AUTOMATIC ERROR DETECTION IN INTEGRATED CIRCUITS IMAGE SEGMENTATION: A DATA-DRIVEN APPROACH

Zhikang Zhang [1] *Bruno Machado Trindade* [2] *Michael Green* [2] *Zifan Yu* [1]
Christopher Pawlowicz [2] *Fengbo Ren*[1]

[1] Arizona State University, Tempe, AZ, USA [2] TechInsights Inc., Ottawa, ON, Canada

ABSTRACT

Due to the complicated nanoscale structures of current integrated circuits(IC) builds and low error tolerance of IC image segmentation tasks, most existing automated IC image segmentation approaches require human experts for visual inspection to ensure correctness, which is one of the major bottlenecks in large-scale industrial applications. In this paper, we present the first data-driven automatic error detection approach that targets two types of IC segmentation errors: wire and via errors. On an IC image dataset collected from real industry, we demonstrate that, by adapting existing CNN-based approaches of image classification and image translation with additional pre-processing and post-processing techniques, we are able to achieve recall/precision of 0.92/0.93 in wire error detection and 0.96/0.90 in via error detection, respectively.

Index Terms— error detection, reverse engineering, image segmentation, image classification, image translation

1. INTRODUCTION

Ever-developing integrated circuit (IC) manufacturing technologies have led to the ever-increasing complexity of IC structures built at the nanoscale. Such high complexity can put normal users and IC designers at risk by providing opportunities for adversaries to hide malicious or IP-protected designs within the IC [1]. Reverse engineering is the best or even the only approach to date that can help address this problem [1]. In a nutshell, reverse engineering operates by repeatedly imaging and physically removing the topmost layer of an IC chip to reveal all inner 3D structures for analysis. Because of the extremely small size of IC structures, scanning electron microscopy(SEM) is often used as an imaging tool. Once SEM images are acquired, coarse IC designs can be retrieved by segmenting SEM images into different objects(we only target vias and wires in this work). SEM image quality can be affected by various unexpected factors, such as random contamination, improper exposure, and improper layer removal, which lead to unexpected segmentation errors in existing SEM image segmentation approaches [2, 3, 4, 5, 6, 7, 8, 9, 10, 11], as shown in Fig. 1. To correct such errors, human experts are required to visually inspect the segmentation results. However, in the real world, up to thousands of SEM images are generated from a single IC chip [1], which makes manual inspection highly challenging and laborious, leading to one of the major bottlenecks in large-scale industrial applications.

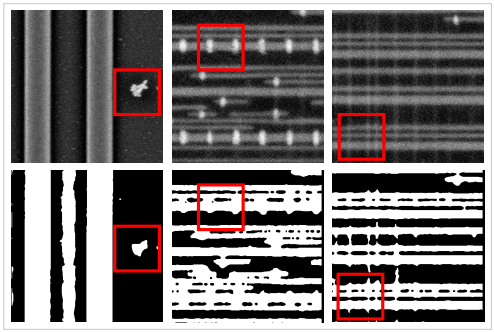

Fig. 1. Image samples with errors caused by unexpected factors. Row 1: SEM images. Row 2: Wire segmentation results. Column 1: Random Contamination. Column 2: Improper exposure. Column 3: Improper layer removal.

In this work, we target the problem of automatic via and wire error detection in segmented SEM images. We formulate the wire error detection and via error detection as image classification and image translation problems, respectively. By adapting existing related approaches with additional image pre-processing and post-processing techniques, we achieve high performance in both via and wire error detection problems on an SEM image dataset collect from the real industry.

Our contributions are summarized as follows:

1. This is the first work to identify and address the problem of automatic error detection in integrated circuit segmentation. Our approach achieves prominent performance on a real industrial dataset, which significantly unblocks a major bottleneck in reverse engineering and implies the potentially broad applicability of this approach.

2. Through the validation of this approach, we draw an insight that the error detection problem can be formulated

This work is supported by a research contract from TechInsights Inc.
Corresponding author: Zhikang.Zhang@asu.edu

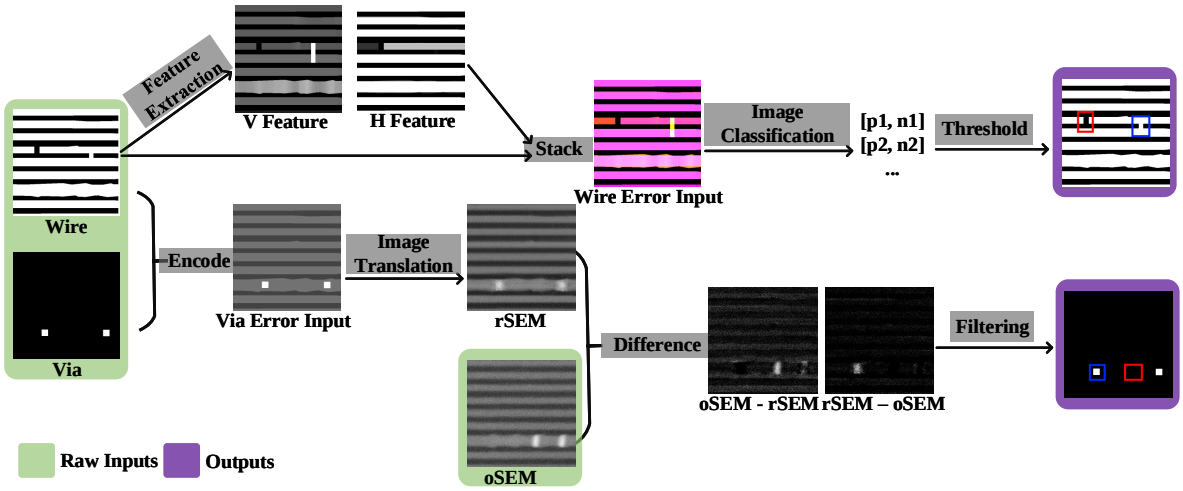

Fig. 2. Pipeline of the proposed approach. rSEM: reconstructed SEM image. oSEM: original SEM image. Red bounding boxes: "open" of wires or "miss" vias. Blue bounding boxes: "short" of wires or "extra" vias

as image classification and image translation problems separately to address. This insight reveals a pathway toward more sophisticated solutions to this problem.

3. This work demonstrates the effectiveness of CNN in error detection of segmented SEM images with the necessary image pre-processing and post-processing techniques, which inspires further research on designing CNN-based solutions to the problem.

2. METHODOLOGY

2.1. Datasets and Evaluation Metrics

Datasets. We collect 39 SEM images of various hardware: microprocessor, radio frequency(RF) transceiver, power management, flash memory, SoC, etc., with a collecting dwelling time of 0.2 μs/pixel. The average pixel and field sizes are 2.92nm and 22.96 μm, respectively. For brevity, this SEM image set is denoted S0. Each image in S0 is grayscale with a size of 8192x8192, which is too high to be fed into common CNNs. Thus, we choose to process images at the patch level (256x256) first and then merge the patch-level results to form full image results. There are two types of objects to be segmented: vias and wires. Vias are electrical connections between the copper layers in ICs, which are often imaged with the highest pixel intensity and are shown as small rounded squares. Wires are imaged with lower pixel intensity than vias but higher pixel intensity than backgrounds and are shown as long strips(Fig. 2.). We collect four wire segmentation sets(denoted as W0, W1, W2, and W3) and two via segmentation sets(denoted as V0 and V1) corresponding to S0. The W0 set is generated by [12] with manual corrections. W1, W2, and W3 sets are generated by [4] using three different settings. V0 and V1 sets are generated by [12] and [4], respectively.

Evaluation metrics. In the context of SEM image segmentation of IC, if and only if the segmentation differences that cause errors in the connectivity of IC are defined as segmentation errors. More precisely, for wires, only an "open" or a "short" are errors(Fig. 2.). For vias, only a "miss" or an "extra" are errors(Fig. 2.). To evaluate the error detection performance of our approach, we use the following metrics: for wire segmentation errors, given a wire error segmentation result EW and the corresponding ground truth segmentation result GW, we denote each patch of the image as either "correct" or "error" based on whether there is an electric-significant-differences(ESD) [12]in the patch. The wire error detection performance is quantified based on the number of detected error patches. For via segmentation errors, given a via error segmentation result EV and the corresponding ground truth segmentation result GV, we first extract all isolated regions from both EV and GV using [13]. Subsequently, each region from the EV that overlaps with a region from the GV is treated as a correctly segmented via. Other regions from either EV or GV that have no overlapping regions from GV or EV are treated as errors(corresponding to extra vias and miss vias, respectively). The via error detection performance is quantified by the number of detected via errors.

2.2. Wire Error Detection

We formulate the wire error detection problem as an image classification problem. A CNN-based binary image classifier slides over the pre-processed wire segmentation images to determine whether each image patch has errors.

Pre-processing. Because the input patch size(256x256) is

Fig. 3. For a given pixel(green), the horizontal and vertical extension values are defined as the length of blue and red lines, respectively.

significantly smaller than the original image size(8192x8192), we compose novel features to implicitly encode global information into inputs. Specifically, we first calculate the horizontal and vertical extension values of each pixel(as defined in Fig. 3), respectively, in a full-size wire segmentation image. Then the values are normalized into the $[0, 255]$ range to form an H feature and a V feature image, respectively. The original wire segmentation image is stacked with V and H features to form an RGB image as the input of the image classifier, as shown in Fig .2.

Training an image classifier. To compose a training set, we take the W0 set as the ground truth and randomly sample 256x256 image patches from W1, W2, and W3 as training samples. Each image patch is labeled as either positive or negative based on the presence of an error. We use ResNet [14] as the image classifier. The FC layer that generates the final output is modified to generate a binary output. The training strategy from [15] is used to train the network. Because the training set is highly unbalanced(positive samples are only 3%), we used a weighted cross-entropy loss whose weights for positive and negative samples are defined as $\frac{N}{P+N}$ and $\frac{P}{P+N}$ where P and N are the numbers of positive and negative samples, respectively.

Post-processing. In the inference stage, for each 256x256 patch, the direct output of the image classifier is a two-element vector denoted as $[p, n]$, where a positive detection is made when $p > n$. We apply a threshold over outputs of multiple overlapping patches to localize errors into smaller areas. The more positive detections are made, the more likely there is an error in the overlapping area. It should be noted that this post-processing step is only required to localize wire errors more precisely. To detect whether a given 256x256 patch contains errors, using the direct output of the image classifier is sufficient.

2.3. Via Error Detection

We formulate the via error detection as a one-to-one image translation problem. Pix2pix [16] is used as the image translation method. Given a pair of wire and via segmentation image W and V and corresponding original SEM image(oSEM), W and V are first encoded into one image and then fed into the pix2pix model to be translated into a reconstructed SEM image(rSEM). Finally, errors are detected by processing the differences between rSEM and oSEM, as shown in Fig. 2. Note that W is assumed to contain no error.

Pre-processing. We observed that the pixel values of the via, wire, and background are relatively static within the same SEM image but highly dynamic across different SEM images. Thus we dynamically encode via and wire segmentation images as input to pix2pix model with respect to each SEM image. We first estimate three representative pixel values for the via, wire, and background, denoted as v, w, and b. v, w, and b are the 90th, median, and 10th percentiles of the via, wire, and background pixel values, respectively. The pixels are isolated from oSEM according to via and wire segmentation images V and W.

Training a pix2pix model. In our experiments, W0 is used as the wire segmentation set. We compose a training set consisting of 39936 image patches(1024 patches per full-sized image) that were randomly sampled from both V0 and V1. The training strategy is adopted from [17]. Most of the training settings remain consistent with those of the original work. The differences are as follows:1. We set the number of input and output channels to one. 2. All the data augmentations are disabled. 3. The total number of training epochs are set to ten. In the last five epochs, the learning rate decays linearly.

Post-processing. We first calculate two differences between oSEM and rSEM: $D1 = rSEM - oSEM$ and $D2 = oSEM - rSEM$. The negative values of $D1$ and $D2$ are set to zero. $D1$ and $D2$ are used for detecting extra "vias" and "miss" vias, respectively. We transform the positive pixels in $D1$ and $D2$ into isolated regions using [13]. For each region, we first measure the following two values:1. Size of the bounding box covering region. 2. Average pixel values of nonzero pixels within the region. By setting proper upper and lower thresholds for these values, we can filter out candidate vias. Finally, the remaining regions in D1/D2 are marked as extra/miss vias when they overlap or do not overlap with any of the vias in oSEM.

3. EXPERIMENTS

Wire error detection. We compose two training sets of different sizes, denoted as "small" and "large. "small" set consists of 229,049 samples with 6883 positive samples. "large" set consists of 1,030,699 training samples with 27833 positive samples. The testing set consists of regularly sampled nonoverlapping image patches from W1, W2, and W3, including 99956 samples with 2437 positive samples. We trained six models corresponding to different input image encoding, network structures, training set sizes, and network structures. The experimental results are listed in Table 1.

The highest error detection performance is a recall/precision of 0.92/0.93 with ResNet-18 trained on the "large" set. The input image is encoded with the wire segmentation result, V and H features. These results indicate that our approach can effectively detect wire segmentation errors without human

Network	Input	Dataset	Strategy	Recall	Precision
ResNet18	W	small	From scratch	0.83	0.77
ResNet18	VH	small	From scratch	0.83	0.64
ResNet18	WVH	small	From scratch	0.84	0.79
ResNet34	WVH	small	From scratch	0.82	0.76
ResNet18	WVH	large	From scratch	0.92	0.93
ResNet18	WVH	small	Tuning	0.61	0.30

Table 1. The wire error detection performance of various trained models. W: encodes wire segmentation only. VH: encodes only the V and H features. WVH: encodes the wire segmentation with V and H features. From scratch: Training the network from scratch. Tuning: use a ResNet18 pre-trained on ImageNet and only tune the last FC layer in training.

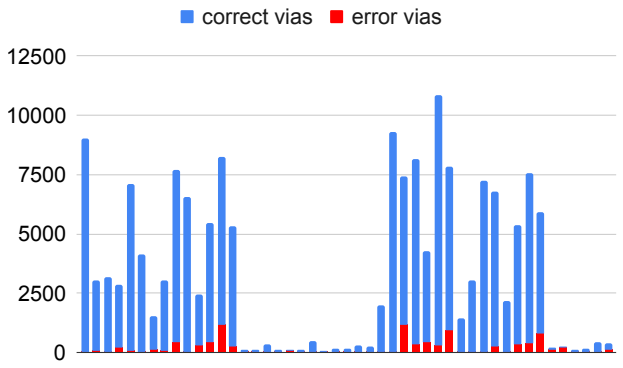

Fig. 4. The statistics of correct vias and error vias in the testing set.

intervention.

Compared with other trained models in Table 1, we conclude the followings:1. A deeper network structure does not necessarily lead to higher detection performance(ResNet18 vs. ResNet34, row 3 and row 4). This implies that the wire error detection task relies more on low-level local information in the images, which can be extracted by shallower neural networks. 2. Using wire segmentation images together with V and H features improves the detection performance, while using only V and H features is not sufficient(row 1, row 2, and row 3). 3. A larger training set effectively improves detection performance(row 3 and row 5). 4. Pre-training on ImageNet is not necessary(row 3 and row 6), which further implies that wire error detection relies more on low-level information. Thus, a network trained for high-level tasks does not perform well.

Via error detection. We alternately take V0 and V1 as the ground truth set and the error set for testing. Via images with no errors are excluded. The statistics of correct vias and error vias in the testing set are shown in Fig. 4. The number of correct vias and error vias varies dramatically for different images. The average ratio of via errors is 0.09. The experimental results of error detection are shown in Fig. 5.

Fig. 5. The via error detection results. Wrongly detected errors are either actual errors not detected(false negative results) or no-existing errors detected(false positive results)

We achieve an average recall/precision of 0.96/0.90 on the testing set, which shows that our approach can effectively detect via errors. We further examine the only outlier result in Fig. 5, which has 956 wrong detections and is significantly higher than the rest of the results. It turns out that most of the wrong detections in this image are false positives caused by the low contrast between the vias and wires. Because of the low contrast, the estimated v and w values tend to be very close, which results in filtering failure. Although such samples are rare in our dataset, this result reveals a limitation of our approach. We will address the problem of such cases in future work.

4. CONCLUSION

In this work, we propose an automatic error detection approach for segmented SEM images of IC circuits. We first formulate the wire and via error detection problems into image classification and image translation problems, respectively. By adapting existing approaches in these two domains with the necessary image pre-processing and post-processing techniques, we achieve an average recall/precision of 0.92/0.93 in wire error detection and 0.96/0.90 in via error detection, respectively. Our approach significantly unblocks one of the major bottlenecks for automatic SEM image segmentation approaches. The evaluation is conducted on a real industrial dataset, which implies the broad applicability of our approach to real-world applications.

5. REFERENCES

[1] Ronald Wilson, Hangwei Lu, Mengdi Zhu, Domenic Forte, and D. Woodard, "Refics: A step towards linking vision with hardware assurance," *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3461–3470, 2022.

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

[3] 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*, vol. 65, pp. 1849–1853, 2018.

[4] Zifan Yu, Bruno Machado Trindade, Michael Green, Zhikang Zhang, Pullela Sneha, Erfan Bank Tavakoli, Christopher Pawlowicz, and Fengbo Ren, "A data-driven approach for automated integrated circuit segmentation of scan electron microscopy images," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 2851–2855.

[5] 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*, vol. 34, pp. 36–43, 2019.

[6] 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," *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2019.

[7] Ronald Wilson, Domenic Forte, Navid Asadizanjani, and D. Woodard, "Lasre: A novel approach to large area accelerated segmentation for reverse engineering on sem images," 2020.

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

[9] Ronald Wilson, Navid Asadizanjani, Domenic Forte, and D. Woodard, "Histogram-based auto segmentation: A novel approach to segmenting integrated circuit structures from sem images," *ArXiv*, vol. abs/2004.13874, 2020.

[10] Janghee Lee and Suk In Yoo, "An effective image segmentation technique for the sem image," *2008 IEEE International Conference on Industrial Technology*, pp. 1–5, 2008.

[11] Dmitry Lagunovsky, Sergey V. Ablameyko, and M. Kutas, "Recognition of integrated circuit images in reverse engineering," *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, vol. 2, pp. 1640–1642 vol.2, 1998.

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

[13] Kesheng Wu, Ekow Otoo, and Arie Shoshani, "Optimizing connected component labeling algorithms," in *Medical Imaging 2005: Image Processing*. SPIE, 2005, vol. 5747, pp. 1965–1976.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] "Imagenet training in pytorch," `https://github.com/pytorch/examples/tree/main/imagenet`, Accessed: 2022-01-01.

[16] 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*, 2017, pp. 1125–1134.

[17] "Pix2pix model training in pytorch," `https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`, Accessed: 2022-01-01.