CS230

Label-free SEM Content Recognition with Cycle-consistent Generative Adversarial Networks (Computer Vision)

Ching-Ting Tsai
Stanford University
tingtsai@stanford.edu

Xiaohan Li
KLA Corporation
xiaohan1@stanford.edu

Xiaojun Sun
KLA Corporation
sunxj@stanford.edu

Csaba Forró
Stanford University
cforro@stanford.edu

Abstract

Deep learning-based approaches for image segmentation have been widely applied in a range of fields. In material science, segmenting scanning electron microscope (SEM) images is of particular interest, given its importance in feature detection and characterization. However, traditional neural networks perform on image segmentation requires laborious labeling efforts. Inspired by CycleGAN networks' ability to perform style transfer with unpaired examples, we propose a label-free method for SEM image segmentation. The SEM images can be trained with original lithographic designs or nanostructures' labels to achieve style transfer. The results benchmark against the supervised ResNet approach shows that this method is competitive in SEM segmentation and can be further used in synthetic SEM image generation.

1 Introduction

The scanning electron microscope (SEM) plays a crucial role in characterizing a wide range of materials, including semiconductors, nanoelectronics, or photonic devices. Image segmentation for SEM images is a critical application in nanoscience. The ability to identify features from the nanostructures is useful for researchers and manufacturers to evaluate their patterning capabilities. However, currently, there is a lack of a suitable tool to segment SEM images efficiently. Previous research for SEM image recognition has focused on classification but little on segmentation. [1] Here, we introduce both supervised and unsupervised models to address this research question. The models' input will be SEM images, and output will be segmented images highlighting the features.

2 Related work

Deep convolutional neural networks (CNNs) have proved extremely useful in the segmentation problem.[2] For the task of pixels classification for image segmentation, Res-Net is one of the widely applied algorithms. [3] It adds a skip connection to the networks to prevent the loss of the original input information. This design helps solve the vanishing gradient problem and allows for training in a deep neural network. However, a model based on CNN usually required paired datasets, which

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

means the researchers have to provide images and corresponding labels for training. This practice could be problematic since labeling data requires lots of manual efforts, and the amount of data to be processed is usually unclear.

CycleGAN is a model to process image-to-image translation without paired training data. [4] The architecture without needing paired data makes this approach very flexible because one can replace labeled with synthetic data that share similarities with the images to perform segmentation. These concepts have been demonstrated applied to image segmentation tasks like images of cell cultures. [5] As for the SEM image segmentation, we stand on a more advantageous point. The lithography design of the nanostructures will be a good approximation for the shape in our SEM image. Therefore, we can directly put lithography input generated to replace the original labels in the CycleGAN to achieve label-free image segmentation.

3 Dataset and Features

In this segmentation task, We prepared two datasets for supervised (CNN) and unsupervised (cycle-GAN) models. The dataset for the supervised model is obtained from 88 raw SEM images showing arrays of nanostructures with various sizes and shapes. The SEM images were cropped to show individual nanostructures, and the labels (segmented images) were generated manually using the Ilastik tool. We applied random flip (horizontal and vertical), rotation (10-degree steps), and zoom factor to augment the dataset.

For the unsupervised model, the SEM images were collected from a publicly available dataset (NFFA-EUROPE) produced at CNR-IOM. A total of roughly 18,577 SEM images at the nanoscales are classified into ten categories. [1] We used images from the class "Patterned surface" and "MEMES Device" as those two categories resemble the synthetic labels we generated. The images were cropped to remove scale bars to avoid training error. The synthetic labels were generated from a semiconductor lithography design program called PhiDL. Gaussian noises were introduced on the synthetic labels to help the convergence of the training. All images in the datasets are reshaped to a size of 256x256. The training and test examples were divided with an 80:20 ratio.

4 Methods

In this project, we conducted supervised learning and unsupervised learning methods to predict the contours from the SEM images. For the supervised learning method, we used the ResNet34 [3] (Figure 2), which combines 34 convolution layers with residual blocks. The residual block allows for gaining accuracy from considerable network depth. The model was trained with a cross-entropy loss function and Adam optimizer with L2 regularization (weight decay) of 0.01. we also customized a function to measure the pixel-wise accuracy. We used a pre-trained model for several reasons. First, we have a relatively small dataset; training a heavy network from the ground with a small dataset would cause a high bias issue. Second, the SEM images we have involves common features like lines and curves, which serve as lower level features and have already been learned by the model. Third, using a pre-trained model would allow the learning to converge faster.

The Cycle-GAN (Figure 3) was proposed as an unsupervised learning method; in other words, it does not require the one-to-one relationship between the SEM images and contours. [4] However, to achieve this goal, it involves two generators (U-net generator) and two discriminators (Conv- net) in training. The first generator (G) will map the SEM image (X) to the contour domain (Y), and the

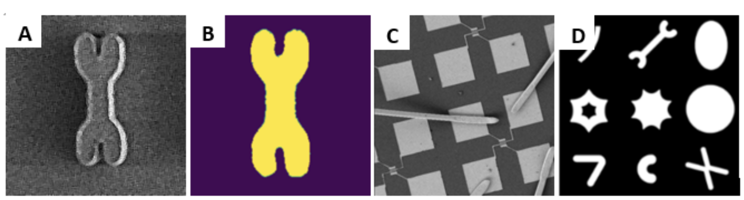

Figure 1: (A) Sample preprocessed SEM images and (B) corresponding manualy generated labels using ilastik. (C) SEM image from NFFA-EUROPE dataset, and (D) Synthetic label generated using PhIDL

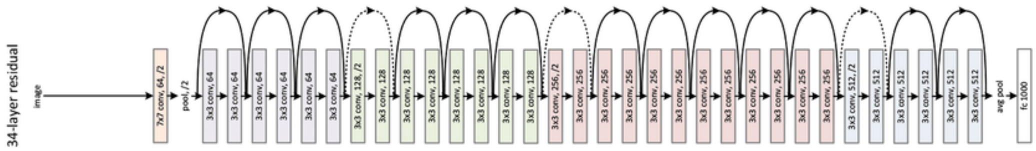

Figure 2: ResNet-34 structure. The network consists of multiple convolution layers with a skip connection of every two layers.

second generator (F) will be responsible for mapping a contour image (Y) back to an SEM image (X). Since the training data is unpaired, two discriminators (D_G and D_F) will be used to regularize each generator. There are a couple of loss functions that need to be calculated in the process. In theory, after x going through a full circle, it should be mapped to itself: $F(G(x)) = x$, same for y: $G(F(y)) = y$. Therefore a cycle consistency loss measures were defined to quantify how pixel-wise information was predicted by the generators:

$$L_{cyc}(G, F) = E_{x\sim p(x)}[\|F(G(x)) - x\|] + E_{y\sim p(y)}[\|G(F(y)) - y\|].$$

The discriminators force the output of the image by generators to fall in the correct distribution. Therefore a cross-entropy loss was applied:

$$L_{GAN}(G, D_Y, x, y) = E_{y\sim p(y)}[\log D_Y(y)] + E_{x\sim p(x)}[\log(1 - D_Y(G(x)))]$$

The full objective is:

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F),$$

where λ weights the importance of the two types of loss.

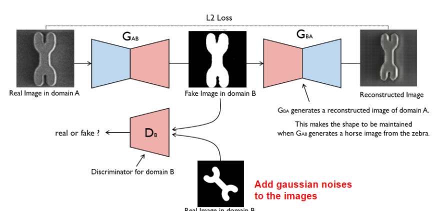

Figure 3: Cycle-GAN structure. Domain A is the SEM image domain, and domain B is the contour image domain. Generator G_{AB} will predict the contour for an SEM image while generator G_{BA} will predict the SEM image of a given contour. Discriminator D_A (not shown) and D_B will judge whether the predictions are a real SEM image (or correspondingly a real contour image). To differentiate from supervised learning, we used synthetic contour images for training, results in section 5.3.

5 Experiments/Results/Discussion

5.1 Canny Edge detection algorithm

We first implemented a traditional image processing edge detector, namely, a Canny edge detector as a baseline model. [6] The canny edge detector is a multi-step algorithm that can detect edges with noise suppressed, which is suitable for this task. The thresholds and sigma were set at [0.1 0.5] and 4, respectively, for this dataset. As shown in the Fig.4A, the outline of the structure can be detected with a clean background. However, some edges were being double reported due to the incident beam direction. It's noteworthy that the limitation of the Canny edge detector is the tuning of parameters. The parameters that worked for this dataset may not work well with another set of SEM images with different background noise levels, thus making it hard to find an optimal set of parameters that works in various cases.

5.2 ResNet model

We then used a more generalized algorithm to avoid manual parameter tuning. Specifically, we used ResNet, a supervised learning model, with excellent generalization performance on recognition tasks. We used the Fastai [7] package to train ResNet34 with our data on Google Colab. We used the first dataset with paired customized SEM images and manual labels. Using the ResNet model described in the methods section, we tune the two significant learning hyperparameters, rate, and batch size. The summary of the loss and accuracy of the four learning rates chosen are listed in Table 1. In this study, with the optimal learning rate found at 1e-3, we found the batch size to be 4. Fig 4B shows the predicted results on the right column. The model can correctly recognize the shape contour, but it also labeled some background noise in the first example.

Table 1: Training loss, validation loss and accuracy with respect to learning rate

	lr = 1e-2	lr = 1e-3	lr = 4e-4	lr = 1e-4
Training loss	0.285253	0.048427	0.053615	0.071367
Validation loss	0.188275	0.063001	0.064227	0.089752
Accuracy	0.977639	0.959030	0.963425	0.970521

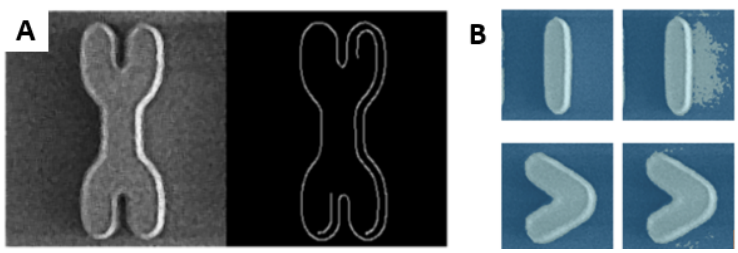

Figure 4: Canny edge detection and Resnet-34 results. (A) SEM image and corresponding results with traditional canny detection method applied. (B) The ResNet comparison between ground truth (left column) and predicted results (right column).

Table 2: Summary of cycleGAN settings and results

	Domain A	Domain B	Loss	Epoch	Accuracy
1	single nanostructures	Manual labels	BCE	100	0.892
2	single nanostructures	Synthetic labels	BCE	100	not converged
3	single nanostructures	Synthetic labels	MSE	200	N/A
4	NFFA datasets	Manual labels + Synthetic labels	MSE	200	N/A

5.3 CycleGAN model

Although ResNet showed decent results, the training requires a paired image and label. It's also challenging to enlarge the dataset due to the labor-intensive label generation. Therefore, we further implemented a CycleGAN model to uncoupled the one-to-one relationship between SEM images with synthetic labels. In the cycleGAN networks, the generator and discriminator we used is based on the Pix2Pix paper. [8] We used a U-net structure with skip connections for the generator and a PatchGAN structure for the discriminator. The learning rate is 2e-4 for generator and 1e-4 for discriminator, respectively. We started the training on the same dataset with ResNets to compare the supervised and unsupervised results. Fig.5 A and B showed the input SEM image and corresponding segmented image. The nanostructure was marked. Quantitatively, We use the same pixel-wise accuracy to evaluate the segmented images, and the model achieved 89.2 percent accuracy (Table 2). Fig. 5C and D also showed the back-transformed image pairs by feeding generator F with manual labels. The predicted image did show SEM signatures with shadowing effect and texture. The results indicate that the model can reach convergence in 100 epochs, which is shown by the plot shown in Fig. 6.

We then moved on to the dataset with unpaired images by using synthetic labels rather than manual labels. The synthetic labels are design files of different nanostructures. Initially, we found that

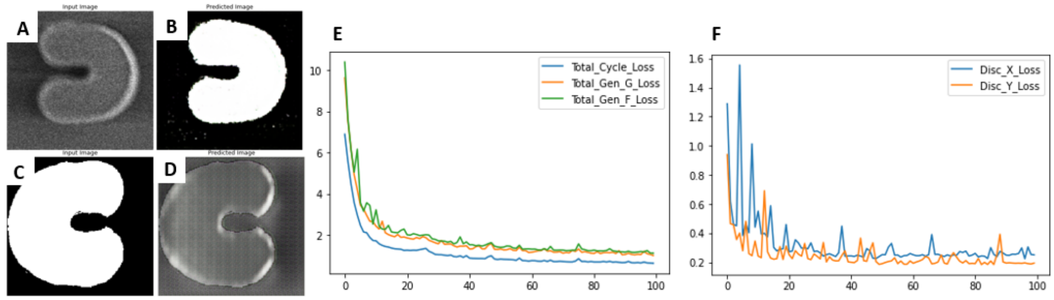

Figure 5: Results obtained from cycleGAN model using paired SEM images and manual labels (A) Input SEM image (B) predicted label image (C) Input manual label (D)Predicted SEM image (E, F) Plot of total cycle loss, generator-G loss, generator-F loss, discriminator-X loss and discriminator-Y loss with respect to number of epochs trained

the model couldn't converge in 100 epochs as the previous case. After performing several control experiments to evaluate the role of the loss function and input data, we identified an optimized condition where mean squared loss and addition of Gaussian noise on synthetic labels helped to stabilize the training. Due to the lack of paired image and label sets, we couldn't calculate the accuracy of prediction. Fig.6A and B show the input SEM and predicted label, which qualitatively resembles baseline models, proving that cycleGAN also works for unpaired image sets. We also used an expanded dataset (NFFA-EUROPE) to test the performance further. Fig 6E and F show the input SEM of a MEMES device and the predicted segmented label. The majority of the contour is correctly recognized. We also noticed that some artifacts happened at the center of the prediction image. This artifact could be attributed to the synthetic label's distribution where most of the pattern is located at the center of the image.

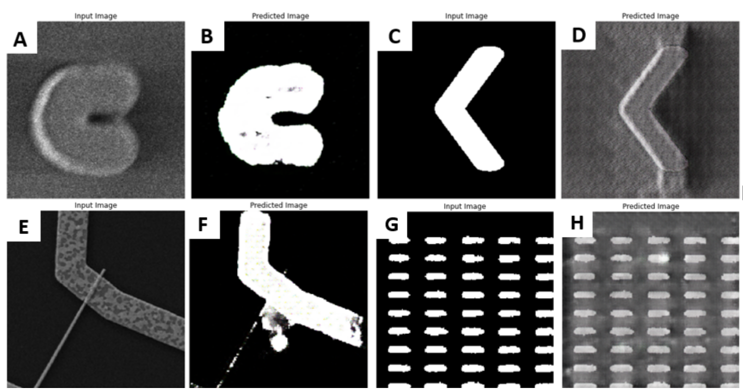

Figure 6: Results obtained from cycleGAN model using (top row) unpaired SEM images and synthetic label, (bottom row) NFFA dataset with synthetic labels (A, E) Input SEM image, and (B, F) predicted label image (C, G) Input synthetic label (D, H) Predicted SEM image.

6 Conclusion/Future Work

We demonstrated that cycleGAN could perform competitively as a traditional Canny edge detector or ResNet in segmenting SEM images but doesn't require manual parameter tuning or paired datasets. However, several future directions still could be worked on for better image conversion. First, the model has a lower performance on some scattering SEM images. We could add random brightening and darkening in the preprocessing step to perform a better job on SEM images with scattering. Second, the checkboard artifact was identified in the synthetic SEM images. This artifact comes from the upsampling during the deconvolution step and could be improved by better upsampling methods like NN-Resize Convolution. Moreover, efforts include enlarging the dataset of synthetic labels with more nanostructures that are also required to apply this algorithm in more generalized settings.

7 Contributions

Ching-Ting Tsai prepared the SEM images for single nanostructures, labeled images with ilastik, created synthetic shapes from PHIDL, and implemented the ResNet from the fastAI source code. Xiaojun Sun implemented canny edge detection and prepared SEM images from NFFA-Europe datasets. Xiaohan Li adapted the CycleGAN from the tensorflow source code. Csaba Forró helped design the experiments and analyzed the results. All of the authors helped preprocessing the data, model training, and report writing.

References

[1] M. Modarres, R. Aversa, S. Cozzini, R. Ciancio, A. Leto G. Brandino. (2017) Neural Network for Nanoscience Scanning Electron Microscope Image Recognition. *Scientific Reports* **7**, 13282.

[2] Y. LeCun, Y. Bengio G. Hinton. (2015) Deep Learning. *Nature* **521**, 436-444.

[3] K. He, X. Zhang, S. Ren J. Sun. (2016) Deep Residual Learning for Image Recognition. *CVPR*, 770-778.

[4] Y. Zhu, T. Park., P. Isola A. Efros. (2018) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CVPR*, 2242-2251.

[5] S. Ihle, A. Reichmuth, S. Girardin, H. Han, F. Stauffer, A. Bonnin, M. Stampanoni, K. Pattisapu, J. Vörös C. Forró. (2019) Unsupervised Data to Content Transformation with Histogram-Matching Cycle-Consistent Generative Adversarial Networks. *Nature Machine Learning* **1**, 461-470.

[6] J. Canny. (1986) A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell* **8**, 679-698.

[7] J. Howard, S. Gugger. (2020) fastai: Layered API for Deep Learning (2020) *arXiv* **11**, 108. (url: https://github.com/fastai/fastai)

[8] P. Isola, J. Zhu, T. Zhao, A. Efros. (2017) Image-to-image Translation with Conditional Adversarial Networks. **8**, 1125-1134.