Simultaneous Denoising and Edge Estimation from SEM Images using Deep Convolutional Neural Networks

Narendra Chaudhary
Dept. of Electrical and Computer Engineering
Texas A&M University
College Station, Texas 77843-3128
Email: narendra5@tamu.edu

Serap A. Savari
Dept. of Electrical and Computer Engineering
Texas A&M University
College Station, Texas 77843-3128
Email: savari@ece.tamu.edu

***Abstract*—We propose deep convolutional neural networks LineNet1 and LineNet2 for simultaneous denoising and edge image prediction from low-dose scanning electron microscope images. Edge estimation of nanostructures from SEM images is needed for line edge roughness (LER) and line width roughness (LWR) estimation. Our method uses supervised learning datasets of single-line SEM images and multiple-line SEM images together with edge positions information for the training of LineNet1 and LineNet2. We simulate single-line and multiple-line SEM images with Poisson noise and other artifacts using the ARTIMAGEN library developed by the National Institute of Standards and Technology. The line edges were generated using the Thorsos method and the Palasantzas spectral model. The convolutional neural networks LineNet1 and LineNet2 each contain 17 convolutional layers, 16 batch-normalization layers and 16 dropout layers. Our results show that this approach (1) facilitates edge estimation in multiple-line images and (2) significantly reduces the memory needed for edge estimation in single-line images with a slight impact on accuracy.**

***Index Terms*—Deep learning, SEM metrology, line edge roughness, denoising, deep convolutional neural networks.**

I. Introduction

Scanning electron microscopy (SEM) images are used in semiconductor metrology to estimate the geometry of nanostructures. The estimation of nanostructure geometries is difficult in low-dose SEM images as they are corrupted by Poisson noise and other artifacts. The correct estimation of line roughness and contour geometry roughness is needed to measure the quality of lithographic processes and semiconductor performance. Line edge roughness (LER) and line width roughness (LWR) are popular metrics to measure certain aspects of the roughness of lines. There are multiple earlier approaches to estimating roughness involving filtering [1] and physics-based model fitting methods [2]. Filtering based approaches can change the original edge geometry, and model fitting methods are constrained by the modeling assumptions.

In recent years deep learning has influenced the practice of image processing [3], [4] and natural image classification problems [5], [6], [7]. We had previously proposed and demonstrated the superiority of the deep convolutional neural network (CNN) SEMNet in the Poisson denoising of SEM images with a single rough line [8]. SEMNet [8] has 17 convolutional [9], 16 batch-normalization [10] and 16 dropout [11] layers; and it was used to directly predict denoised images from original noisy images. We applied the Canny edge detector to estimate edges from denoised images. We subsequently proposed the neural network EDGENet to predict the two edge positions in single rough line images [12]. EDGENet [12] offers an alternative to traditional edge detection algorithms with filters and automates the edge detection procedure. However, EDGENet can only predict edges in single-line SEM images. In this paper, we propose a generic method for simultaneously denoising and predicting the edge image. We consider these tasks together to (1) facilitate machine learning, (2) save time, and (3) handle a larger class of SEM images. We introduce the deep convolutional neural networks LineNet1 for noisy SEM images with a single rough line, and we extend and generalize our approach in LineNet2 to noisy SEM images with multiple rough lines.

We trained LineNet1 with the same dataset of 100800 simulated single rough line SEM images used to train SEMNet [8] and EDGENet [12]. We trained LineNet2 with a different dataset of 50,400 simulated multiple rough line SEM images. The initial line edges were generated using the Thorsos [13] method with an underlying Palasantzas [14] spectral model. The rough line images were generated using the ARTIMAGEN SEM simulator library [15][16]. LineNet1 and LineNet2 predict grayscale edge images, and we use a peak detection algorithm to estimate exact edge positions. We also process larger images by splitting them into smaller images and using LineNet2 for the processing of the smaller images. This flexibility enables line edge roughness estimation through the use of multiple lines edges.

In Section II, we discuss the proposed neural network architectures. In Section III, we discuss simulation and the training datasets. In Section IV, we provide the LER/LWR accuracy results. In Section V, we conclude the paper.

978-1-5386-7601-1/19/$31.00 ©2019 IEEE

II. Neural networks

We devised LineNet1 and LineNet2 to estimate denoised rough line images and the corresponding edge images from noisy images with arbitrary levels of Poisson noise. The paper [17] pointed out that neural networks can share information while performing multiple image processing tasks. LineNet1 and LineNet2 can save both time and memory by performing the tasks previously achieved by the two neural networks SEMNet [8] and EDGENet [12]. LineNet1 and LineNet2 use neural network architectures similar to the SEMNet [8] architecture; i.e., they all have 17 convolutional [9], 16 batch-normalization [10] and 16 dropout [11] layers. Each convolutional layer of LineNet1 and LineNet2 except for the last layer has 64 filters of dimension $(3 \times 3 \times input\ depth)$ with bias terms and is followed by a batch normalization layer and a dropout layer with dropout probability of 0.2. The last convolutional layer of LineNet1 and LineNet2 each has two filters to output two grayscale images.

LineNet1 inputs a single-line SEM image of dimension 64×1024 pixels and outputs an array of dimension $64 \times 1024 \times 2$. The output array contains a denoised image and a grayscale edge image. Similarly, LineNet2 inputs a multiple-line SEM image of dimension 256×256 pixels and outputs an array of dimension $256 \times 256 \times 2$. We can estimate a larger denoised image and the corresponding edge image of larger dimension by first splitting the larger image into smaller images of dimension 256×256 pixels. We elected to study images with a pixel size of 0.5×2 nm, but the approach applies generally.

The rectified linear unit (ReLU) [7] is a widely used nonlinearity which is defined by the following half wave rectifier:

$$y = \max(0, x) \qquad \text{(Rectified linear unit (ReLU)).}$$

We use ReLU as the nonlinearity in LineNet1 and in LineNet2. Two widely used loss functions in deep learning are mean squared error (MSE) and mean absolute error (MAE)

$$\text{MSE} = \frac{1}{N} * \sum_{i=1}^{N} \|Y^i - f(X^i)\|_2^2. \tag{1}$$

$$\text{MAE} = \frac{1}{N} * \sum_{i=1}^{N} \|Y^i - f(X^i)\|_1. \tag{2}$$

We use MSE as the loss function for LineNet1 and LineNet2, since we found that the predicted edge image can have multiple missing edge positions if we instead use mean absolute error (MAE). LineNet1 and LineNet2 each have 559,810 parameters compared to the 10,972,993 parameters of EDGENet [12].

III. Simulation and Training Datasets

We trained LineNet1 with the same dataset of 100800 simulated single-line SEM images used to train SEMNet [8] and EDGENet [12]. The images have dimension of 64×1024 pixels with a pixel size of 0.5×2 nm. The initial line edges were generated using the Thorsos [13] method with an underlying Palasantzas [14] spectral model, where the parameters are described below.

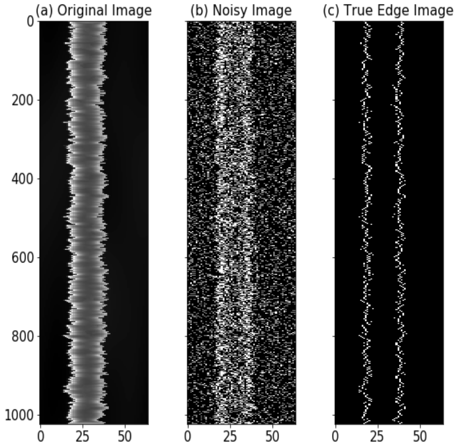


Fig. 1. Single-line dataset images. (a) An original simulated SEM image of dimension 64×1024. (b) Noisy image after Poisson noise addition. (c) Edge image. The aspect ratio of the images has been scaled for a better view.

$$PSD(f) = \frac{\sqrt{\pi}\Gamma(\alpha + 0.5)}{\Gamma(\alpha)} \cdot \frac{2\sigma^2\xi}{(1 + (2\pi f \xi)^2)^{\alpha+0.5}}.$$

We selected a wide range of values for the LER (σ = 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8 nm), the Hurst/roughness exponent (α = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9) and the correlation length (ξ = 6, ..., 40) parameters to generate rough edges and create rough lines from them. For each of the 2520 possible combinations of parameters we generated eight edges. The rough line images were generated using the ARTIMAGEN SEM simulator library [15][16] and included the artifacts of random backgrounds, a fixed edge effect, fine structure and Gaussian blur. Our original image set is the collection of 10080 images. From each original image we generated ten noisy images corrupted by Poisson noise with electron density per pixel in the range $\{2,\ 3,\ 4,\ 5,\ 10,\ 20,\ 30,\ 50,\ 100,\ 200\}$. Thus, the noisy image dataset consists of 100800 images. From our initial line edges, we also generated 10080 binary edge images. We constructed a supervised learning dataset with the noisy image as input x^i and an array of the original image with edge image as output y^i. Figure 1 shows one example of single-line dataset images with a noisy image, an original image and a corresponding edge image.

We trained LineNet2 on a different dataset of 50,400 simulated images with dimension of 256×1024 pixels. Each larger image contains three or four rough lines of width 15nm or 10nm separated by a space of twice the linewidth. The locations of the lines within the images vary. The rough edges in these images were also constructed using the Thorsos method with the same combinations of parameters for LER, roughness exponent and correlation length. LineNet2 is designed to construct denoised and estimated edge images of size 256×256 pixels. Hence the initial images of dimensions

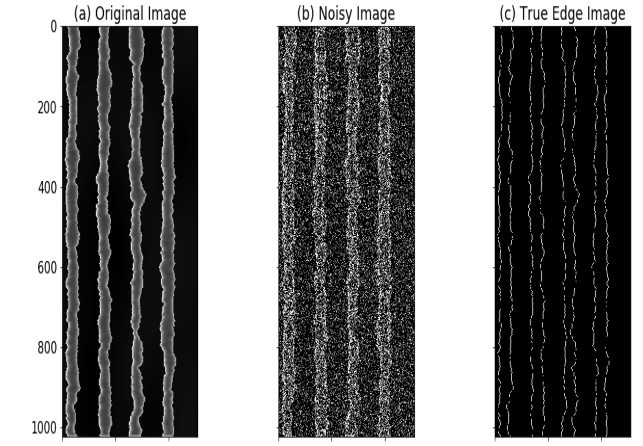


Fig. 2. Multiple-line dataset images. (a) An original simulated SEM image of dimension 256×1024. (b) Noisy image after Poisson noise addition. (c) Edge image. The aspect ratio of the images has been scaled for a better view.

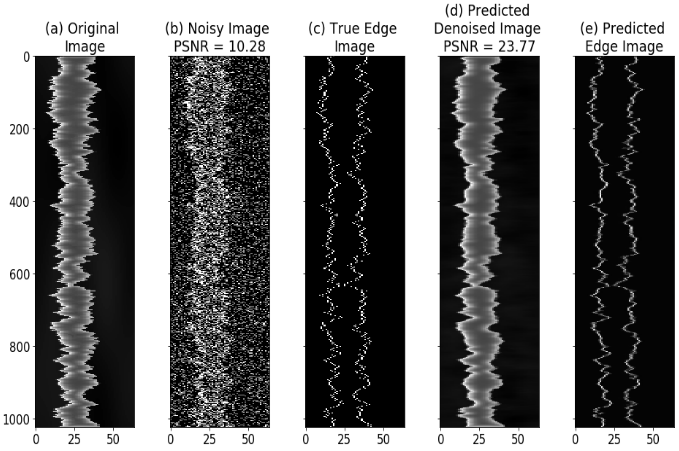


Fig. 3. Results with LineNet1: (a) Original image with $\sigma = 1.6$nm, $\alpha = 0.5$, $\xi = 30$nm. (b) Noisy image with a noise level of 2 electrons per pixels. (c) Edge image (d) Predicted denoised image. (e) Predicted edge image.

256×1024 were each separated into four nonoverlapping images before being processed by LineNet2. Each set of four denoised and estimated edge images were later recombined to form an image of dimension 256×1024. This approach can be generalized to handle arbitrary images. Even though the number of pixels in the dataset images of LineNet1 and LineNet2 are same, the estimation problem is harder in LineNet2 due to multiple lines within the images. Furthermore, additional complexity arises from having a variable number of lines in an image instead of a fixed number of lines. We believe the approach of LineNet2 can be extended in the future to images with arbitrary contours. Figure 2 shows one example of a multiple-line dataset image with a noisy image, an original image, and a corresponding edge image.

IV. Experiments and Results

The simulated dataset for LineNet1 was divided into three subsets, namely the training set, the validation set and the test set. The test set consisted of the 8640 noisy-original SEM images and edge images with correlation length ξ in the set $\{10,\ 30,\ 40\}$ nm. The validation set consisted of the 2880 noisy-original SEM images and edge images with correlation length $\xi = 20$ nm. The training set consisted of the remaining 89280 noisy-original SEM images and edge images. The simulated dataset for LineNet2 was also divided into a training set, a validation set and a test set. These LineNet2 dataset images had dimension 256×1024 pixels. The test set consisted of the 4320 noisy-original SEM images and edge images with correlation length ξ in the set $\{10,\ 30,\ 40\}$ nm. The validation set consisted of the 1440 noisy-original SEM images and edge images with correlation length $\xi = 20$ nm. The training set consisted of the remaining 44,640 noisy-original SEM images and edge images.

The training processes for both LineNet1 and LineNet2 involved solving an optimization problem to minimize the mean squared error (MSE) between the predicted images and the original images. We applied a stochastic gradient descent algorithm Adam [18] for training. We used the python deep learning libraries Keras and TensorFlow to construct and train the deep neural networks in our experiments. We trained the neural networks with a Tesla K80 graphical processing unit (GPU). The results presented here use test dataset images which were not utilized during the training process. Figure 3 demonstrates the effectiveness of simultaneously denoising and estimating edge images in single-line images using LineNet1.

Figures 4 and 5 demonstrate the effectiveness of simultaneously denoising and estimating edge images in multiple-line images using LineNet2. The edge images predicted from LineNet1 and LineNet2 are grayscale images with pixel intensity between 0 and 1 instead of binary images. The grayscale values in a row of an edge image have approximate peaks. Figure 6 shows the pixel intensity values in a row of a grayscale edge image with four lines / eight edges. We apply the scientific python (SciPy) peak detector function find_peaks to estimate the exact edge positions in the predicted edge images.

Table I provides the peak signal-to-noise ratio (PSNR) results of LineNet1 and SEMNet [8]. Observe that the PSNR values are close even though LineNet1 is devoting some resources to edge detection. Table II provides the line roughness results of LineNet1 and EDGENet [12] on single-line images. EDGENet offers better accuracy than LineNet1 in the case of both high noise and high frequency at the price of significantly higher memory; recall that LineNet1 has only 559,810 parameters compared to the 10,972,993 parameters of EDGENet. LineNet1 and EDGENet have similar performance in other cases. Figure 7 compares the predicted edge spectra of LineNet1 and EDGENet for three different types of edges. Observe that the edge spectra do not significantly differ. Table III provides PSNR and line roughness results for multiple-line images of dimension 256×1024 pixels. The line

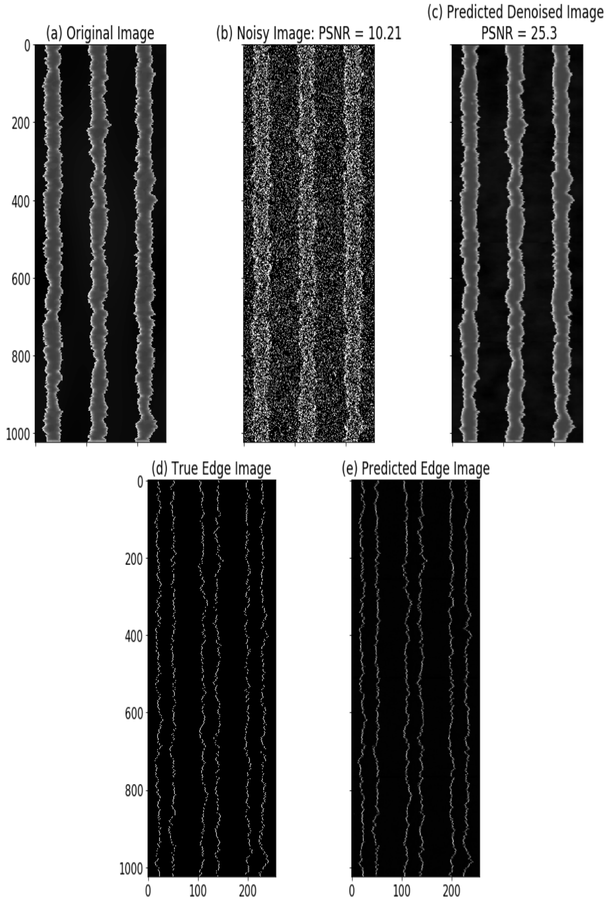


Fig. 4. Results with LineNet2: (a) Original image with $\sigma = 1.6$nm, $\alpha = 0.5$, $\xi = 30$nm, linewidth = 15nm. (b) Noisy image with a noise level of 2 electrons per pixels. (c) Predicted denoised image. (d) Edge image. (e) Predicted edge image.

edge roughness data in Table III come from the average of multiple line edges in a single image. In summary, our results show that LineNet1's performance for image denoising and line edge roughness (LER) estimation are comparable to those of SEMNet [8] and EDGENet [12] on the same set of images. The average prediction time per image of LineNet1 and LineNet2 was 0.12 seconds on the K-80 GPU. In comparison, prediction time per image of EDGENet was 0.15 seconds and that of SEMNet was 0.12 seconds on the K-80 GPU; recall, however that LineNet1 and LineNet2 are followed by a peak detection algorithm while EDGENet is a stand-alone line edge detection algorithm. Additionally, LineNet1 takes approximately 41 hours to train while SEMNet and EDGENet each take more than 40 hours to train.

V. Conclusion and Future work

Deep learning offers an effective method for the simultaneous denoising and edge estimation of SEM images. Our results show that time and memory resources can be saved by performing these tasks simultaneously using neural networks. We have also shown that our new method applies to a more

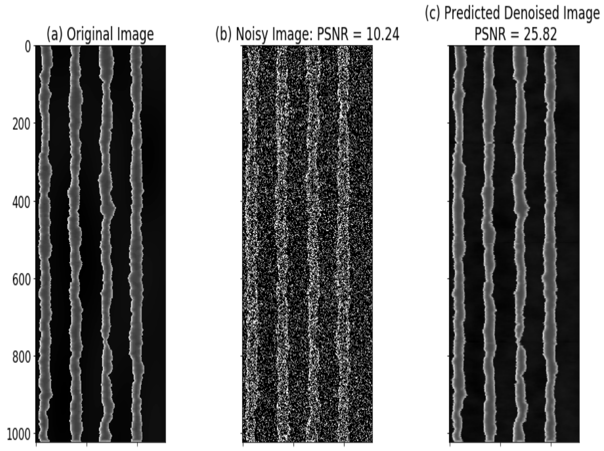


Fig. 5. Results with LineNet2 (a) Original image with $\sigma = 1.2$nm, $\alpha = 0.8$, $\xi = 40$nm, linewidth = 10nm. (b) Noisy image with a noise level of 2 electrons per pixels. (c) Predicted denoised image.

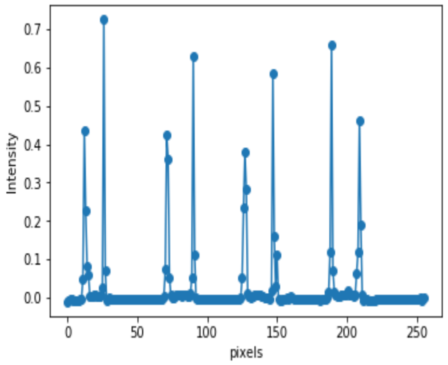
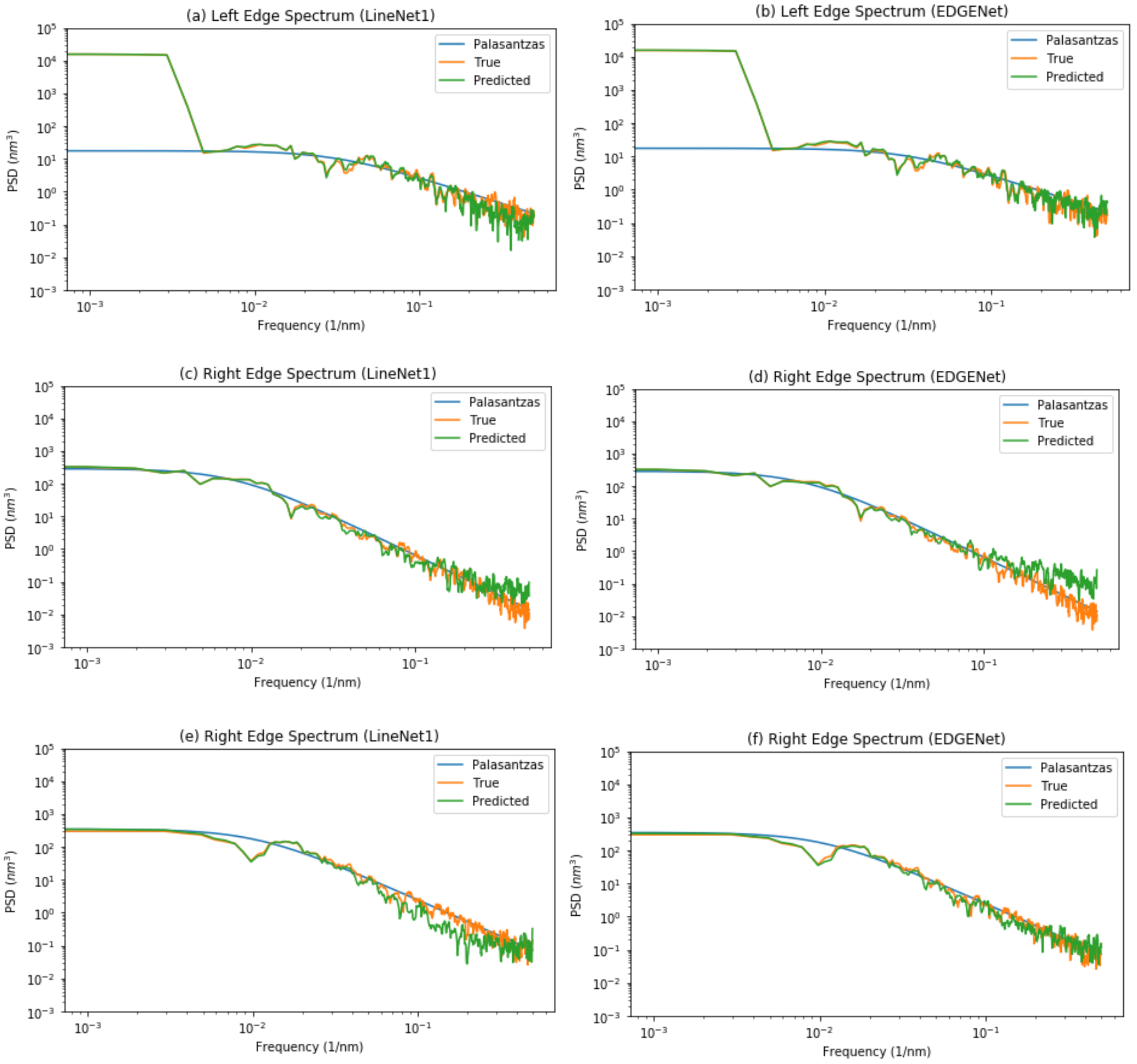


Fig. 6. The pixel intensity values in one row of a predicted grayscale edge image. The original image has four lines / eight edges.

TABLE I
Denoising results for single-line images with LineNet1 and SEMNet.

Original Image σ(nm), ξ(nm), α	Poisson noise level	PSNR (dB) Noisy Image	 Predicted LineNet1	 Predicted SEMNet
0.8, 10, 0.3	2	10.35	23.89	24.27
0.8, 10, 0.3	5	13.64	25.58	26.02
0.8, 10, 0.3	10	16.38	28.26	28.66
0.8, 10, 0.3	100	26.18	41.87	42.89
1.2, 40, 0.7	2	9.96	26.39	26.39
1.2, 40, 0.7	5	13.20	29.07	29.0
1.2, 40, 0.7	10	15.96	31.22	31.26
1.2, 40, 0.7	100	25.71	42.07	42.79
1.6, 30, 0.5	2	10.28	23.77	24.09
1.6, 30, 0.5	5	13.67	26.33	26.65
1.6, 30, 0.5	10	16.33	28.89	29.14
1.6, 30, 0.5	100	26.13	41.47	42.64


(a) Left Edge Spectrum (LineNet1)

(b) Left Edge Spectrum (EDGENet)

(c) Right Edge Spectrum (LineNet1)

(d) Right Edge Spectrum (EDGENet)

(e) Right Edge Spectrum (LineNet1)

(f) Right Edge Spectrum (EDGENet)

Fig. 7. (a), (b) LineNet1 and EDGENet predicted edge spectra for $\sigma = 0.8$ nm, $\alpha = 0.3$, $\xi = 10$ nm, Poisson noise = 10. (c), (d) LineNet1 and EDGENet predicted edge spectra for $\sigma = 1.2$ nm, $\alpha = 0.7$, $\xi = 40$ nm, Poisson noise = 5. (e), (f) LineNet1 and EDGENet predicted edge spectra for $\sigma = 1.6$ nm, $\alpha = 0.6$, $\xi = 30$ nm, Poisson noise = 2.

TABLE II
ROUGHNESS RESULTS FOR SINGLE-LINE IMAGES WITH LINENET1 AND EDGENET.

Original Image σ(nm), ξ(nm), α	Poisson noise level	Left edge (nm) σ			Right edge (nm) σ			LWR (nm) σ		
			σ (observed) LineNet1	EDGENet		σ (observed) LineNet1	EDGENet		σ (observed) LineNet1	EDGENet
0.8, 10, 0.3	2	0.75	0.64	0.71	0.74	0.66	0.66	1.06	0.89	0.95
0.8, 10, 0.3	5	0.75	0.71	0.73	0.74	0.65	0.67	1.06	0.96	0.97
0.8, 10, 0.3	10	0.75	0.73	0.75	0.74	0.72	0.72	1.06	1.04	1.04
0.8, 10, 0.3	100	0.75	0.75	0.75	0.74	0.74	0.74	1.06	1.06	1.06
1.2, 40, 0.7	2	1.06	1.03	1.06	1.21	1.26	1.23	1.41	1.42	1.42
1.2, 40, 0.7	5	1.06	1.06	1.06	1.21	1.20	1.22	1.41	1.40	1.42
1.2, 40, 0.7	10	1.06	1.06	1.08	1.21	1.20	1.21	1.41	1.41	1.42
1.2, 40, 0.7	100	1.06	1.06	1.06	1.21	1.21	1.21	1.41	1.41	1.42
1.6, 30, 0.5	2	1.56	1.50	1.53	1.59	1.49	1.46	2.34	2.27	2.23
1.6, 30, 0.5	5	1.56	1.55	1.57	1.59	1.54	1.52	2.34	2.30	2.28
1.6, 30, 0.5	10	1.56	1.54	1.57	1.59	1.61	1.59	2.34	2.34	2.33
1.6, 30, 0.5	100	1.56	1.56	1.56	1.59	1.60	1.58	2.34	2.34	2.33

TABLE III
RESULTS FOR MULTIPLE-LINE IMAGES WITH LINENET2.

Original Image	Poisson	PSNR (dB)		Mean LER (nm)	
σ(nm), ξ(nm), α, linewidth(nm)	noise level	Noisy Image	Predicted Image	σ (true)	σ (obs.)
0.8, 10, 0.3, 10	2	10.18	24.23	0.76	0.75
0.8, 10, 0.3, 10	5	13.48	26.52	0.76	0.75
0.8, 10, 0.3, 10	10	16.22	28.90	0.76	0.76
0.8, 10, 0.3, 10	100	26.03	42.33	0.76	0.76
1.2, 40, 0.8, 10	2	10.24	25.82	1.16	1.18
1.2, 40, 0.8, 10	5	13.54	28.71	1.16	1.17
1.2, 40, 0.8, 10	10	16.33	31.22	1.16	1.17
1.2, 40, 0.8, 10	100	26.11	42.50	1.16	1.16
1.6, 30, 0.5, 15	2	10.21	25.30	1.52	1.48
1.6, 30, 0.5, 15	5	13.58	27.74	1.52	1.52
1.6, 30, 0.5, 15	10	16.35	30.22	1.52	1.52
1.6, 30, 0.5, 15	100	26.17	42.60	1.52	1.52

complex class of multiple-line SEM images while our earlier neural networks SEMNet and EDGENet were trained for single-line images. In the future, we plan to estimate the edges in SEM images of contour nanostructure geometries.

ACKNOWLEDGMENT

The authors used the Texas A&M University High Performance Research Computing Facility to conduct part of the research.

REFERENCES

[1] V. Constantoudis, G. Patsis, A. Tserepi, and E. Gogolides, "Quantification of line-edge roughness of photoresists. ii. Scaling and fractal analysis and the best roughness descriptors," *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, vol. 21, no. 3, pp. 1019–1026, 2003.

[2] C. A. Mack and B. D. Bunday, "Using the analytical linescan model for SEM metrology," *Metrology, Inspection, and Process Control for Microlithography XXXI*, vol. *Proc. of SPIE* 10145, p. 101451R, 2017.

[3] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[4] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[8] N. Chaudhary, S. A. Savari, and S. S. Yeddulapalli, "Deep supervised learning to estimate true rough line images from SEM images," *34th European Mask and Lithography Conference*, vol. *Proc. of SPIE* 10775, p. 107750R, 2018.

[9] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, p. 1995, 1995.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, 07–09 Jul 2015.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] N. Chaudhary, S. A. Savari, and S. S. Yeddulapalli, "Automated rough line-edge estimation from SEM images using deep convolutional neural networks," *Photomask Technology 2018*, vol. *Proc. of SPIE* 10810, p. 108101L, 2018.

[13] E. I. Thorsos, "The validity of the Kirchhoff approximation for rough surface scattering using a Gaussian roughness spectrum," *The Journal of the Acoustical Society of America*, vol. 83, no. 1, pp. 78–92, 1988.

[14] G. Palasantzas, "Roughness spectrum and surface width of self-affine fractal surfaces via the k-correlation model," *Physical Review B*, vol. 48, no. 19, p. 14472, 1993.

[15] P. Cizmar, A. E. Vladár, B. Ming, and M. T. Postek, "Simulated SEM images for resolution measurement," *Scanning*, vol. 30, no. 5, pp. 381–391, 2008.

[16] P. Cizmar, A. E. Vladár, and M. T. Postek, "Optimization of accurate SEM imaging by use of artificial images," *Scanning Microscopy 2009*, vol. 7378, p. 737815, 2009.

[17] E. Schwartz, R. Giryes, and A. M. Bronstein, "DeepISP: Learning end-to-end image processing pipeline," *arXiv preprint arXiv:1801.06724*, 2018.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015.