE, 2018.

[47] Nidish Vashistha, M Tanjidur Rahman, Haoting Shen, Damon L Woodard, Navid Asadizanjani, and Mark Tehranipoor. Detecting hardware trojans inserted by untrusted foundry using physical inspection and advanced image processing. *Journal of Hardware and Systems Security*, 2(4):333–344, 2018.

[48] Ronald Wilson, Navid Asadizanjani, Domenic Forte, and Damon L Woodard. Histogram-based auto segmentation: A novel approach to segmenting integrated circuit structures from sem images. *arXiv preprint arXiv:2004.13874*, 2020.

[49] Ronald Wilson, Domenic Forte, Navid Asadizanjani, and Damon Woodard. Lasre: A novel approach to large area accelerated segmentation for reverse engineering on sem images. In *ISTFA 2020: 46th International Symposium for Testing and Failure Analysis*, page To be published. ASM International, 2020.