Original Article

Determining the Composition of a Mixed Material with Synthetic Data

Cuong Ly[1,2]*, Cody A. Nizinski[3], Ada Toydemir[4], Clement Vachet[2], Luther W. McDonald IV[3] and Tolga Tasdizen[1,2]

[1]Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT 84112, USA; [2]Scientific Computing and Imaging Institute, Salt Lake City, UT 84112, USA; [3]Nuclear Engineering Program, Department of Civil and Environmental Engineering, University of Utah, Salt Lake City, UT 84112, USA and [4]Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA

Abstract

Determining the composition of a mixed material is an open problem that has attracted the interest of researchers in many fields. In our recent work, we proposed a novel approach to determine the composition of a mixed material using convolutional neural networks (CNNs). In machine learning, a model "learns" a specific task for which it is designed through data. Hence, obtaining a dataset of mixed materials is required to develop CNNs for the task of estimating the composition. However, the proposed method instead creates the synthetic data of mixed materials generated from using only images of pure materials present in those mixtures. Thus, it eliminates the prohibitive cost and tedious process of collecting images of mixed materials. The motivation for this study is to provide mathematical details of the proposed approach in addition to extensive experiments and analyses. We examine the approach on two datasets to demonstrate the ease of extending the proposed approach to any mixtures. We perform experiments to demonstrate that the proposed approach can accurately determine the presence of the materials, and sufficiently estimate the precise composition of a mixed material. Moreover, we provide analyses to strengthen the validation and benefits of the proposed approach.

Key words: composition estimation, convolutional neural networks, machine learning, scanning electron microscopy, synthetic data, uranium oxides

(Received 14 May 2021; revised 5 August 2021; accepted 11 September 2021)

Introduction

In recent years, convolutional neural networks (CNNs) have achieved astonishing performance in the tasks of object classification, segmentation, and object detection (Girshick, 2015; Ronneberger et al., 2015; Simonyan & Zisserman, 2015; He et al., 2016; Chen et al., 2018). A CNN is a type of machine learning model that consists of a series of convolution layers [as well as other layers in modern architectures, such as pooling layers and/or batch normalization (BN) layers (Ioffe & Szegedy, 2015)]. A convolution layer contains a set of learnable filters with a specified kernel size. Each learnable filter iterates through an input and computes an inner product between the filter and the region of the input overlapping with it at each iteration.

A CNN "learns" to select a meaningful set of filters through data that enables it to be used in diverse sets of problems. Thus, we have witnessed a large volume of works in the field of materials science utilizing CNNs to tackle various aspects of the field. For example, Kaufmann et al. (2020) adopted a well-known object classification CNN model, called Xception, to determine the phase of diffraction patterns of crystalline materials captured in electron backscatter diffraction (EBSD). Matson et al. (2019) utilized CNNs to categorize structures of carbon nanotubes and nanofibers. Similarly, Hanson et al. (2019) and Heffernan et al. (2019) also used CNNs for the task of categorizing the characteristics of materials. In addition to using CNNs for the task of classification, CNNs have also been deployed for other tasks such as estimating optimal operational parameters (such as the focus setting) during the image acquisition of scanning electron microscopy (SEM) images (Yang et al., 2020), segmenting structures characterized in SEM images (Ly et al., 2019; Pazdernik et al., 2020), denoising the drifted microscopic images (Vasudevan & Jesse, 2019), and reconstructing sparse SEM images (Trampert et al., 2019).

Moreover, CNNs have also performed impressively in the image synthesis task thanks to the seminal work of Goodfellow et al. (2014), who proposed a new approach, called generative adversarial training. More specifically, they used two models competing against each other, in which one model tries to generate realistic samples whereas the other simultaneously seeks to distinguish the synthetic samples from the real samples. They were able to generate perceptually realistic images using two CNNs via adversarial training, which is referred to as generative adversarial networks (GANs). Since then, many works have proposed to further improve image quality, the complexity of the type of images being generated, the diversity of generated images, etc. (Arjovsky

*Corresponding author: Cuong Ly, E-mail: nly8292@sci.utah.edu

et al., 2017; Isola et al., 2017; Odena et al., 2017; Karras et al., 2018, 2019; Brock et al., 2019). Concurrently, another area of image synthesis, called neural style transfer, has seen rapid advancement (Gatys et al., 2016; Huang & Belongie, 2017; Li et al., 2017). Neural style transfer is the process of representing the semantic content of an image in different styles, for example, an image is represented under various seasons or times of day. Similarly, these image synthesis approaches have inspired many works in material sciences. For instance, GANs were utilized to synthesize microstructures of alloys (Singh et al., 2018; Iyer et al., 2019). Meanwhile, motivated by the style transfer model, Ma et al. (2020) proposed a model to transform the style of simulated labels from the Potts model to be similar to how they would have appeared had they been captured by a microscope.

Determining the composition of a mixed material is of interest in many fields (Sarkar et al., 2009; Samad et al., 2014; Rossen & Scrivener, 2017; Heffernan et al., 2019). For instance, composite metal oxides show tremendous benefits in absorption, separation, and photosensitive operations over single metal oxides in catalytic and electrocatalytic processes. When using mixed oxides, knowing the portion of each oxide in a mixture is essential in understanding the electro- and physico-chemical properties (Samad et al., 2014). Meanwhile, in India, measuring the percentage of uranium in a mixture of thorium–uranium mixed oxide is one of the required steps in quality assurance of nuclear fuels (Sarkar et al., 2009). On the other hand, determining the composition of calcium aluminum silicate hydrate (C–A–S–H) in a cement paste is part of the study of phase assemblages (Rossen & Scrivener, 2017). Various elemental analysis tools, such as powder X-ray diffraction (pXRD), SEM coupled with energy-dispersive spectroscopy (EDS), and laser-induced breakdown spectroscopy (LIBS) have been employed in these studies. Diverging from the elemental analysis methods and taking full advantage of the powerful performance of CNN, we proposed a novel approach to estimate the composition of mixed materials characterized in the SEM images in our previous work (Ly et al., 2021). Our proposed method deployed two CNN models. The first CNN is tasked with generating SEM images of mixtures from images of the pure materials appearing in the mixtures. The synthesized images are then used to train a second CNN model that is used to estimate the composition of a given input image. The main advantage of this proposed approach is that it does not require SEM images of the mixtures; thereby, it eliminates the monetary cost and laborious process of preparing and imaging samples of the mixtures (Ly et al., 2021). This advantage is further amplified when more materials are involved in the mixtures.

In the present study, we derive the mathematical details of the proposed approach in Ly et al. (2021), and present extensive experiments and analyses for further validation. Specifically, we validate the approach on two types of mixtures (binary and tertiary): (a) mixtures of triuranium octoxide (U_3O_8) synthesized from ammonium diuranate (ADU) and uranyl peroxide (UO_4); and (b) mixtures of U_3O_8 synthesized from ADU, uranyl hydroxide (UH), and sodium diuranate (SDU). The image synthesis process of these two sets of mixtures is the same, which emphasizes the ease of extending the proposed image synthesis model to many other mixtures. Furthermore, we implemented two variants for the mixture estimation model that are tasked with (a) determining the presence of materials and (b) estimating the precise composition of a given SEM image. From these experiments, the proposed approach in Ly et al. (2021) can reliably determine the materials present in a mixture characterized in an SEM image

(with the area under the ROC curve >0.9). Moreover, this approach can also provide estimated compositions in agreement with the actual compositions.

Materials and Methods

Mixtures of Uranium Oxides

Two sets of uranium oxides mixtures were used in this study: (a) mixtures of U_3O_8 synthesized from ADU and UO_4 and (b) mixtures of U_3O_8 synthesized from ADU, UH, and SDU. We abbreviate these sets of mixtures as ADU–UO_4 and ADU–UH–SDU, respectively, for the rest of the paper. For mixtures of ADU–UO_4, we utilized images from Heffernan et al. (2019). These images were acquired at a resolution of $1,024 \times 884$ with a horizontal field width (HFW) of $5.11\,\mu m$, which represents the scale across the width of the image. The details of how the ADU–UO_4 mixtures were prepared and imaged can be found in Sections 2.1 and 2.2 in Heffernan et al. (2019).

For ADU–UH–SDU mixtures, we utilized images from Schwerdt et al. (2019) as well as prepared and imaged samples. Specifically, we utilized images of pure materials (i.e., images of 100% ADU, 100% UH, and 100% SDU) from Schwerdt et al. (2019). The images of 100% ADU were acquired at a resolution of $1,024 \times 884$ with a HFW of 1.53 and $3.06\,\mu m$, whereas the images of 100% UH and 100% SDU were acquired at the same resolution but with a HFW of 3.06 and $6.13\,\mu m$, respectively. The images of mixtures of ADU–UH–SDU were acquired by first preparing the mixtures with U_3O_8 samples that were previously described individually by Schwerdt et al. (2019). The samples were stored under vacuum and at room temperature between their initial synthesis and mixing. Three binary U_3O_8 mixtures were prepared: ADU with UH, ADU with SDU, and UH with SDU. A tertiary mixture was prepared of U_3O_8 from ADU, UH, and SDU. Each mixture was prepared by aliquoting approximately 40 mg of each U_3O_8 component into a small PTFE vial containing a Teflon-coated stir bar, followed by 15 min of agitation with a Vortex mixer at the medium intensity setting as described by Heffernan et al. (2019). Table 1 lists the measured mass and weight% of each mixture.

Samples were prepared for analysis by SEM by dusting approximately 5–10 mg of mixed sample powder onto conductive double-sided carbon tape and aluminum pin stub mounts. An FEI Nova NanoSEM 630 scanning electron microscope was used to image the samples in immersion mode with the through-lens detector (TLD). Acquisitions were made at an image resolution of $1,024 \times 884$ at a HFW of $6.13\,\mu m$. Moreover, all the images were acquired with a high-voltage (HV) field setting of $7.00\,kV$ with the exception of the images of the ADU–SDU mixture,

Table 1. The Measured Mass (in mg) and Corresponding Percentage Weight of Each Pure Material in the ADU–UH–SDU Mixtures.

Mixture	Mass (Weight%)		
	ADU	UH	SDU
ADU–UH	42.6 (50.4%)	42.0 (49.6%)	
ADU–SDU	43.2 (51.0%)	–	41.5 (49.0%)
UH–SDU	–	42.2 (50.1%)	42.1 (49.9%)
ADU–UH–SDU	41.4 (33.6%)	41.1 (33.4%)	40.6 (33.0%)

which were acquired at 5.00 kV. Each sample was imaged without sputter coating, except for the ADU–SDU mixture, which showed signs of charging during SEM analysis; this sample was sputter coated with 20.0 ± 0.1 nm of Au/Pd film with a Gatan 682 Precision Etching and Coating System (PECS).

Image Synthesis Model

Synthesizing Mixed Samples

The proposed image synthesis model is based on the texture synthesis work in Gatys et al. (2017). They proposed to control the spatial location of a specific reference texture appearing in the generated image by minimizing the difference between the Gram matrices of the reference texture image and the generated image only in that specific region. Multiple regions with multiple reference textures can be easily synthesized at once by summing up that difference. In the present work, there is no constraint on where the reference textures should be located in the generated image. Thus, we instead minimized the difference between the Gram matrices of the generated images and the weighted sum of the Gram matrices of the reference textures. Formally speaking, to generate a new image, $\mathbf{x}_G$, from a given set of desired reference texture images $T = \{\mathbf{x}_{T_1}, \mathbf{x}_{T_2}, \ldots, \mathbf{x}_{T_n}\}$, the following objective function is optimized:

$$\mathcal{L} = \sum_{l \in L} \frac{1}{4C_l^2} \left\| \mathbf{G}^l(\mathbf{x}_G) - \sum_{k=T_1}^{T_n} \omega_k \mathbf{G}^l(\mathbf{x}_k) \right\|_2^2, \quad (1)$$

where L is a set of extracted features from a pre-trained CNN model (refer to Section A.1.1 for more detail), and $\mathbf{G}^l(\mathbf{x})$ is the Gram matrix at layer l representing the normalized correlation of the vectorized feature maps, $\mathbf{F}^l(\mathbf{x}) \in \mathcal{R}^{C^l \times N^l(\mathbf{x})}$ with C^l the number of channels, and $N^l(\mathbf{x})$ the product of the spatial dimension, $H^l \times W^l$:

$$\mathbf{G}^l(\mathbf{x}) = \frac{1}{N^l(\mathbf{x})} \mathbf{F}^l(\mathbf{x})^\top \mathbf{F}^l(\mathbf{x}), \quad (2)$$

ω_k is a scalar that dictates the influence of texture k on the generated image. Hence, by controlling ω_k, we can condition a certain percentage of a texture k to appear on the synthesized image.

The proposed image synthesis model adopts equation (1) to synthesize images of mixed material. To achieve this objective, we first defined each pure material in a desired mixture as a texture. Each image in the set of reference texture images used as input for each synthesis process is an image of the pure material present in a desired mixture. By conditioning ω_k, a new image of a mixture can easily be synthesized with the desired percentage of each pure material appearing in the mixture. In other words, the percentage of a specific pure material occupying the synthesized image corresponds to ω_k. Furthermore, we added the total variation (TV; Chambolle, 2004) objective function to increase the smoothness of the generated images. The final objective function used in the present study is

$$\mathcal{L} = \alpha \sum_{l \in L} \frac{1}{4C_l^2} \left\| \mathbf{G}^l(\mathbf{x}_G) - \sum_{k=T_1}^{T_n} \omega_k \mathbf{G}^l(\mathbf{x}_k) \right\|_2^2 + \gamma \mathcal{L}_{\text{TV}}, \quad (3)$$

where α and γ are adjustable weights to control the influence of each objective function on the overall function.

Pyramid Optimization

Here, we present the details of the proposed pyramid optimization strategy that speeds up the process of generating an image of size 512×512 by more than 50% compared to that of the optimization strategy used in Gatys et al. (2015, 2017); consequently, a large amount of data can be generated in a much more efficient manner.

The optimization strategy used in Gatys et al. (2015, 2017) initializes a generated image with white noise sampled from a uniform distribution $\sim \mathcal{U}(0, 1)$ and optimizes for a certain number of iterations. To speed up the optimization process, we used the motivation presented in the progressively growing GANs work (Karras et al., 2018). In that work, Karras et al. (2018) discovered that generating a large-scale structure at a smaller resolution first and then focusing on the fine detail at a larger resolution reduces training time. Taking advantage of that observation, we first initialized the generated image with white noise at a lower resolution. We then optimized the low-resolution generated image for a certain number of iterations. Next, we upsampled the generated image to twice its current size and optimized it further. This process is repeated until the final resolution of the desired generated image is reached. Hence, we refer to this optimization strategy as pyramid optimization. Furthermore, different to Karras et al. (2018), the proposed pyramid optimization does not add more layers as the spatial resolution increases.

Another advantage of using the pyramid optimization strategy is its ability to capture large structures in an image. For each filter in a convolution layer, a kernel with a fixed size is chosen to iterate through a given input. The fixed size limits how large of a region the output of that convolution layer represents. By using the proposed pyramid optimization, we essentially reduce the spatial dimension of the input (at the first few levels) to the convolution layers, while maintaining the kernel sizes; thereby, we ultimately enlarge the region the outputs of the convolution layers represent.

Figure 1 provides the progression details of the proposed pyramid optimization. Each row in that figure represents a specific resolution. In this study, we used three scales for the pyramid optimization strategy. In other words, we started the pyramid optimization process with white noise input at a resolution 4× smaller than the resolution of the final output. We optimized that input for K_1 iterations. After K_1 iterations, we upsampled the generated images to twice its current size and optimized for additional K_2 iterations. We again upsampled the generated images to twice its current size and optimized for another K_3 iterations to obtain the desired image. The values of K_1, K_2, and K_3 were empirically determined and correspond to 10,000, 10,000, and 1,000, respectively.

Mixture Estimation Model

For the mixture estimation model, we implemented two variants for two separate tasks. The complete architecture of these two models can be found in Section A.2.1. Both variants have a similar architecture except in the last layer, and the objective function due to the task for which each variant was designed. In the first variant, we implemented the model to predict the presence of materials in a given image. In other words, for example, this variant is used to determine if ADU or UO_4 or both are present in a given SEM of an ADU–UO_4 mixture. We refer to the first variant as MEM-A. The second variant is tasked with estimating the exact composition of a given image, and is referred to as MEM-B.
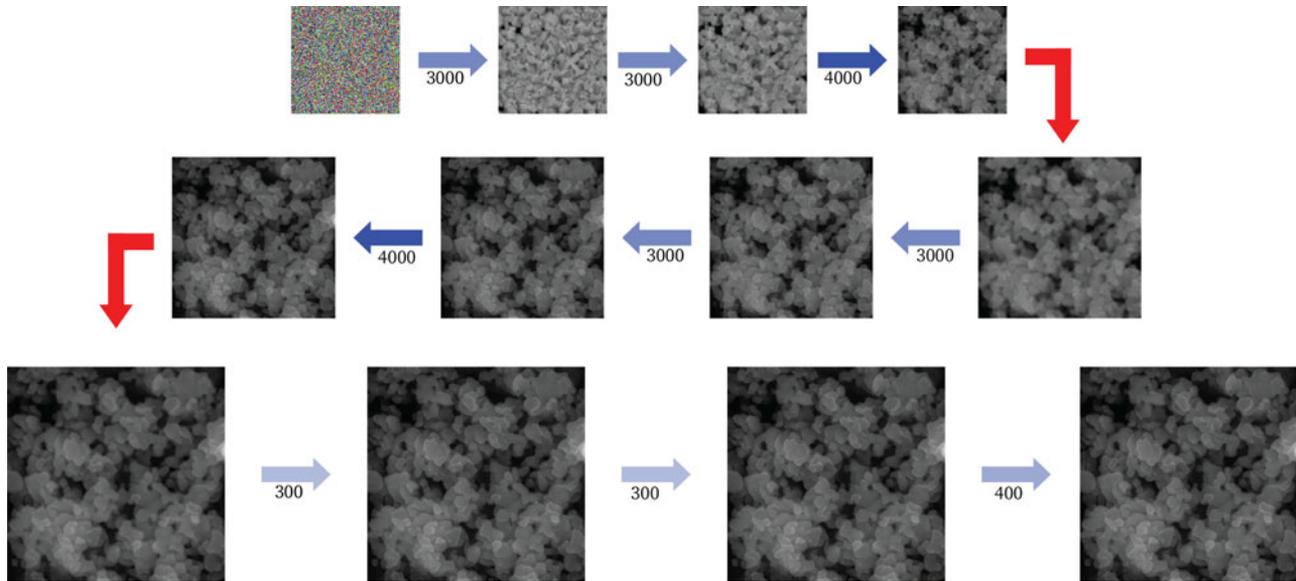
Fig. 1. The proposed pyramid optimization strategy. The top left image shows the starting point (random initialization), whereas the bottom right image shows the output image at the end of the optimization process. Each row in the figure represents a different resolution used in the optimization process. Lastly, the red arrows in the figure represent the bilinear upsampling process, and the blue arrows represent the number of iterations (indicated underneath each arrow) done to update the generated images using equation (3).

Results and Discussion

Image Synthesis Model

We used the proposed image synthesis model to generate images of two sets of mixtures: ADU–UO$_4$ and ADU–UH–SDU. The images of ADU–UO$_4$ were generated at a resolution of 512×512, whereas the images of ADU–UH–SDU were generated at a resolution of 128×128 to account for the difference in scale between input images used for the image synthesis model. The scale correction process is detailed in Section A.1.2.

For each mixture, we generated two sets of images. In the first set, we manually selected the weights, ω, in equation (3). We named this set dataset A. The purpose of generating dataset A is to have a similar set of compositions as real images. We also generated images containing various compositions by randomly sampling ω for each synthesized sample, and we refer to this set as dataset B. For this dataset, we generated twice the amount of images compared with dataset A to be able to sample all the possible compositions and obtain more than one image per composition. Dataset B is constructed to accurately represent the real-world scenario in which we would like to have an approach that is able to estimate all possible compositions. Table 2 details the number of synthesized samples for each dataset as well as the number of real images of ADU–UO$_4$ and ADU–UH–SDU mixtures.

Figures 2 and 3 show a side-by-side comparison between a few representative samples of real and synthesized images. As seen in these two figures, the synthesized images are qualitatively similar to real images. For instance, in Figure 2, one noticeable characteristic of ADU–UO$_4$ mixtures is the correlation between the size of particles and the percentage of UO$_4$ in the images. With the higher percentage of UO$_4$ in a given image is the more larger particles appearing in the images. This particular characteristic can be easily identified in both real and synthesized images. For ADU–UH–SDU mixtures, the morphological characteristics of each individual material are distinguishably different from each other. The particles of SDU have a rough surface and are granular,

Table 2. The Number of Images Available in Each Dataset for ADU–UO$_4$ and ADU–UH–SDU Mixtures.

Dataset	Number of Images	
	ADU–UO$_4$	ADU–UH–SDU
Real images	1,548	1,906
Dataset A	2,500	2,800
Dataset B	5,000	5,600

whereas the particles of ADU are more rounded and smooth. On the other hand, the particles of UH are much larger in size compared with that of ADU or SDU and have smooth plate-like structures. These characteristics are clearly visible in both real and synthesized images. For instance, the large plate-like structures can be located in both real and synthesized images of 100% UH (second row in Fig. 3). Moreover, both plate-like and smaller rounded particles are found in both real and synthesized images of 50% ADU–50% UH mixture (fourth row in Fig. 3).

Mixture Estimation Model

Identifying Materials

The main purpose of this experiment is to demonstrate that the synthesized images can be used to train a model used for determining the materials present in a given SEM image. We trained the mixture estimation model, MEM-A, using only synthesized images and then tested the trained model on only real samples. In addition, we also trained MEM-A on only real images and tested this trained model on the same test set for performance comparison. Tables 3 and 4 show the area under the receiver operating characteristic curve (AUROC or AUC) of each material in the ADU–UO$_4$ and ADU–UH–SDU mixtures, respectively. We also reported the micro-average and macro-average. The micro-
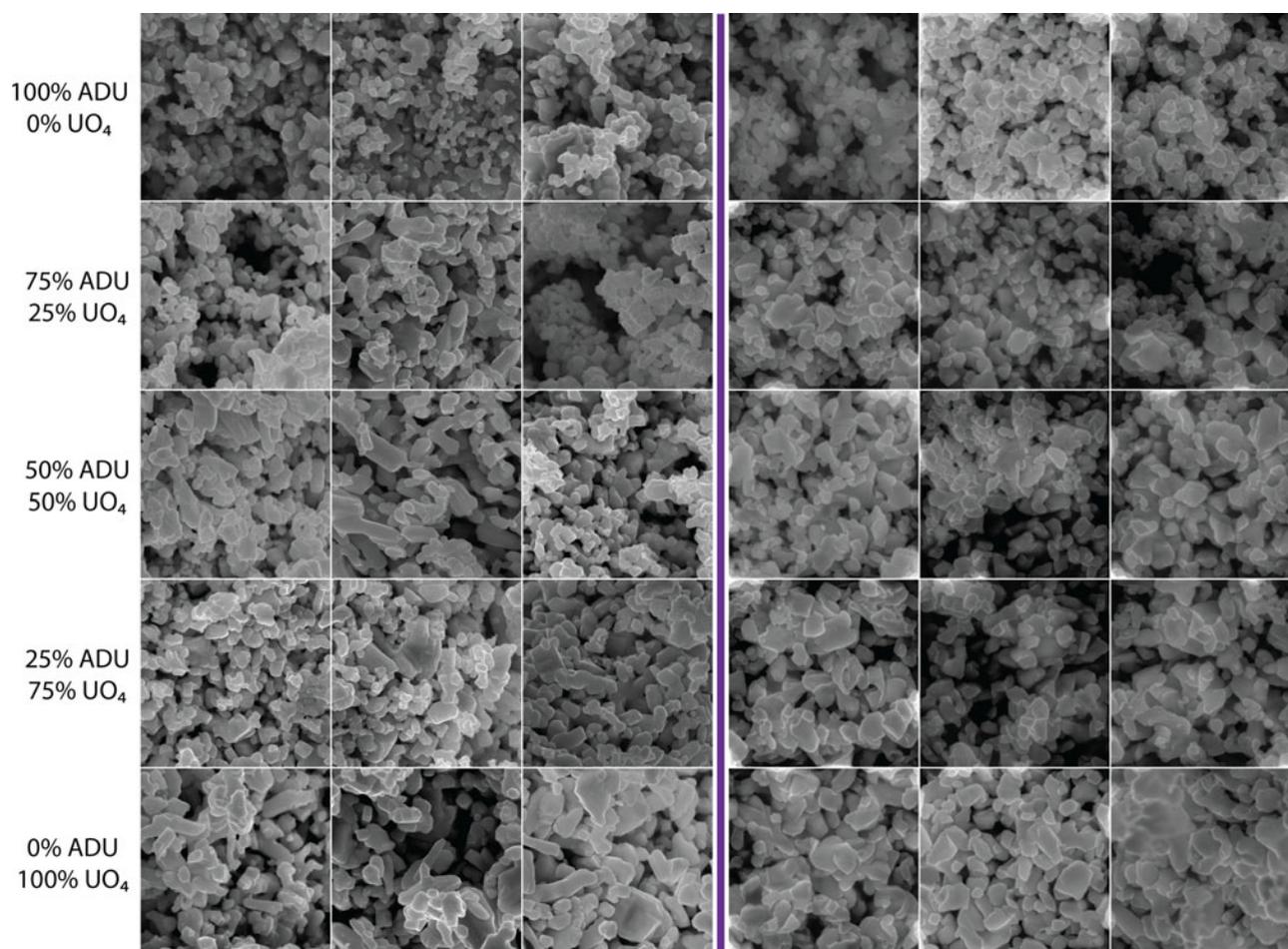
Fig. 2. The qualitative assessment of a few representative samples between real (the first three columns) and generated images (the last three columns). Each row in the figure represents a different ADU–UO$_4$ mixture.

average represents the weighted performance based on the frequency of each class. In other words, a class with more samples has more influence on the final result. In contrast, the macro-average treats each class equally. Refer to Section A.3 for the formal definition of the micro-average and macro-average of AUC as well as AUC itself.

As seen in the tables, the performance of the mixture estimation model MEM-A when trained with synthesized images achieved high AUC values (>0.9 in both the micro-average and macro-average) for both mixtures. Even though the AUC results of MEM-A trained with synthetic data are lower than when the model trained with real images, the high AUC values of the model trained with synthesized images implies that the model can still reliably identify the presence of pure materials in a mixture. Moreover, these results further validate that the synthesized images have similar characteristics to those of real images.

Composition Estimation
In this experiment, we used the second variant of the mixture estimation model, MEM-B, to estimate the composition of a given SEM image. The overall results of both mixtures are shown in Tables 5 and 6. As seen in both tables, the MEM-B model provides a reasonable estimate for both mixtures. Since this is a much more challenging task compared with materials prediction in the previous section, the results are not as accurate as for the previous task. However, the overall performance of the model trained with only synthesized images and only real images is still comparable in the ADU–UO$_4$ mixtures, as indicated by the coefficient of determination (R^2) and root-mean-square error (RMSE) metrics. For the ADU–UH–SDU mixtures, the performance of the model trained with only synthesized images is reasonable, but the gap in performance between the model trained with only real images and with only synthesized images is larger than the one observed in ADU–UO$_4$. The larger gap in performance in ADU–UH–SDU is attributed to the smaller resolution of the synthesized images and the larger number of materials involved in the mixtures.

Moreover, as expected, the model trained with dataset A outperformed the one trained with dataset B since the compositions in dataset A are tailored to the test set. However, it can be argued that the model trained with dataset B would perform better in practice because capturing a broader set of compositions would help eliminate bias on unseen compositions in the test set.

Analysis

Computation Time
One of our motivations is to provide an alternative approach that can be used to accurately determine materials in a mixture while eliminating the high cost and time-consuming process of sample
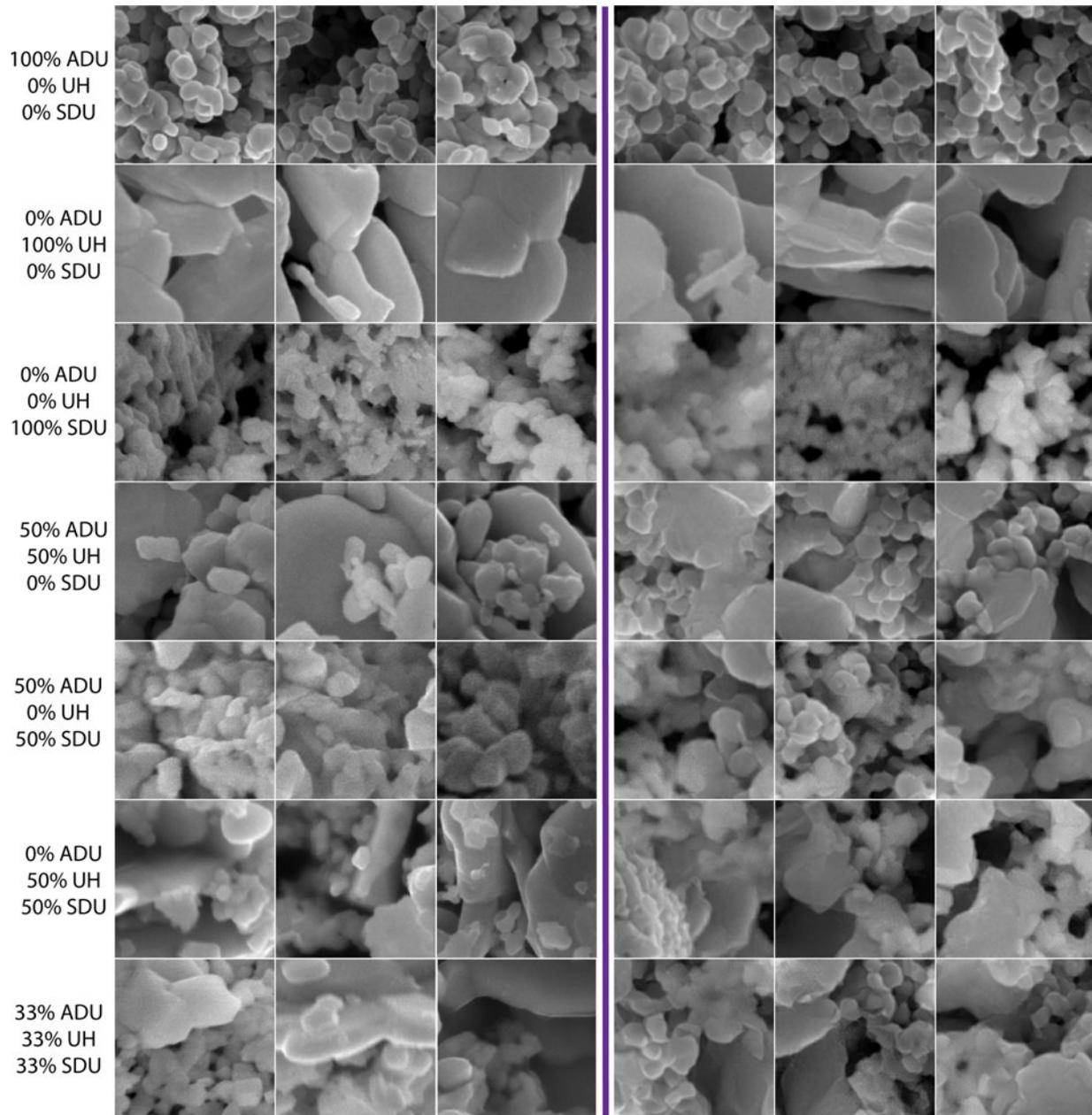
Fig. 3. The qualitative assessment of a few representative samples between real (the first three columns) and generated images (the last three columns). Each row in the figure represents a different ADU–UH–SDU mixture.

preparation and imaging by building a synthetic dataset. Thus, the computation time of an image synthesis model is one of the key criteria that justifies deploying the proposed approach. Table 7 lists the computation time for generating an image using the image synthesis model without and with our proposed pyramid

optimization scheme implemented using Pytorch library (Paszke et al., 2019) on a single Titan RTX graphics processing unit (GPU) hardware. As clearly seen from the table, the image synthesis model can synthesize an image effortlessly in a short period of time. The computation time of the image synthesis model further decreases with the pyramid optimization scheme. Furthermore, this computation time can be improved with model parallelization on multiple GPUs if resources are available.

Table 3. The AUC of MEM-A When Trained with Different Datasets in Determining the Presence of Materials in ADU–UO$_4$ Mixtures.

Training Dataset	ADU	UO$_4$	Micro-Avg	Macro-Avg
Real images	0.999	0.990	0.997	0.995
Dataset A	0.990	0.853	0.949	0.922

Diversity Analysis

In this analysis, we analyzed the diversity of synthetic images. The diversity measures the variation of images within a given class. A small variation indicates that the generated images look too similar to each other. Consequently, having generated images with

Table 4. The AUC of MEM-A When Trained with Different Datasets in Determining the Presence of Materials in ADU–UH–SDU Mixtures.

Training Dataset	ADU	UH	SDU	Micro-Avg	Macro-Avg
Real images	1.00	0.998	0.994	0.995	0.998
Dataset A	0.939	0.997	0.934	0.950	0.957

Table 5. The Mean Estimated Composition of ADU–UO_4 Mixtures.

Training Dataset	100% ADU / 0% UO_4	75% ADU / 25% UO_4	50% ADU / 50% UO_4	25% ADU / 75% UO_4	0% ADU / 100% UO_4	RMSE	R^2
Real images	0.865	0.687	0.531	0.303	0.107	0.0880	0.937
	0.135	0.313	0.469	0.697	0.893		
Dataset A	0.867	0.745	0.600	0.435	0.042	0.122	0.876
	0.133	0.255	0.400	0.565	0.958		
Dataset B	0.829	0.726	0.609	0.460	0.150	0.153	0.806
	0.171	0.274	0.391	0.540	0.850		

Table 6. The Mean Estimated Composition of ADU–UH–SDU Mixtures.

Training Dataset	100% ADU / 0% UH / 0% SDU	0% ADU / 100% UH / 0% SDU	0% ADU / 0% UH / 100% SDU	50% ADU / 50% UH / 0% SDU	50% ADU / 0% UH / 50% SDU	0% ADU / 50% UH / 50% SDU	33% ADU / 33% UH / 33% SDU	RMSE	R^2
Real images	0.975	0.0150	0.0500	0.393	0.551	0.104	0.312	0.109	0.890
	0.0200	0.980	0.0940	0.471	0.208	0.575	0.470		
	0.00500	0.00500	0.856	0.136	0.241	0.321	0.218		
Dataset A	0.925	0.0700	0.112	0.351	0.463	0.178	0.249	0.123	0.866
	0.0530	0.905	0.0970	0.327	0.0740	0.495	0.407		
	0.0220	0.0250	0.791	0.322	0.463	0.327	0.344		
Dataset B	0.703	0.135	0.151	0.269	0.309	0.165	0.212	0.199	0.662
	0.173	0.740	0.160	0.268	0.154	0.390	0.312		
	0.124	0.125	0.689	0.463	0.537	0.445	0.476		

less diversity means that the synthetic data fail to capture the underlying distribution of the dataset of interest. We used multi-scale structural similarity (MS-SSIM) (Wang et al., 2004) to measure the diversity in this study. The MS-SSIM of two given input images has a value between 0.0 and 1.0. The larger value indicates that the two images are much more similar to each other. For this analysis, we computed the MS-SSIM for each composition in the mixture. For each composition, we first computed the mean MS-SSIM of 7,000 randomly selected distinct pairs of images in that mixture. Then, the mean MS-SSIM across all compositions

Table 7. The Computation Time of the Proposed Image Synthesis Model without and with the Pyramid Optimization Scheme to Generate an Image of Size 512 × 512.

Optimization Scheme	Computation Time
Without pyramid optimization	~420 s
With pyramid optimization	~150 s

was evaluated. Tables 8 and 9 show the MS-SSIM of real images and synthetic images for each composition and for the entire dataset. As seen in Table 8, the synthetic images achieved a comparable MS-SSIM metric as the real images for ADU–UO_4 mixtures. Meanwhile, the difference in MS-SSIM metric between real images and synthetic images for ADU–UH–SDU is larger. We hypothesized this larger gap is due to the smaller resolution of synthesized images compared with the real images. However, the MS-SSIM metric of synthetic images for ADU–UH–SDU mixtures is still relatively small. Thus, we believe that the synthetic images of ADU–UH–SDU mixtures still reasonably capture the underlying distribution.

Conclusion

In this study, we demonstrated that the proposed approach in Ly et al. (2021) can be easily applied to many different mixtures. At the same time, the proposed approach provides an accurate prediction (>0.9% in AUC) of the presence of materials in a mixture

Table 8. The MS-SSIM of Real and Synthesized Images of ADU–UO$_4$ Mixtures.

Dataset	100% ADU 0% UO$_4$	75% ADU 25% UO$_4$	50% ADU 50% UO$_4$	25% ADU 75% UO$_4$	0% ADU 100% UO$_4$	Average MS-SSIM
Real images	0.0029	0.0041	0.0041	0.0039	0.0048	0.0040
Synthesized images	0.0054	0.0055	0.0056	0.0056	0.0052	0.0055

Table 9. The MS-SSIM of Real and Synthesized Images of ADU–UH–SDU Mixtures.

Dataset	100% ADU 0% UH 0% SDU	0% ADU 100% UH 0% SDU	0% ADU 0% UH 100% SDU	50% ADU 50% UH 0% SDU	50% ADU 0% UH 50% SDU	0% ADU 50% UH 50% SDU	33% ADU 33% UH 33% SDU	Average MS-SSIM
Real images	0.0046	0.0049	0.0046	0.0043	0.0012	0.0046	0.0078	0.0046
Synthesized images	0.0093	0.012	0.012	0.0096	0.010	0.011	0.011	0.011

characterized in an SEM image and a reasonable estimation of the composition. Furthermore, the proposed approach in Ly et al. (2021) achieves these accuracies relying solely on the synthetic data generated without requiring any images of mixed materials. This advantage eliminates the cumbersome process of sample preparation and imaging, which scales with the number of materials involved in the mixtures.

The proposed approach provides promising results for how generation of synthetic data can be beneficial in material science research. However, many challenges still remain that follow-up studies need to address. First and foremost, the performance of both mixture estimation models, MEM-A and MEM-B, when trained on synthetic data is still lagging behind the performance of models when trained with real images. This finding indicates that a gap between synthesized images and real images still exists. Thus, the next essential step is to address this gap by developing an image synthesis model that can generate much more realistic images.

Second, even though the mixture estimation model can estimate the compositions fairly well (when trained with either real or synthesized images), the need for a more accurate estimation is still of interest. This challenge can be tackled by developing new CNN architectures or learning methodologies to improve the estimation. For example, a semi-supervised learning method, combining a small number of real images of mixed materials along with a larger number of synthetic data, would have a potential of improving the overall performance.

References

Arjovsky M, Chintala S & Bottou L (2017). Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research (PMLR)*, vol. 70, Precup D & Teh YW (Eds.), pp. 214–223. Available at http://proceedings.mlr.press/v70/arjovsky17a.html.

Brock A, Donahue J & Simonyan K (2019). Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019*, New Orleans, LA, USA, May 6–9, 2019. Available at https://openreview.net/forum?id=B1xsqj09Fm.

Chambolle A (2004). An algorithm for total variation minimization and applications. *J Math Imaging Vis* **20**, 89–97. https://doi.org/10.1023/B:JMIV.0000011325.36760.1e

Chen L, Papandreou G, Kokkinos I, Murphy K & Yuille AL (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Pattern Anal Mach Intell* **40**, 834–848. https://doi.org/10.1109/TPAMI.2017.2699184

Gatys LA, Ecker AS & Bethge M (2015). Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, Montreal, Quebec, Canada, December 7–12, 2015, Cortes C, Lawrence ND, Lee DD, Sugiyama M & Garnett R (Eds.), pp. 262–270. Available at https://proceedings.neurips.cc/paper/2015/hash/a5e00132373a7031000fd987a3c9f87b-Abstract.html.

Gatys LA, Ecker AS & Bethge M (2016). Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, USA, June 27–30, 2016, pp. 2414–2423. IEEE Computer Society. Available at https://doi.org/10.1109/CVPR.2016.265.

Gatys LA, Ecker AS, Bethge M, Hertzmann A & Shechtman E (2017). Controlling perceptual factors in neural style transfer. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, July 21–26, 2017, pp. 3730–3738. IEEE Computer Society. Available at https://doi.org/10.1109/CVPR.2017.397.

Girshick RB (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, Santiago, Chile, December 7–13, 2015, pp. 1440–1448. IEEE Computer Society. Available at https://doi.org/10.1109/ICCV.2015.169.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A & Bengio Y (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, vol. 27, Ghahramani Z, Welling M, Cortes C, Lawrence N & Weinberger KQ (Eds.). Curran Associates, Inc. Available at https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Hanson AB, Lee RN, Vachet C, Schwerdt IJ, Tasdizen T & Mcdonald LW (2019). Quantifying impurity effects on the surface morphology of α-U$_3$O$_8$. *Anal Chem* **91**, 10081–10087.

He K, Zhang X, Ren S & Sun J (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, USA, June 27–30, 2016, pp. 770–778. IEEE Computer Society. Available at https://doi.org/10.1109/CVPR.2016.90.

Heffernan ST, Ly C, Mower BJ, Vachet C, Schwerdt IJ, Tasdizen T & Iv LWM (2019). Identifying surface morphological characteristics to differentiate between mixtures of U$_3$O$_8$ synthesized from ammonium diurante and uranyl peroxide. *Radiochim Acta* **108**, 29–36.

Huang X & Belongie SJ (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision, ICCV 2017*, Venice, Italy, October 22–29, 2017, pp. 1510–1519. IEEE Computer Society. Available at https://doi.org/10.1109/ICCV.2017.167.

Ioffe S & Szegedy C (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, Lille, France, 6–11 July, 2015, *JMLR Workshop and Conference Proceedings*, vol. 37, Bach FR & Blei DM (Eds.), pp. 448–456. Available at http://proceedings.mlr.press/v37/ioffe15.html.

Isola P, Zhu J, Zhou T & Efros AA (2017). Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, July 21–26, 2017, pp. 5967–5976. IEEE Computer Society. Available at https://doi.org/10.1109/CVPR.2017.632.

Iyer A, Dey B, Dasgupta A, Chen W & Chakraborty A (2019). A conditional generative model for predicting material microstructures from processing methods. *CoRR*. Available at http://arxiv.org/abs/1910.02133.

Karras T, Aila T, Laine S & Lehtinen J (2018). Progressive growing of GANs for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings. Available at https://openreview.net/forum?id=Hk99zCeAb.

Karras T, Laine S & Aila T (2019). A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, Long Beach, CA, USA, June 16–20, 2019, pp. 4401–4410. Computer Vision Foundation/IEEE.

Kaufmann K, Zhu C, Rosengarten AS, Maryanovsky D, Wang H & Vecchio KS (2020). Phase mapping in EBSD using convolutional neural networks. *Microsc Microanal* **26**, 458–468.

Li Y, Fang C, Yang J, Wang Z, Lu X & Yang M (2017). Universal style transfer via feature transforms. In Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN & Garnett R (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, Long Beach, CA, USA, December 4–9, 2017, pp. 386–396. Available at https://proceedings.neurips.cc/paper/2017/hash/49182f81e6a13cf5eaa496d51fea6406-Abstract.html.

Ly C, Nizinski C, Vachet C, McDonald IV LW & Tasdizen T (2021). Learning to estimate the composition of a mixture with synthetic data. *Microsc Microanal* **27**, 2522–2525.

Ly C, Olsen AM, Schwerdt IJ, Porter R, Sentz K, Mcdonald LW & Tasdizen T (2019). A new approach for quantifying morphological features of U_3O_8 for nuclear forensics using a deep learning model. *J Nucl Mater* **517**, 128–137.

Ma B, Wei X, Liu C, Ban X, Huang H, Wang H, Xue W, Wu S, Gao M, Shen Q, Mukeshimana M, Abuassba AO, Shen H & Su Y (2020). Data augmentation in microscopic images for material data mining. *npj Comput Mater* **6** (125). doi: https://doi.org/10.1038/s41524-020-00392-6.

Matson T, Farfel M, Levin N, Holm E & Wang C (2019). Machine learning and computer vision for the classification of carbon nanotube and nanofiber structures from transmission electron microscopy data. *Microsc Microanal* **25**, 198–199.

Odena A, Olah C & Shlens J (2017). Conditional image synthesis with auxiliary classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, Sydney, NSW, Australia, August 6–11, 2017, *Proceedings of Machine Learning Research*, vol. 70, Precup D & Teh YW (Eds.), pp. 2642–2651. PMLR. Available at http://proceedings.mlr.press/v70/odena17a.html.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J & Chintala S (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E & Garnett R (Eds.), pp. 8024–8035. Curran Associates, Inc. Availeble at http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pazdernik K, Lahaye NL, Artman CM & Zhu Y (2020). Microstructural classification of unirradiated $LiAlO_2$ pellets by deep learning methods. *Comput Mater Sci* **181**, 109728.

Ronneberger O, Fischer P & Brox T (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference*, Munich, Germany, October 5–9, 2015, Proceedings, Part III, Lecture Notes in Computer Science, vol. 9351, Navab N, Hornegger J, Wells WM & Frangi AF (Eds.), pp. 234–241. Springer. Available at https://doi.org/10.1007/978-3-319-24574-4_28.

Rossen J & Scrivener K (2017). Optimization of SEM-EDS to determine the C–A–S–H composition in matured cement paste samples. *Mater Charact* **123**, 294–306.

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein MS, Berg AC & Li F (2015). Imagenet large scale visual recognition challenge. *Int J Comput Vis* **115**, 211–252. https://doi.org/10.1007/s11263-015-0816-y

Samad JE, Hashim S, Ma S & Regalbuto JR (2014). Determining surface composition of mixed oxides with pH. *J Colloid Interface Sci* **436**, 204–210.

Sarkar A, Alamelu D & Aggarwal SK (2009). Laser-induced breakdown spectroscopy for determination of uranium in thorium–uranium mixed oxide fuel materials. *Talanta* **78**, 800–804.

Schwerdt IJ, Hawkins CG, Taylor B, Brenkmann A, Martinson S & Iv LWM (2019). Uranium oxide synthetic pathway discernment through thermal decomposition and morphological analysis. *Radiochim Acta* **107**, 193–205.

Simonyan K & Zisserman A (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 7–9, 2015, Bengio Y & LeCun Y (Eds.), Conference Track Proceedings. Available at http://arxiv.org/abs/1409.1556.

Singh R, Shah V, Pokuri BSS, Sarkar S, Ganapathysubramanian B & Hegde C (2018). Physics-aware deep generative models for creating synthetic microstructures. *CoRR*. Available at http://arxiv.org/abs/1811.09669.

Trampert P, Schlabach S, Dahmen T & Slusallek P (2019). Deep learning for sparse scanning electron microscopy. *Microsc Microanal* **25**, 158–159.

Trapezoidal rule (2021). https://en.wikipedia.org/wiki/Trapezoidal_rule.

Vasudevan RK & Jesse S (2019). Deep learning as a tool for image denoising and drift correction. *Microsc Microanal* **25**, 190–191.

Wang Z, Simoncelli EP & Bovik AC (2004). Multi-scale structural similarity for image quality assessment. *Signals Syst Comput* **2**, 1398–1402.

Yang HJ, Oh M, Jang J, Lyu H & Lee J (2020). Robust deep-learning based autofocus score prediction for scanning electron microscope. *Microsc Microanal* **26**, 702–705.

Appendix A

A.1. Image Synthesis Model Architecture and Synthesis Process

A.1.1. Model Architecture

Figure A.1 shows the architecture of the proposed image synthesis model. This model uses VGG-16 (Simonyan & Zisserman, 2015) pretrained on ImageNet (Russakovsky et al., 2015) to extract feature maps for Gram matrix computation. We extracted five layers from the network in which each layer corresponds to the first convolution layer of each block in VGG-16 (Simonyan & Zisserman, 2015). Equation (3) is then carried out to minimize the difference between the Gram matrices of the extracted layers of a set of reference images and those of the synthesized image for a certain number of iterations to obtain the desired synthesized image. Moreover, for each synthesized sample, the vector ω can be either randomly sampled or controlled by a user to control the presence of each reference image in the synthesized image.

A.1.2. Image Synthesis Process

We used the architecture presented in Figure A.1 to generate images of mixtures of ADU–UO_4 and ADU–UH–SDU. We set both α and γ to 1 and optimized equation (3) with a learning rate of 0.001. For the ADU–UO_4 mixtures,

$$\mathcal{L} = \alpha \sum_{l \in L} \frac{1}{4C_l^2} \left\| \mathbf{G}^l(\mathbf{x}_G) - \sum_{k=T_1}^{T_n} \omega_k \mathbf{G}^l(\mathbf{x}_k) \right\|_2^2 + \gamma \mathcal{L}_{\mathrm{TV}}$$
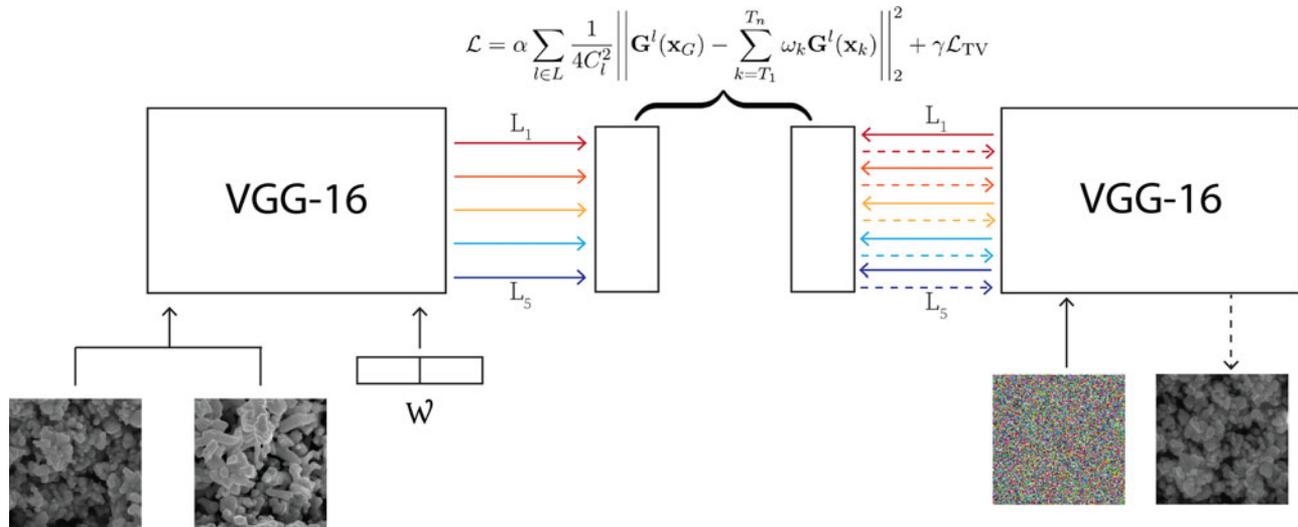

Fig. A.1. The architecture of the proposed image synthesis model. We utilized the VGG-16 network (Simonyan & Zisserman, 2015) pretrained on ImageNet (Russakovsky et al., 2015) to extract feature maps used to compute Gram matrices in the objective function. Each extracted feature map is labeled with a different color in the figure. The solid lines show the forward path and the dashed lines represent the backward path to update the white noise input.

we generated images of size 512×512. We generated images of size 128×128 for the ADU–UH–SDU mixtures. The smaller resolution in the generated images of ADU–UH–SDU mixtures is done to account for the difference in scale between the reference images (i.e., images of 100% ADU, 100% UH, and 100% SDU). Specifically, the images of 100% ADU, 100% UH, and 100% SDU are of size 512×512. These images were obtained by cropping four overlapping regions of size 512×512 from the original SEM images. However, the images of 100% ADU and 100% UH were acquired with HFW of 1.53 and $3.06 \, \mu m$, whereas the images of 100% SDU were acquired with HFW of $6.13 \, \mu m$, which means that the images of 100% ADU and 100% UH are $4\times$ and $2\times$ larger than the images of SDU. Thus, we performed a scale correction process before using them as input to the image synthesis model. The scale correction process includes resizing to 128×128 for any image that is $4\times$ larger, resizing to 256×256 first, and then randomly cropping a region of 128×128 for any image that is $2\times$ larger. Finally, we randomly cropped a region of 128×128 for images of SDU.

A.2. Mixture Estimation Model Architecture and Operation

A.2.1. Model Architecture

The mixture estimation model is built based on the ResNet-50 (He et al., 2016) model. We replaced the last fully connected (FC) layer with a set of layers including FC, BN (Ioffe & Szegedy, 2015), and dropout. Moreover, we added a global max pooling (GMP) layer in conjunction with global average pooling (GAP) to improve the stability of feature selection. The two variants, MEM-A and MEM-B, have the same architecture except the last FC layer and the objective function used to train them. In the MEM-A model, the number of nodes in the last FC corresponds to the number of materials in the mixture. In other words, the MEM-A used for ADU–UO_4 mixtures has two nodes in the last FC layer, whereas the last FC layer in the model used for ADU–UH–SDU mixtures has three nodes. This model was trained with binary cross entropy objective function defined as

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i). \quad (A.1)$$

where $\hat{y}_i$ and y_i are the predicted and ground-truth values of sample i, respectively, and M is the total number of training samples. The MEM-B model has three nodes in the last FC layer when used for the ADU–UH–SDU mixtures, and a single node when used for the ADU–UO_4 mixtures since the percentage of ADU and UO_4 in a mixture is complementary and can be inferred from the

other. This variant was trained with L_1 objective function defined as

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} |\hat{y}_i - y_i|, \quad (A.2)$$

Figure A.2 details the architecture of both variants. The top output (dashed blue box) is used to determine the presence of materials in a given image, and the bottom output (dashed red box) is used to estimate the precise composition of a given image.

A.2.2. Training and Inference

For both variants, we trained all the layers except the convolution layers within the ResNet-50 (He et al., 2016) model for 20 epochs with a batch size of 8 and learning rate of 0.002. After 20 epochs, we then trained the entire model with the learning rate of 0.0002 and 0.002 for the convolution layers within the ResNet-50 (He et al., 2016) and the rest, respectively, for another 30 epochs. Moreover, we also used learning rate decay, which decreases the learning rates by a factor of 0.95 every 800 iterations.

For mixtures of ADU–UO_4, we trained the mixture estimation models with input images of size 512×512, and the resolution of input was the same during the inference stage. However, since there is a difference in scale between real images of ADU, UH, and the rest of the mixtures in ADU–UH–SDU dataset, we needed to account for this difference. During the training process, we performed a similar scale correction process as described in the image synthesis model above. On the other hand, we wanted to predict materials or estimate the composition on the entire image. Thus, we resized any images that are $4\times$ larger to 128×128 and any images that are $2\times$ larger to 256×256 in the inference stage. Furthermore, the scale correction process is applied only to real images since we already take into account the scale difference for synthesized images in the image synthesis model.

A.3. Micro-Average and Macro-Average AUC

The AUC value is the area under the curve defined by the true positive rate (TPR) as a function of the false positive rate (FPR). Thus, the AUC of a class, k, can be defined as

$$\mathrm{AUC}_k = \mathrm{TRAPZ}(\mathrm{TPR}_k, \mathrm{FPR}_k), \quad (A.3)$$

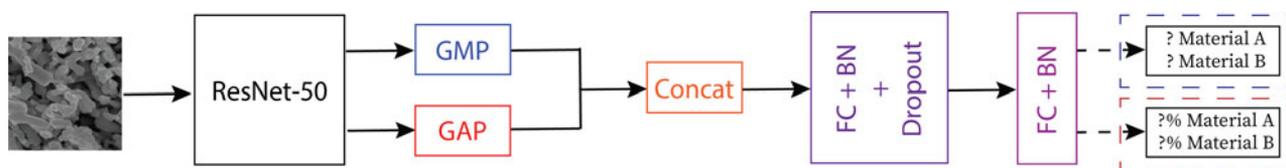where TRAPZ is the area under a curve, which is defined by TPR_k as a

Fig. A.2. The architecture of the mixture estimation model, which has two variants. Both variants have the same architecture except the output layer. The first variant (dashed dark blue rectangle), referred to as MEM-A, is used to predict the presence of materials in a given input image. The second variant (dashed dark red rectangle), referred to as MEM-B, is tasked with estimating the exact composition of a given input image. These two variants were trained separately for the experiments described above.
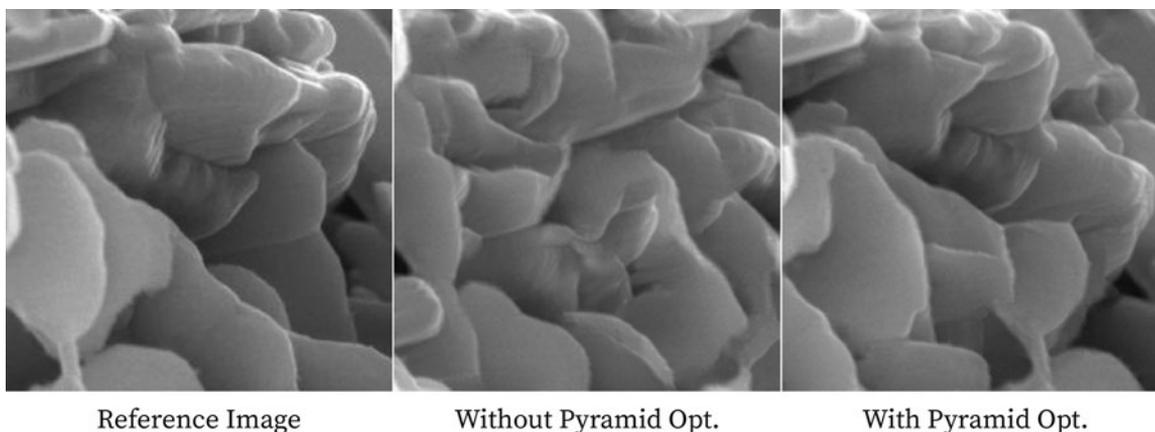

Fig. A.3. A side-by-side comparison of generated images that were synthesized without and with pyramid optimization strategy from the same reference image.

function of FPR_k, computed using the trapezoid rule (2021). Then, the micro-average AUC is computed as

$$AUC_{\text{micro-avg}} = \text{TRAPZ}\left(\sum_k TPR_k, \sum_k FPR_k\right). \quad (A.4)$$

Meanwhile, the macro-average AUC is defined as

$$AUC_{\text{macro-avg}} = \frac{1}{N_k}\sum_k^{N_k} AUC_k, \quad (A.5)$$

where N_k is the total number of classes.

A.4. Large Structures Synthesis with Pyramid Optimization

Generating images with the pyramid optimization strategy helps reduce the computation time as indicated in Table 7. In this section, we demonstrate another advantage of using the proposed pyramid optimization strategy. The pyramid optimization strategy operates on different resolution scales while maintaining the pre-determined kernel size of the filters in the convolution layers; in turn, it enlarges the region the outputs of the convolution layers represent. Hence, the proposed pyramid is able to capture larger structures. Figure A.3 presents an example of this advantage. The last two images in that figure are generated images synthesizing without and with the pyramid optimization strategy, respectively, from using the same reference image on the left. As seen in that figure, the generated image synthesized without the pyramid optimization strategy failed to capture large structures. In contrary, the generated image synthesized with the pyramid optimization strategy properly generated large structures similar to those in the reference image.