

Defect-aware data augmentation for improving machine learning-based contour extraction from SEM images

Hongming Zhang^{a,b}, Hawren Fang,^a Xuefeng Zeng,^a Le Hong,^a Yuyang Sun,^a Renyang Meng,^c Werner Gillijns,^c Wei Wu^{b,*}, and Yuansheng Ma^{a,*}

^aSiemens Digital Industries Software, Fremont, California, United States

^bUniversity of Southern California, Ming Hsieh Department of Electrical and Computer Engineering, Los Angeles, California, United States

^cIMEC, Leuven, Belgium

ABSTRACT. **Background:** Contour extraction from scanning electron microscopy (SEM) images is crucial for various fields because of its ability to provide precise surface morphology analysis. Accurate contour extraction methods are essential for quality control, failure analysis, defect detection, and the development of advanced devices.

Aim: However, existing machine learning-based SEM images contour extraction methods face the problem of difficulty in obtaining enough training data, particularly the lack of effective SEM image/contour pairs. This limitation often leads to defects in the extracted contours, such as open line-ends and discontinuous lines. Our aim is to address this problem.

Approach: We propose a defect-aware data augmentation pipeline that uses conditional generative adversarial networks and BicycleGAN to solve these issues. We generate defective contours by artificially introducing defects into ideal contours, which then serve as input for the generative model to generate synthetic SEM images with low contrast or weak signal in the corresponding parts.

Results: These generated SEM images, paired with the original ideal contours, are then used to retrain the contour extraction model, significantly improving the model's ability to handle previously challenging cases.

Conclusions: Our experimental results demonstrate that the proposed pipeline greatly enhances contour extraction performance, resolving previously open line-end defects. We highlight the potential of defect-aware data augmentation strategies and provide a practical framework for the future incorporation of more diverse defect types.

© 2025 Society of Photo-Optical Instrumentation Engineers (SPIE) [DOI: [10.1117/1.JMM.24.3.034005](https://doi.org/10.1117/1.JMM.24.3.034005)]

Keywords: scanning electron microscopy images; contour extraction; conditional generative adversarial network; BicycleGAN; data augmentation; image generative model

Paper 25056G received May 9, 2025; revised Jul. 18, 2025; accepted Jul. 24, 2025; published Aug. 21, 2025.

1 Introduction

Scanning electron microscopy (SEM) image contour extraction is an important task in semiconductor manufacturing, where precise morphology analysis is essential for quality control and

*Address all correspondence to Wei Wu, wu.w@usc.edu; Yuansheng Ma, yuansheng.ma@siemens.com

defect detection.^{1,2} Accurate contours of surface structures enable better identification of fabrication errors. Traditional image processing techniques such as mathematical morphology-based methods³ and Canny edge detection⁴ are often time-consuming and computationally intensive. In addition, when dealing with noisy, low-contrast SEM images, those methods often get inaccurate contour extraction results, limiting their applicability.

Recent advances in machine learning (ML) have led to the development of better-performing contour extraction models for SEM images. Especially recently, the emergence of large vision models (LVMs), such as the segment anything model⁵ and DINOv2,⁶ enables general-purpose contour extraction across diverse visual domains and has very strong generalization ability. These models demonstrate remarkable capabilities in contour detection and segmentation with simple fine-tuning for a specific task. However, applying LVMs directly to semiconductor SEM images contour extraction typically requires substantial computational power and large-scale datasets.

In this work, we want to develop a lightweight, computationally efficient model capable of ultra-fast, high-accuracy contour extraction specifically for our SEM image datasets. Our previous work^{7,8} successfully developed a convolutional neural network-based SEM contour extraction model, achieving efficient and highly accurate results on some datasets. However, there is a major limitation, that is, it is difficult to obtain high-quality ground-truth data pairs (SEM images and corresponding contours), leading to a lack of training data, resulting in errors such as open line-ends and broken contours in weak-signal or low-contrast regions. Figure 1 shows one sample failure case. These defects result in incomplete contour extraction, making further analysis and applications unreliable. To address these issues, we propose to develop a method that generates synthetic SEM images based on input contours, which will create paired datasets that can be used to train more robust and accurate contour extraction models.⁹

Traditional approaches for generating synthetic SEM images typically rely on Monte Carlo simulations,^{2,10} which model the interactions of electron beams with material surfaces on the physical level to generate realistic imaging results. These methods can realize high-quality results, but they are computationally intensive and extremely time-consuming, often requiring a long time to generate a single high-resolution image. This limits the application of Monte Carlo-based SEM image generation to large-scale dataset expansion.

In recent years, there has been a growing interest in machine learning based image generation techniques, which offer a more efficient and scalable way. Generative adversarial networks (GANs),¹¹ variational autoencoders (VAEs),¹² and diffusion models¹³ have demonstrated remarkable capabilities in synthesizing realistic images across diverse domains. In addition, there are conditional generative models,¹⁴ where the output image is generated based on a specific input, which are particularly relevant to our task. We want to generate SEM images conditioned on the input contours or designs.

Although some conditional diffusion models have achieved state-of-the-art performance in high-resolution image-to-image synthesis tasks, they typically require large training datasets and significant computational resources, making them less suitable for our application, where obtaining a lot of paired contour and SEM images for training is particularly challenging. Considering the trade-off between data availability, computational efficiency, and generation

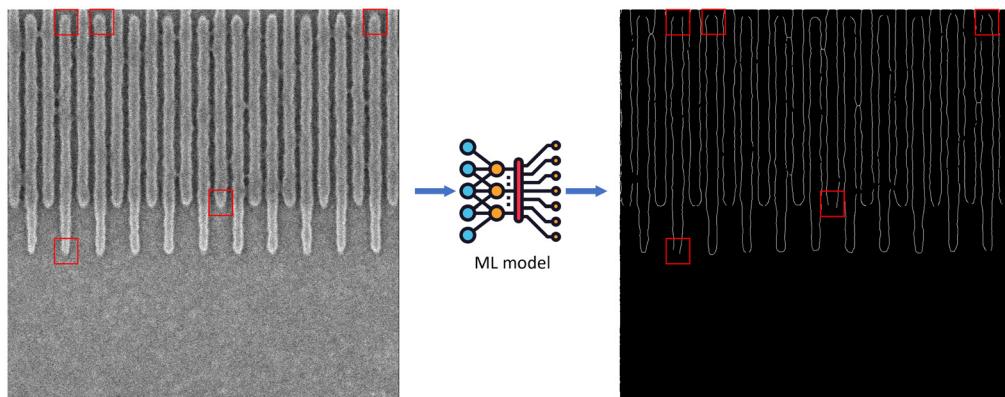


Fig. 1 Failure case of our previous⁸ contour extraction model with open line-ends.

quality, we adopt a conditional GAN (cGAN) based model¹⁵ and a BicycleGAN based model to generate corresponding SEM images from input contours. cGAN-based frameworks have also been applied successfully in medical image translation tasks,^{16–18} further demonstrating their feasibility in generating realistic structured outputs from conditional inputs. Meanwhile, BicycleGAN has shown strong potential in broader data generation problems,^{19,20} where it enables the synthesis of diverse and physically possible outputs from a single conditional input.

Based on this, we propose an efficient data augmentation pipeline for improving SEM contour extraction, which addresses the challenge of limited training data and significantly enhances the model's ability to handle the previously challenging cases.

2 Method

2.1 Overview of the Data Augmentation Pipeline

To enhance the performance of SEM contour extraction models in handling challenging cases, such as open line-ends and discontinuous contours, as illustrated in Fig. 1, we propose a data augmentation pipeline that synthesizes paired contour and SEM images.

The core idea is to simulate realistic imaging defects by artificially adding defects on the input contours and generating corresponding SEM images conditioned on these modified contours. These synthetic contour/SEM image pairs of data are then used to supplement the training dataset for contour extraction model training.

To achieve this, we first utilize our previous contour extraction model to extract contours from SEM images, which will provide contours with defects due to a lack of training data. These defective contours are then paired with their corresponding real SEM images to form an initial dataset for training a generative model. This enables the model to learn the mapping from imperfect contour inputs, such as contours with open line-ends or broken segments, to SEM images featuring low-contrast or weak-signal areas.

After the generative model is trained, we introduce controlled defects into ideal contour images to simulate previous failure cases. These defective contours are then used as inputs to the trained image generative model to generate synthetic SEM images, greatly expanding the training dataset.

The proposed data augmentation pipeline consists of the following stages:

1. Defective contour extraction: Extract defective contours from real SEM images using the existing contour extraction model.
2. Image generative model training: Train the image generative model to learn the mapping from defective contours to SEM images featuring low-contrast or weak-signal areas. In this work, we use a cGAN-based model and a BicycleGAN model.
3. Defects introduction: Controlled defects, such as open line-ends, are introduced into ideal contour images to simulate common defects observed in previous failure cases.
4. Synthetic SEM image generation: Generate synthetic SEM images based on the defective contour images using the trained image generative model.
5. Contour extraction model training: The generated synthetic SEM images, paired with the original ideal contours, are then used to train our contour extraction model.

2.2 Defective Contour Extraction

To improve the performance of contour extraction models on challenging cases, we first extract defective contours from real SEM images using our previously developed contour extraction model, as illustrated in Fig. 1. In this work, we mainly focus on addressing open line-end defects as a representative example to demonstrate the feasibility and effectiveness of the proposed data augmentation pipeline.

By pairing these extracted defective contours with their corresponding real SEM images, we establish an initial dataset that captures the mapping between imperfect contours and the resulting low-contrast, weak-signal regions in SEM images. This initial dataset will be used to train the image generative model in Sec. 2.4.

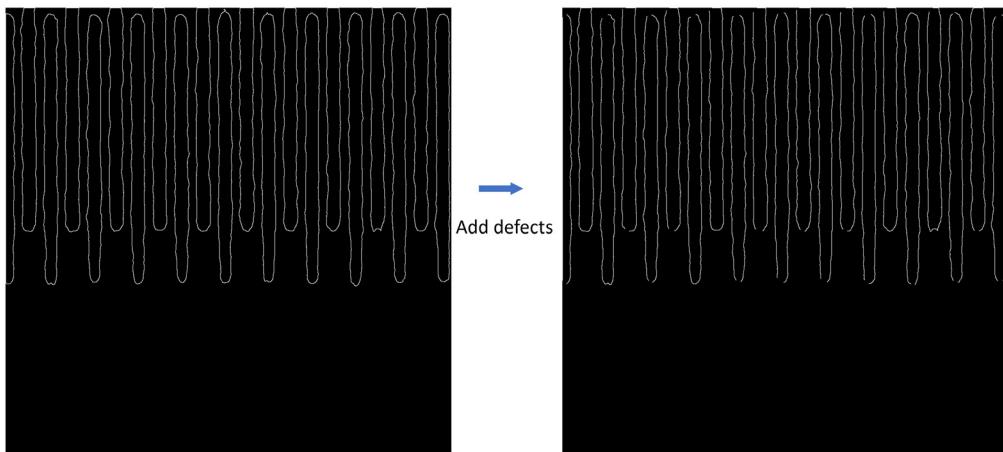


Fig. 2 Process of introducing defects into ideal contours.

2.3 Contour Processing: Defects Introduction into Ideal Contours

Here, for contour processing, we introduce controlled defects into ideal contour images. Although multiple types of defects can occur in SEM imaging, in this dataset, the main issue observed in the original contour extraction results obtained by our previous model is the presence of open line-end defects. Therefore, in this work, we focus on addressing the open line-end issues to improve the overall extraction performance.

To simulate this defect, we developed a targeted defect injection algorithm. This method identifies line-end regions in each ideal contour image and selectively removes segments to introduce defects. Specifically, 80% of the detected line-end points are randomly selected, and at each selected location, a square mask with a random size between 8 and 20 pixels is applied to remove the contour line. Through this focused defect simulation, the synthetic training data become more representative of the challenging cases that our pipeline aims to address. One example of introducing defects into an ideal contour image is shown in Fig. 2.

2.4 Synthetic SEM Image Generation

2.4.1 Conditional GAN(cGAN) model

We first employed a conditional GAN (cGAN) framework based on the pix2pix architecture¹⁵ to generate synthetic SEM images from input contour images. In this setup, the generator learns a mapping from the contour images to corresponding SEM images while the discriminator aims to distinguish between real and fake SEM images conditioned on the input contours.

In the adversarial training process, the generator tries to create SEM images as close to real ones as possible to fool the discriminator while the discriminator learns to distinguish between real and fake SEM images. By iteratively optimizing these two competing objectives, the generator gradually improves the realism of the synthesized SEM images.

The adversarial loss for the cGAN is defined as¹⁵

$$\mathcal{L}_{\text{cGAN}}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x)))] \quad (1)$$

where x is the input contour image, y is the corresponding real SEM image, G is the generator, and D is the discriminator.

To further improve the similarity of the generated images with the real SEM images, an L_1 loss is incorporated

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y}[\|y - G(x)\|_1] \quad (2)$$

The final objective is a weighted sum of the adversarial loss in Eq. (1) and the L_1 loss in Eq. (2)

$$G^* = \arg \min_G \max_D \mathcal{L}_{\text{cGAN}}(G, D) + \lambda \mathcal{L}_{L1}(G), \quad (3)$$

where λ is a hyperparameter that balances the two losses. G tries to minimize this objective, whereas adversarial D tries to maximize it.

2.4.2 Synthetic SEM Image Examples Generated by cGAN

After constructing paired datasets consisting of defective contours and corresponding SEM images, we employ a cGAN to synthesize realistic SEM images based on input contour images. Specifically, we adopt the pix2pix framework,¹⁵ which is widely used for image-to-image translation tasks.

The architecture of the cGAN model follows the standard pix2pix configuration, comprising a U-Net-based generator and a PatchGAN discriminator.¹⁵ The generator maps input contour images to synthetic SEM outputs. The PatchGAN discriminator encourages the generator to produce images that are indistinguishable from real SEM images at the local patch level rather than evaluating the entire image at once. This patch-based discrimination helps improve local details while keeping the computational cost manageable.

However, during experiments on our dataset, we observed that using the original 70×70 PatchGAN configuration led to unwanted periodic patterns in the background of generated SEM images, as shown in Figs. 3(a) and 3(b). These results are probably caused by the limited receptive field of the discriminator, which encourages the generator to overfit to small local patterns. This is reasonable because our training dataset consists of pairs of 1024×1024 SEM images and their corresponding defective contours, with a size that is relatively large. In such high-resolution images, if the receptive field is too small, it only covers a minor portion of the image, resulting in the problem.

To address this, we increased the PatchGAN receptive field size, allowing the discriminator to evaluate larger regions of the image during training. This adjustment significantly improved the generated SEM image quality by eliminating background repetition patterns while preserving fine structural details of the contours, as shown in Figs. 3(c) and 3(d).

After completing model training, we applied the trained cGAN model to generate synthetic SEM images based on contour images with artificially introduced defects. Specifically, the defective contour images created through the defect introduction process described in Sec. 2.3 were used as inputs to the generator. Figure 4 shows an example of the synthetic SEM images generated by the trained cGAN model, using contour images with artificially introduced defects as inputs. The resulting synthetic SEM images have weak-signal regions or low-contrast areas, corresponding to the locations of defects in the input contours.

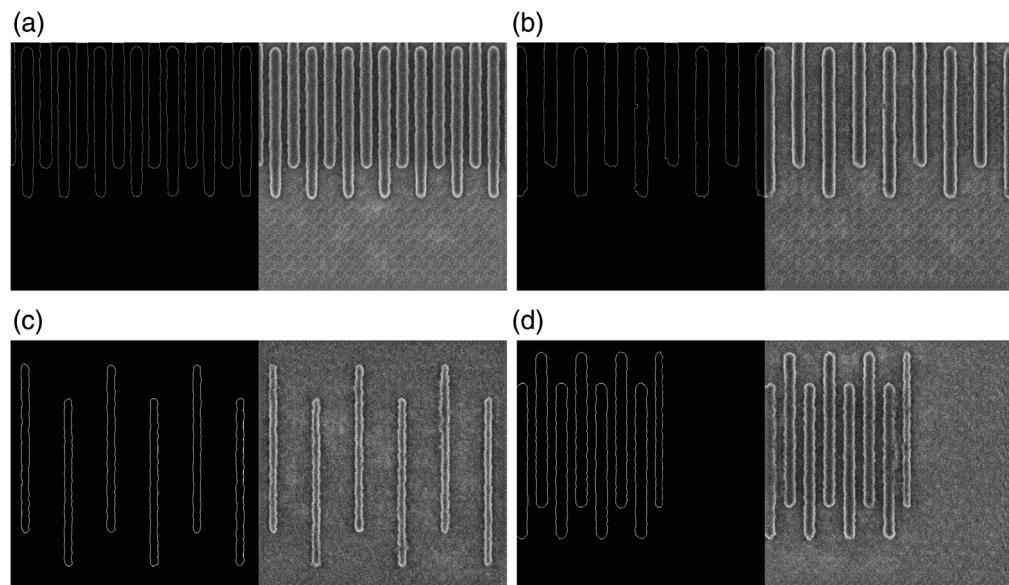


Fig. 3 Sample results of synthetic SEM images. (a), (b) Results with small PatchGAN showing periodic background artifacts. (c), (d) Results with large PatchGAN demonstrating improved results.

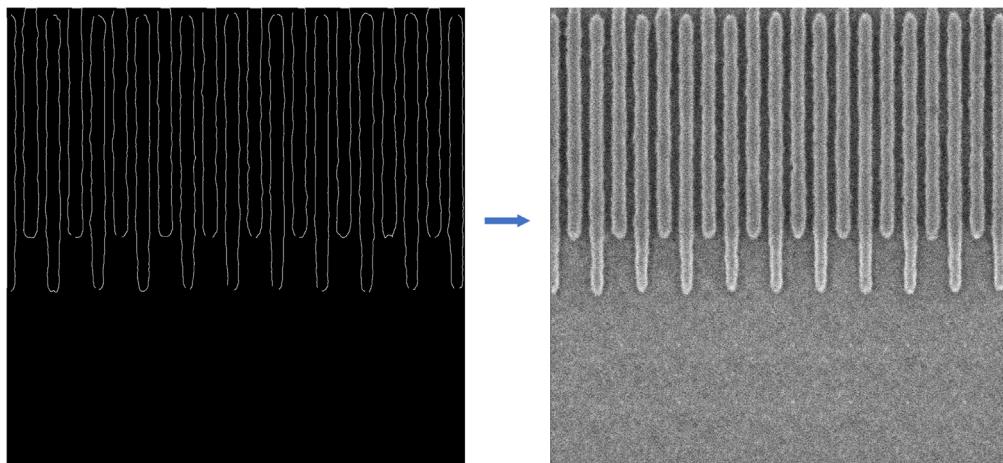


Fig. 4 Example of synthetic SEM image generated by the trained cGAN model using the contour image with artificially introduced defects, following the process described in Sec. 2.2.

2.4.3 BicycleGAN model

Although our cGAN-based model achieves high-quality synthetic SEM images, it still suffers from two important limitations. First, the cGAN is a deterministic model, meaning that a given input produces a single fixed output. This limits the diversity of the augmented training dataset and also increases the computational resources. Second, in our initial design, the entire contour image is used as input to the generator. However, the defective parts we want to do data augmentation are mainly around specific regions, such as line-end parts. This full-image generation strategy is computationally inefficient and unnecessarily increases the training time when the augmented dataset is used to retrain the contour extraction model. Hence, we need a “one-to-many” model and focus more on localized defect regions.

To further enrich the diversity of the synthetic SEM images and enhance the data augmentation process, we additionally trained a BicycleGAN model.²¹ Unlike standard cGAN, which produces a deterministic output for each input, BicycleGAN is a generative model that builds a one-to-many mapping between the input and multiple possible outputs. This property makes the model can generate diverse SEM images conditioned on the same contour input, which greatly enhances the diversity and efficiency of data augmentation.

In BicycleGAN, the diversity of outputs is controlled by a latent code z sampled from a Gaussian distribution $\mathcal{N}(0, I)$. Not only is the generator conditioned on the input contour image, but it also takes a latent vector z as an additional input. Different samples of z correspond to different possible changes in the output. Therefore, for the same input contour, varying the latent code can generate multiple different synthetic SEM images. The Gaussian distribution ensures that the latent space is continuous, is structured, and can produce realistic variations, thereby achieving meaningful diversity without sacrificing output quality.

BicycleGAN combines the conditional variation autoencoder GAN(cVAE-GAN) and conditional latent regressor GAN(cLR-GAN).²¹

In the cVAE-GAN, an encoding function E maps the ground truth y into a latent vector z . The generator G then reconstructs the SEM image from both input image x and latent code z . The loss function for this is

$$\begin{aligned} \mathcal{L}_{c\text{VAE-GAN}}(G, E, D) = & \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,y}[\log(1 - D(x, G(x, E(y))))] \\ & + \lambda_{L1} \|y - G(x, E(t))\|_1 + \lambda_{KL} D_{KL}(E(y) || \mathcal{N}(0, I)), \end{aligned} \quad (4)$$

where x is the input contour image and y is the corresponding real SEM image. The first two terms are similar to cGAN, which represent the standard conditional adversarial loss. The third term is reconstruction loss, which enforces similarity between the reconstructed image $(x, E(t))$ and the ground-truth SEM image y . The last term is a Kullback–Leibler divergence loss to make the encoded latent vector $E(y)$ match the standard Gaussian distribution $\mathcal{N}(0, I)$.

In the cLR-GAN, a latent code z is directly sampled from the Gaussian distribution $\mathcal{N}(0, I)$. The generator G produces an output $G(x, z)$ conditioned on x and z . To ensure consistency between the latent codes and the latent representation extracted from the generated images, the encoder E is trained to recover the original z from $G(x, z)$. The loss function is

$$\mathcal{L}_{\text{cLR-GAN}}(G, E, D) = \mathbb{E}_{x, z \sim p(z)}[\log(1 - D(x, G(x, z)))] + \lambda_{\text{latent}} \|z - E(G(x, z))\|_1, \quad (5)$$

where the first term is an adversarial loss that makes output $G(x, z)$ realistic under the conditional discriminator D . The second term is latent code consistency loss. It penalizes the difference between the latent code z and the code recovered by $E(G(x, z))$. The final objective is the sum of the cVAE-GAN and cLR-GAN objectives in Eqs. (4) and (5)

$$G^*, E^* = \arg \min_{G, E} \max_D \mathcal{L}_{\text{cVAE-GAN}}(G, E, D) + \mathcal{L}_{\text{cLR-GAN}}(G, E, D), \quad (6)$$

2.4.4 BicycleGAN model

We then employ this BicycleGAN model to synthesize realistic SEM images based on input contour images. The training dataset was the same as the one used for cGAN. Here, to adapt the model for focusing specifically on local open line-end defects, we first cut each 1024*1024 image into 256*256 patches, which are then used to train the BicycleGAN model. For the generative task, we only prepared the ideal contours of the line-end regions, and artificial defects were introduced in the same way described in Sec. 2.3.

For each input contour image, multiple synthetic SEM images were generated by sampling different latent codes. Figure 5 shows an example where, with a single input contour image, the model generates seven different possible output SEM images. By increasing the number and variability of training samples, this approach improved the model's ability to generalize to challenging cases involving line-end defects.

2.5 Retraining of the Previous Contour Extraction Model with Augmented Data

After generating many synthetic SEM images, we paired them with the corresponding original ideal contour images. Then, we utilized this augmented dataset to retrain and improve the performance of our previously developed contour extraction model.⁸

Specifically, we generated 1500 1024*1024 synthetic SEM/contour pairs from cGAN and 20160 256*256 synthetic SEM/contour pairs from BicycleGAN, where each synthetic SEM image was generated using the trained image generative model based on a contour image with artificially introduced defects. These synthetic pairs were incorporated as additional training data to supplement the original dataset, addressing the lack of diverse SEM image/contour pairs and improving model robustness.

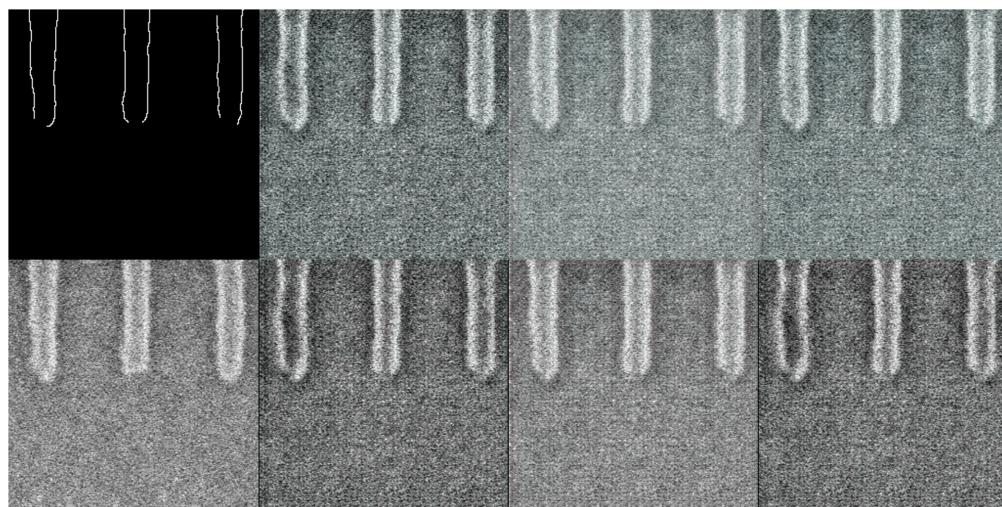


Fig. 5 Example of synthetic SEM images generated by the trained BicycleGAN model. Contours that were made artificially defective with open line-ends are used as inputs to generate multiple synthetic SEM images.

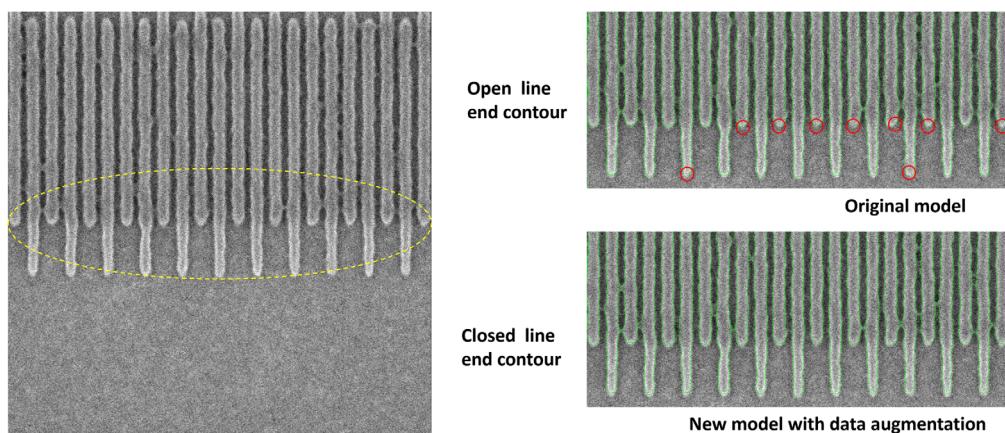


Fig. 6 Comparison of contour extraction results. (a) Original model result showing open line-end defects and partially broken contours. (b) Retrained model result using the augmented dataset, achieving full closure of contour line-ends and improved contour continuity.

The contour extraction model employed in this study is the same ML-based framework proposed in our previous work,⁸ which was specifically designed for efficient and accurate contour extraction for SEM images. The model was retrained using the combined dataset consisting of original data and the newly generated synthetic data.

By augmenting the training dataset with synthetic SEM images with weak-signal or low-contrast regions, paired with their corresponding ideal contour images, the model's ability to handle challenging cases was significantly enhanced, as will be demonstrated in Sec. 3.

3 Results and Discussion

When applying our previous contour extraction model to this set of 360 SEM images, open line-end defects frequently appeared in the extracted contours, as shown in Fig. 1, significantly limiting the model's reliability.

To address this issue, we designed the data augmentation pipeline described in this work, aiming to improve the model's ability to extract continuous and closed line-end contours under challenging cases. The pipeline introduces controlled defects into ideal contour images and synthesizes corresponding SEM images with weak-signal or low-contrast regions.

After retraining the contour extraction model using the augmented dataset, we applied the new model to the same set of 360 SEM images to evaluate the improvement in line-end parts. As shown in Fig. 6, the new model achieved significantly better results: the previously frequently occurring unclosed line-ends are eliminated, and the continuity of the extracted contours is also enhanced.

In the original model's results, ~15% of the 360 evaluated SEM images showed open line-end defects. After retraining with the augmented dataset, the new model achieved around 99% closure of contour line-ends across all images in this dataset.

This clear improvement demonstrates the effectiveness and feasibility of the proposed data augmentation pipeline. By specifically targeting the main failure defects encountered during SEM images contour extraction tasks, the retrained model achieved much greater robustness and accuracy in handling weak-signal and low-contrast regions.

4 Conclusion

In this work, we proposed a targeted data augmentation pipeline to improve contour extraction performance on SEM images, specifically addressing the open line-end defects observed in the outputs of our previous model. By introducing controlled defects into ideal contour images and generating corresponding synthetic SEM images with weak-signal or low-contrast regions, we expanded the training dataset in a way that directly reflects the failure cases encountered in applications.

Using the augmented dataset, we retrained our previously developed ML-based contour extraction model. Evaluation on the same set of 360 SEM images showed significant improvements: The extracted contours from the original model had open line-end defects in ~15% of the cases. The retrained model significantly reduced the occurrence of open line-end defects, achieving complete closure of contour line-ends in ~99% of the evaluated images.

These results demonstrate the effectiveness of the proposed data augmentation pipeline in enhancing model performance. Beyond the specific case of open line-end defects, this approach can be generalized to address other types of defects in contour extraction tasks, offering a practical strategy for improving contour extraction model performance.

Disclosures

This work was initiated and led by Siemens Digital Industries Software. The first author is currently a PhD candidate at the University of Southern California and participated in this work during an internship at Siemens Digital Industries Software from May 2024 to January 2025. Prof. Wei Wu is the first author's PhD advisor. Other co-authors are employees of Siemens Digital Industries Software or IMEC. The SEM images used in this work were provided by IMEC as part of a joint research collaboration. The authors declare no conflicts of interest.

Code and Data Availability

The SEM images used in this study were obtained under a research collaboration between Siemens Digital Industries Software and IMEC. Due to confidentiality agreements and proprietors' restrictions, the data cannot be shared publicly.

Acknowledgments

Language and grammar improvements were assisted by ChatGPT, developed by OpenAI. A preliminary version of this work was presented as a poster and published in the proceedings of the 2025 SPIE Advanced Lithography + Patterning conference⁹ under the title "Machine learning-based on SEM images, contour extraction, and data augmentation."

References

1. L. Reimer, "Scanning electron microscopy: physics of image formation and microanalysis," *Meas. Sci. Technol.* **11**, 1826–1826 (2000).
2. J. I. Goldstein et al., *Scanning Electron Microscopy and X-Ray Microanalysis*, Springer (2017).
3. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, Inc. (1983).
4. J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 679–698 (1986).
5. A. Kirillov et al. "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, Paris, France, 2023, pp. 3992–4003.
6. M. Oquab et al., "DINOv2: learning robust visual features without supervision," arXiv:2304.07193 (2023).
7. Y. Ma et al., "Monotonic machine learning for lithography retargeting layer generation by leveraging contour-based metrology," *Proc. SPIE* **13216**, 132161V (2024).
8. Y. Ma et al., "Machine learning (ML) based SEM contour extraction accelerated by GPU for etch modeling application," in *Advanced Etch Technology and Process Integration for Nanopatterning XIII*, SPIE (2024).
9. H. Zhang et al. "Machine learning-based SEM images contour extraction data augmentation," *Proc. SPIE* **13426**, 134262X (2025).
10. H. Demers et al., "Three-dimensional electron microscopy simulation with the CASINO Monte Carlo software," *Scanning* **33**, 135–146 (2011).
11. I. J. Goodfellow et al., "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst.* 27 (2014).
12. D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv:1312.6114 (2013).
13. J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Adv. Neural Inf. Process. Syst.* 33, pp. 6840–6851 (2020).
14. M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv:1411.1784 (2014).
15. P. Isola et al., "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 5967–5976.
16. Y. Jiang et al., "COVID-19 CT image synthesis with a conditional generative adversarial network," *IEEE J. Biomed. Health Inf.* **25**, 441–452 (2020).

17. H. Yang et al., "Applying conditional generative adversarial networks for imaging diagnosis," in *IEEE 6th Int. Conf. Power, Intell. Comput. and Syst. (ICPICS)*, IEEE (2024).
18. R. Raad et al., "Conditional generative learning for medical image imputation," *Sci. Rep.* **14**, 171 (2024).
19. V. Kosaraju et al., "Social-BiGAT: multimodal trajectory forecasting using bicycle-GAN and graph attention networks," in *Adv. Neural Inf. Process. Syst.* **32** (2019).
20. Y. Li et al., "3D reconstruction of unlimited-size real-world porous media by combining a BicycleGAN-based multimodal dictionary and super-dimension reconstruction," *Geoenergy Sci. Eng.* **228**, 212005 (2023).
21. J.-Y. Zhu et al., "Toward multimodal image-to-image translation," in *Adv. Neural Inf. Process. Syst.* **30** (2017).

Hongming Zhang received his PhD in electrical engineering from the University of Southern California in June 2025. He received his MS degree in computer science from USC in 2024 and his BS degree in physics from Fudan University in 2019. He is currently with Siemens Digital Industries Software. His research interests include integrating machine learning and deep learning with nanotechnology.

Biographies of the other authors are not available.