

# Artificial Scanning Electron Microscopy Images Created by Generative Adversarial Networks from Simulated Particle Assemblies

Jonas Bals and Matthias Eppler\*

Particle assemblies created by software package Blender are converted into artificial scanning electron micrographs (SEM) with a generative adversarial network (GAN). The introduction of height maps (i.e., surface topography or relief structure) considerably enhances the quality of the artificial SEM images by providing 3D information on the input data. These artificial images serve as input data to train a convolutional neural network (CNN) to identify and classify nanoparticles. Although the performance of the CNN trained with artificial SEM images is slightly inferior to the same CNN trained with real SEM images, this offers a pathway to create training data for segmentation and classification networks for SEM image analysis by deep learning algorithms.

## 1. Introduction

Modern deep learning networks are trained to detect image features from datasets containing millions of images in thousands of different classes.<sup>[1]</sup> After learning to detect such features and to correlate them to classify or detect objects, they are able to perform these tasks better than humans.<sup>[2]</sup> Automated image analysis via segmentation, object detection, and classification by artificial intelligence (AI) and particularly by convolutional neural networks (CNNs; deep learning) is a prominent issue today.<sup>[2–5]</sup>

However, such networks require a large number of manually labeled examples for training which can be a challenging issue as it requires considerable input and effort by human trainers. A potential workaround is to use artificially created images as training data. Of course, their quality must be sufficient to permit the subsequent analysis of real images. Generative adversarial networks (GANs) have emerged to generate image data, including

fake images of humans or animals.<sup>[6–8]</sup> Such artificial images may then be used as training data for neural networks.<sup>[4,9–11]</sup>

Scanning electron microscopy (SEM) is one of the most prominent methods to determine the size and shape of particles.<sup>[12–14]</sup> Deep learning by CNNs has been proposed to analyze SEM images to segment and classify particles with a special emphasis on materials science.<sup>[2,5,15–25]</sup> As with most deep learning approaches, human-classified training data are required. The use of artificially created SEM images by GANs was demonstrated earlier.<sup>[11]</sup> The application of AI procedures has also entered the synthesis of

nanoparticles where many synthesis parameters can influence the outcome with respect to particle size, shape, or particle size distribution.<sup>[26–29]</sup> Here, AI can help to analyze data and to predict suitable synthetic routes, taking experimental data (usually from electron microscopy) into account.<sup>[30–33]</sup>


Here, we present a considerable enhancement of the GAN procedure by introducing image depth information in the form of simulated height maps. This gives the GAN additional information on the particle positions and improves the simulation quality. Particles of selected shape were designed as 3D objects.<sup>[34]</sup> Several hundred particles were then arranged in a levitated state and drop-cast on a flat surface (representing the SEM sample holder) using the software package Blender.<sup>[34]</sup> After settling of the particles, two kinds of images were obtained: one containing the segmentation mask of the generated landscape, and a second depicting a height map of the surface topography of the particles on the sample holder. These height maps were used as training data for a CycleGAN<sup>[9]</sup> as introduced by Rühle et al.<sup>[11]</sup> The SEM images created by the GAN were then used to train a CNN (UNet++) for SEM image segmentation and classification.<sup>[4,9,10]</sup> We demonstrate here that the performance of UNet++ trained on artificial data is comparable to the same model trained on real SEM images.<sup>[35]</sup> We also demonstrate the limitations of this approach.

## 2. Results and Discussion

### 2.1. Simulation of Particle Assemblies by Blender

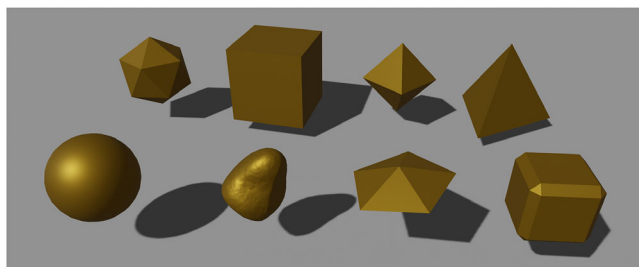
The creation of artificial particles by the program package Blender<sup>[34]</sup> is a straightforward process. Blender offers a variety of predefined shapes which can represent nanoparticles found in

J. Bals, M. Eppler  
Inorganic Chemistry and Center for Nanointegration Duisburg-Essen (CENIDE)  
University of Duisburg-Essen  
45117 Essen, Germany  
E-mail: matthias.eppler@uni-due.de

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300004>.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300004



**Figure 1.** Simulated particles generated by Blender. Several hundred of such objects were dropped on a planar surface to simulate natural particle assemblies.

SEM images (**Figure 1**). Artificial particles with different shapes were created with Blender, followed by simulated dropping onto a planar surface. The particles were held in a levitated state before letting them drop onto the sample holder. The particles were able to interact with each other as well as with the sample holder, i.e., realistic particle assemblies were obtained (**Figure 2**). The Blender module Freestyle was used to create an image of the scenery showing all visible boundaries of objects. In principle, all kinds and numbers of particles can be created and precipitated by Blender. This process yielded height maps and noise-free particle annotation masks. The background was filled in black. The Freestyle module gave each object a dark rim at its border which helped the GAN to distinguish individual particles.

## 2.2. Introduction of Height Maps

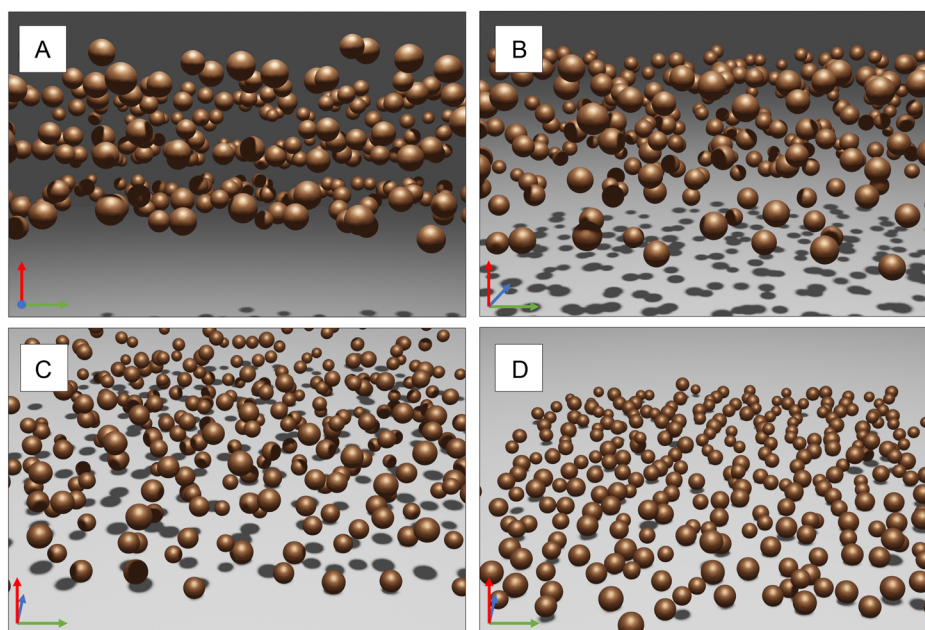
Different particle assemblies were created by Blender with random particle positions in levitation, different shapes, and different particle numbers. Twenty-seven images containing spherical

objects, long and short rods, hexagonal and trigonal plates, cubes, icosahedra, and tetrahedra were used for GAN training. **Figure 3** shows a typical image of spherical particles on a planar surface from Blender.

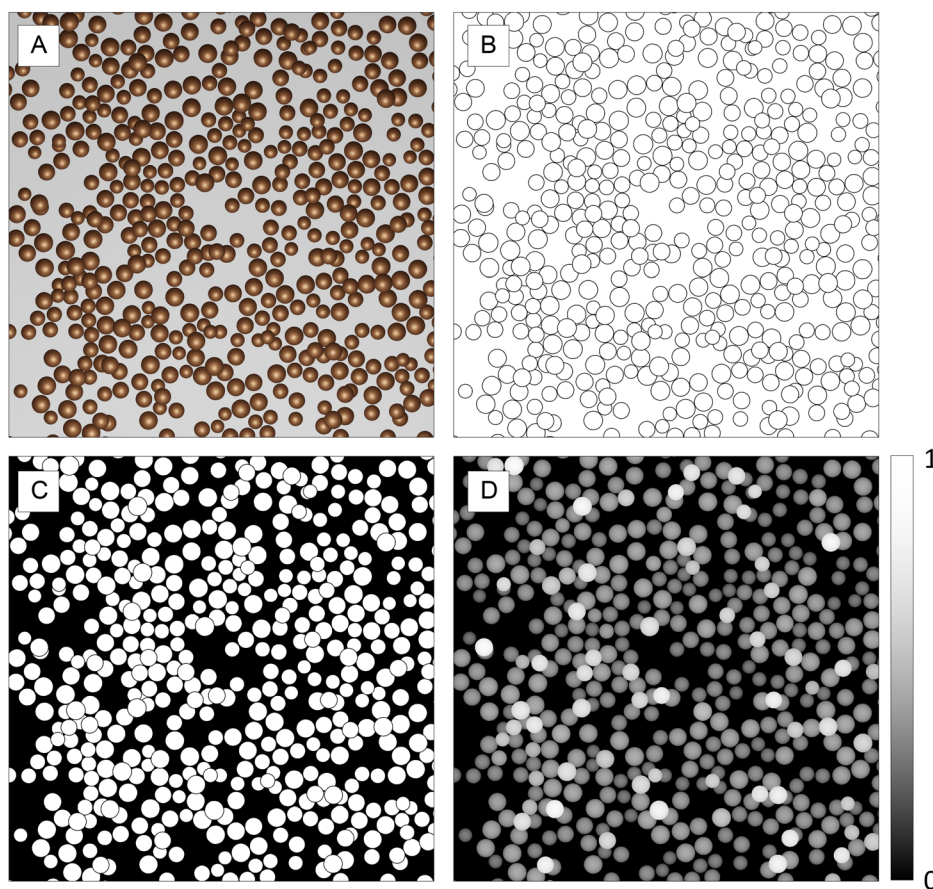
Blender was then used to create a height map of the depicted scenery (**Figure 4**). Essentially, the height map is a parameter assigned to each pixel that denotes the normalized height, i.e., the distance from the surface. This permits the system to distinguish between particles on the top and particles at the bottom. Otherwise, spheres would be just treated as circles with the same height. As SEM images are sensitive to the angle between a particle and the electron beam, this was an important extension to enhance the GAN training process. In essence, this converts 2D training data (annotation masks) to 3D data (height maps).

## 2.3. Training of a GAN on SEM Images

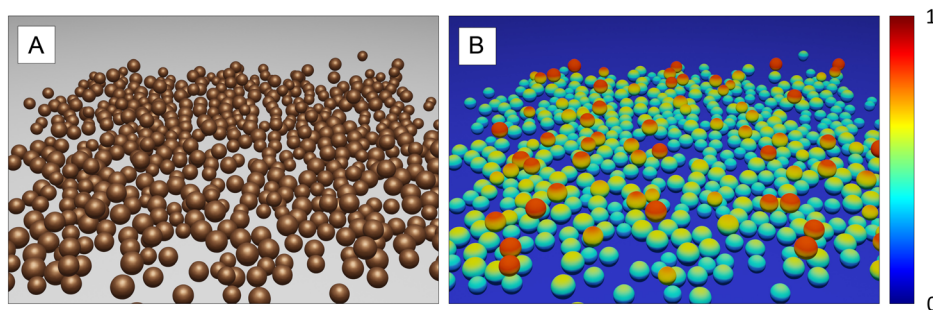
In the original GAN concept, two CNNs are trained. One network, the “Generator”, tries to generate a realistic image, while a second network, the “Discriminator”, evaluates the generator’s result. The discriminator is trained on artificial and real images to assess the image quality while the generator is trained to create images that the discriminator will recognize as real. The GAN training is finished when the generator can produce images that the discriminator cannot distinguish from real images. An extension of this concept is the CycleGAN which is a type of GAN that performs image-to-image translation.<sup>[36]</sup> The CycleGAN is able to transfer an image from one feature space to another without changing the image context. The feature space comprises all features that occur in an image class, i.e., the feature space is the general appearance of an image class (e.g., the appearance of SEM images or their “true nature”). The context of an image is its actual content, e.g., a tree, regardless whether it is shown in a photograph or on a painting.



**Figure 2.** A) Simulation of particles created by Blender, dropping on a planar surface. B,C) The particles start in a layered levitation state and then drop on the surface where they act as individual solid objects. D) Finally, they relax to their final position on the surface.



**Figure 3.** A) Top view of spherical particles settled on a planar surface. B) Top view of the Blender Freestyle map showing all particle borders. C) Annotation mask retrieved from Blender Freestyle. D) Height map generated by Blender. The height map was normalized to [0,1].



**Figure 4.** Topographic landscape of a simulated particle assembly A) with height maps B); relief structure). These images were prepared only for illustration and not used in the simulation.

The CycleGAN architecture is able to convert artificial binary annotation masks (foreground and background) into photorealistic SEM images.<sup>[11]</sup> This was used by Rühle et al. to create artificial SEM images of TiO<sub>2</sub> nanoparticles from their error-free annotation masks.<sup>[11]</sup> Annotation masks of particles contain only 2D information of the depicted objects. As the particle height has considerable influence on the appearance in an SEM image, this is an important additional parameter. The generator of a GAN trained only on annotation maps has to guess the height of each

particle which is an error-prone process; therefore, we extended earlier concepts by introducing height maps.

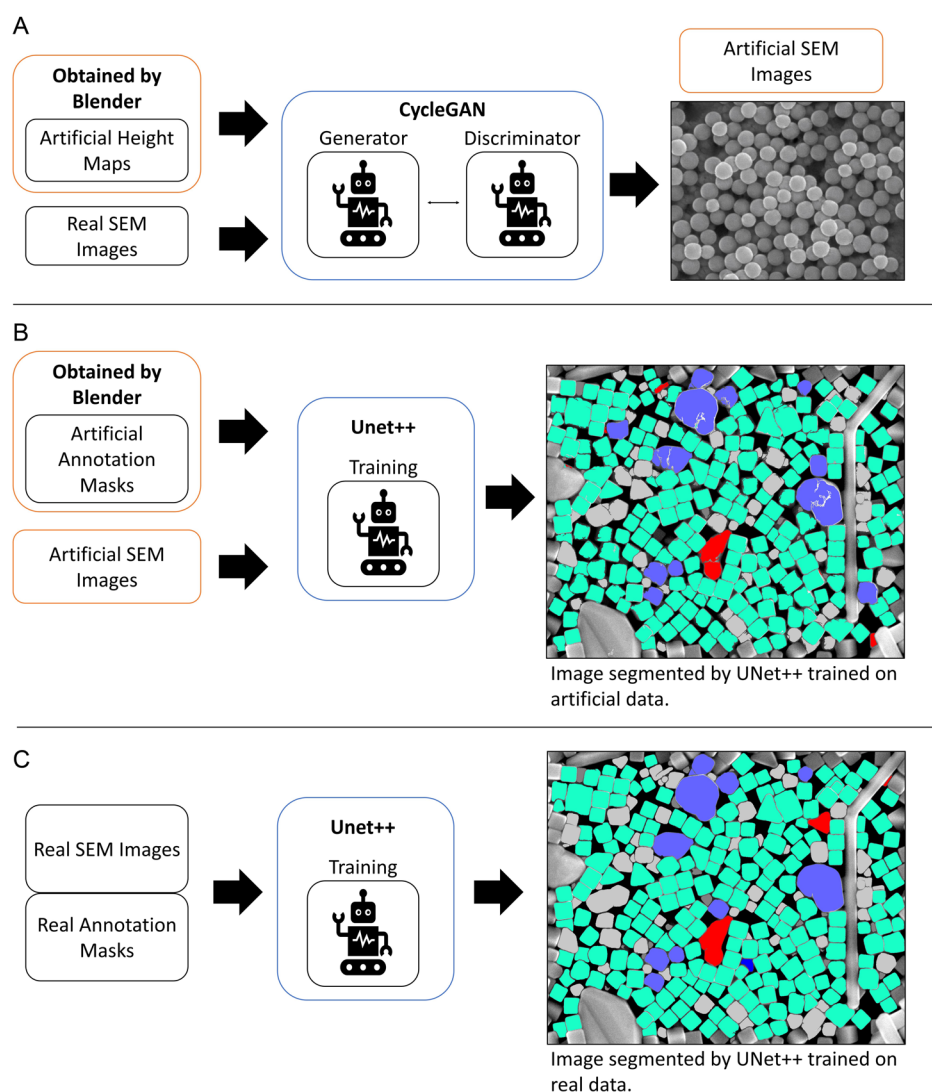
The CycleGAN learns to introduce the features of real SEM particle images into the height maps, making them appear like an SEM image without changing its content concerning particle position and shape. For the training procedure, we adapted the CycleGAN used by Rühle et al.<sup>[11]</sup> We did not change the training method except for the training data and that now included the height maps.



In general, a CycleGAN consists of four neural networks<sup>[36]</sup>: two generator networks and two discriminator networks. The first generator attempts to create a convincing SEM image from an artificial height map. The first discriminator learns to distinguish between artificial SEM images and real SEM images. The artificial SEM image is then cycled back to a second generator which aims to recreate the original height map. The second discriminator learns to distinguish between cycled height maps and artificial height maps. This can be considered as a min–max game in which the generator tries to fool the discriminator that attempts to distinguish artificial SEM images from real ones.<sup>[6]</sup> The training ends when the generator creates images that the

discriminator denotes as realistic. The cycling process of height map to SEM image and back to height map ensures that the SEM image preserves its content. The content is preserved when particles that are present on the height map are present in the created SEM image. Gaussian noise was added to the height maps before conversion into artificial SEM images. After the completed training procedure, the first generator can be used to create artificial SEM images from Blender input data. The other components of the GAN are then not used anymore.

Twenty-seven images of artificial height maps from Blender containing particles of different shape (spheres, plates, pyramids, indented particles, octahedral, and pentagonal bipyramids) were



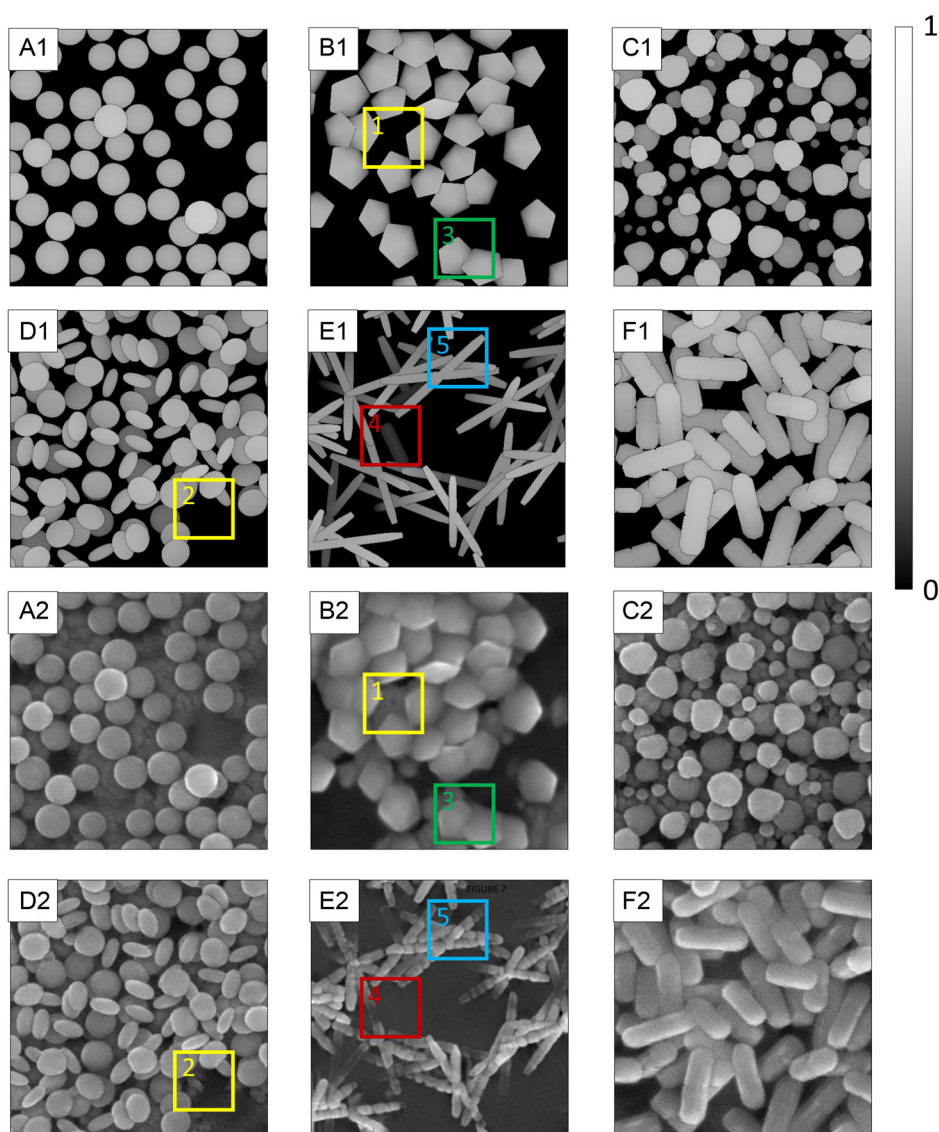
**Figure 5.** Full workflow depicting all stages of data creation and training to create artificial SEM images and to validate their quality for training. A) A CycleGAN is trained with real SEM images and artificial height maps. The generator learns to create realistic SEM images from height maps while the discriminator learns to distinguish them from real SEM images. The discriminator gives feedback to the generator whether the created image appears realistic or not. After the artificial images are judged as realistic, the training is finished. Next, a UNet++ segmentation model is trained on either artificial SEM images B) or on real SEM images C) to segment and classify particles. Both segmentation networks are finally validated with real SEM data. The example shows an image of silver nanoparticles, mostly cubes. The particles were recognized as rods (red), cubes (green), and sphere-like (blue). Note the particles that were excluded from the analysis because they crossed the image border (shown in gray) and the artificial speckles introduced on larger particles by the CNN trained on artificial data.

used to train the GAN. Thirty real SEM images containing nanoparticles in different sizes, shapes (spheres, cubes, indented spheres, and rods), and materials (Ag, Au, CaP, SiO<sub>2</sub>, TiO<sub>2</sub>) were used to train the GAN. Interestingly, the results were not improved by extending the training dataset with more training images. The full workflow is depicted in **Figure 5**.

Typical images created by the GAN are shown in **Figure 6**. The GAN showed a high ability to create realistically appearing images. Spherical (A1/A2) and indented globular objects (C1/C2) led to the most convincing results. We were also able to create good artificial SEM images of pentagonal bipyramids (B1/B2), discs (D1/D2), and short rods (F1/F2). In contrast, long rods were not content preservative, i.e., they were only partially reproduced by the generator (E1/E2). They were split into a chain

of spheres. Assemblies of particles of different shape in one image were also simulated, but with less convincing results (not shown). We were not able to resolve these issues within the presented workflow. Notably, the extension of the training dataset for the GAN with more images did not enhance the image quality.

A Gaussian statistical noise ( $\mu = 0$  and  $\sigma = 0.01$ ) was added before processing to the height maps to mimic surface irregularities. The scattering of electrons leads to Gaussian noise in real SEM images.<sup>[37]</sup> Low levels of noise led to realistic SEM images, whereas a high noise levels (e.g.,  $\mu = 0$  and  $\sigma > 0.1$ ) led to noisy and unrealistically appearing images. In contrast, the simulated images appeared unrealistically smooth without the addition of noise.



**Figure 6.** Different artificial SEM images generated by the GAN. Height maps are shown in a grayscale [0;1] with the sample holder a flat plane at the bottom in black (height value 0) A1–F1). The artificial SEM images after conversion by the GAN are shown in A2–F2). The generator was capable to adapt to most particle morphologies. The simulated SEM images have a realistic illumination, and visible areas of the sample holder appear blurred as they often appear in real SEM images. The colored cutouts represent limits of the simulation process (see text and Figure 7).

The training dataset of real SEM images for the GAN did not contain discs or pentagonal bipyramids; nevertheless, the network learnt to imitate the generic look of SEM images of such particles in images and transferred this to the height map, indicating its ability to emulate previously unknown particles shapes. In some cases, the GAN created “new” particles or tried to fill the background with some structure (Figure 7). It also tended to convert long rods into chains of spheres. Furthermore, the GAN tended to fill empty background areas with texture and even newly created particles. This shows the limits of the GAN simulation procedure. Clearly, feature space and image context are diverging here.

#### 2.4. CNNs (UNet++) Trained on Either Real or Artificial SEM Images

The primary objective of this work was to use artificial SEM images as training data for CNNs to segment and classify particles. In principle, the GAN yields artificial SEM images and error-free segmentation masks without any human interaction, although it tends to change the input context (Figure 7).

The UNet++ architecture was trained on two different datasets, respectively.<sup>[9]</sup> The first CNN was trained on 62 real SEM images and human labeled annotation masks. The training set consisted of the images used to train the GAN as well as of SEM images published by Rühle et al.<sup>[11]</sup> The second CNN was trained on 27 artificial SEM images generated as described above by the GAN. We adopted the methods of Ronneberger et al. for data preparation and training.<sup>[4]</sup>

The CNNs were trained to segment the SEM images into coherent areas of foreground and background and to identify the individual particles (classification).

#### 2.5. Segmentation and Analysis of SEM Images by the CNNs Trained with Either Real or Artificial SEM Images

The performance of both networks was tested against human labeled validation data, containing 14 real SEM images of different materials and shapes, previously unknown to the CNNs. For

this, a previously described routine was used (see ref. [35] for details). The performance of a network was defined by the Intersection over Union (IoU), defined as

$$\text{IoU} = \frac{100 \cdot \text{TP}}{\text{TP} + \text{FP} + \text{FN}} [\%] \quad (1)$$

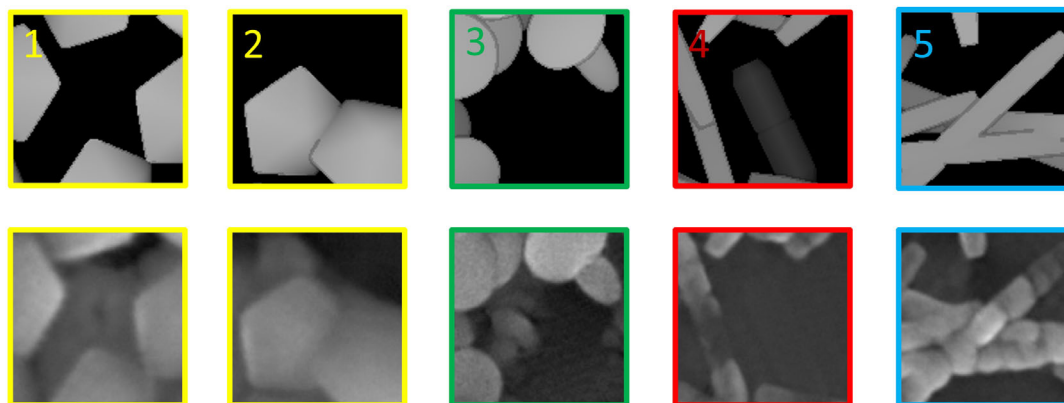
with TP: true positive, FP: false positive, TN: true negative, and FN: false negative, ranging from 0 to 100. The IoU measures the ability of a segmentation network to assign pixels to the classes foreground (i.e., particle) and background.

Particles at the edges of the images were not considered as they might be cut off by the image border.<sup>[17]</sup> Furthermore, particles with a minimal diameter of the bounding box below 10 pixels (i.e., below 88 nm) were excluded from the analysis.<sup>[17]</sup>

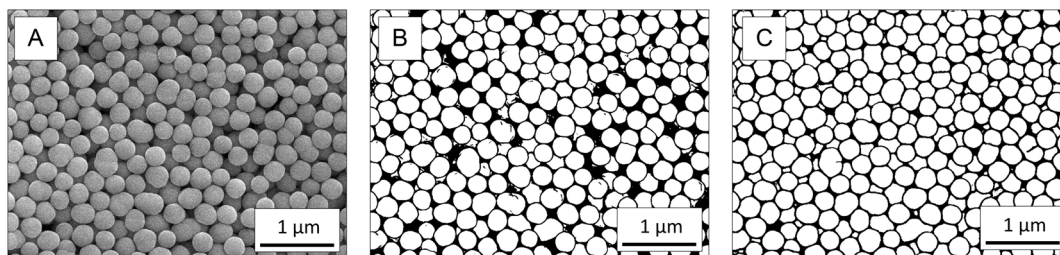
The UNet++ model trained on real data and human labeled annotation masks reached an IoU of 93.46%. It was clearly able to segment most particles in the validation dataset. Particles that were partially covered were also segmented as foreground. In contrast, the UNet++ model trained on artificial data reached an IoU of 81.78%, i.e., significantly poorer, mainly due to the inability to segment covered particles.

Figure 8 shows a typical analysis of a validation image of SiO<sub>2</sub> microspheres, segmented and classified by both models. Despite the poorer segmentation of the model trained on artificial data, it was able to separate most particles. One major difference of both networks was the gap that was generated in the segmentation mask to separate two adjacent particles. This gap was wider for the artificial data model. Partially covered particles were more often ignored or poorly segmented by the artificial data model.

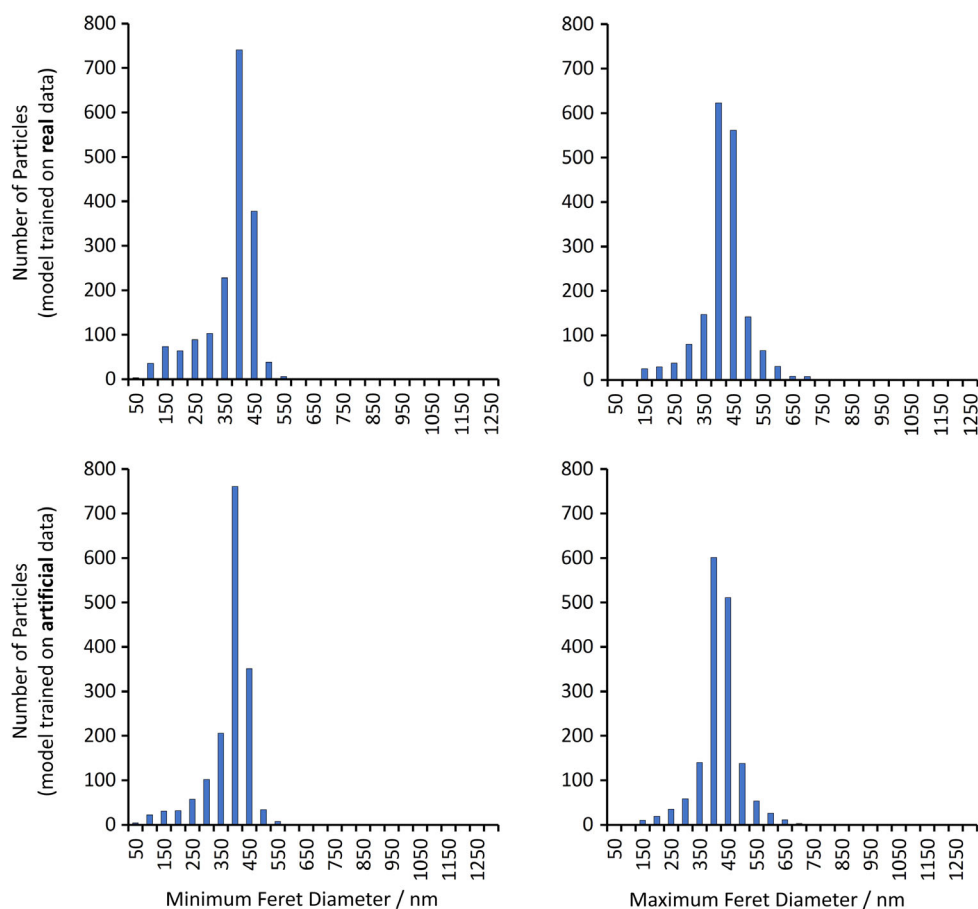
The segmentation and particle identification were quantitatively expressed by computing the Feret diameters of all particles in both segmentation maps, created by the artificial data model and the real-data model (Figure 9). The CNN trained on artificial images showed a poorer separation of single particles, especially of partially covered particles. The particle size distribution was therefore shifted toward smaller diameters. This model also tended to assign some particles with larger Feret diameters in the range of 600–1000 nm due to insufficient separation of



**Figure 7.** Limits of the GAN simulation, taken from Figure 6. Top row: Image from Blender height maps used as input for the GAN. Bottom row: Output of the GAN. Images 1 and 2 show the generation of new particles between the original particles. Image 3 shows how sharp borders between particles led to a blurred fusion. Image 4 shows how particles in the shadow almost vanished. Image 5 shows how the generator converted long rods into a straight chain of spheres. Although the created images appear more realistic than the input data, the GAN changed the context of the original image in all these cases.



**Figure 8.** A) An SEM validation image depicting SiO<sub>2</sub> microspheres, B) segmented either by a CNN trained on artificial data and C) a CNN trained on real data. A notable difference between the segmentations is the width of the black gap between adjacent particles. The segmentation model trained on real data was better to recognize partially covered particles in the background. The image depicted here is a cutout of the full image that was analyzed for particles (Figure 9).



**Figure 9.** Particle size distribution of the particles shown in Figure 8 from the segmentation mask created by UNet++ models trained on real images (top) and artificial images (bottom).

adjacent particles. This is corroborated by the number of detected particles: the CNN trained on artificial images found only 1312 particles whereas the CNN trained on real images found 1658 particles. The equivalent diameter for particles after training with artificial images was  $412 \pm 85$ , and  $383 \pm 65$  nm after training with real data. An experienced human analyst manually determined the diameter of 100 particles in a section of the same image and found  $430 \pm 60$  nm. Within the error margins, both segmentation models and the human analyst found

the same average particle diameter. The same trends were found for images that depicted other uniform particle shapes (data not shown).

Finally, we asked 20 chemically experienced persons to distinguish between artificial and real SEM images. Twenty-eight images were shown to them: 16 artificial and 12 real SEM images. On average, 8 (50%) of the artificial images were wrongly identified as real whereas 2 (17%) or the real images were wrongly identified as artificial. This shows that the quality of



the artificial images was in most cases high enough to fool even humans who are used to analyze SEM images.

### 3. Conclusions

Artificial SEM images created by a GAN can be used as training data for neural networks that segment SEM images. The simulation of particles dropping onto a plane is a convenient method to obtain naturally appearing particle assemblies on a sample holder, together with height maps that give 3D information. In principle, it is possible to create assemblies of any kind of particle shape, including a mixture of them, using the program package Blender. Noise must be added to some degree to the particle height maps, to help the GAN to create realistic images. Too much noise led to worse results.

After training with real SEM images, the GAN produced realistically appearing SEM images of particles of different shape. However, it tended to add nonexisting particles to the background and to convert long rods into chains of spheres. Thus, it changed the content of the image, thus constraining the use of artificial images as training datasets that contain a predefined number of particles with preset size and shape.

The artificially created SEM images were well suited to train a CNN to segment and classify particles from real SEM images. In sets of validation images, a CNN trained on artificial SEM images performed slightly worse than a CNN trained on real (human-classified) SEM images, but its performance was still satisfactory. The extraction of particle size distribution data was comparable for both CNNs and close to that of a human analyst.

In conclusion, SEM images created by GANs are suitable to train CNNs to extract particle properties from SEM images. Although limitations remain, this is a promising way to create large datasets for deep learning that require only little human effort for training.

### 4. Experimental Section

SEM micrographs used for the training were recorded with two different scanning electron microscopes (SEM): first, an Apreo S LoVac (Thermo Fisher Scientific) instrument, and second, a FEI Quanta 400F instrument. For electrically isolating materials, the particles were sputter-coated with AuPd (80:20).

Neural network training was performed with an NVIDIA GeForce GTX 1660 on a Lenovo IdeaCentre T540—15ICK G Workstation. Images used in final validation were not used in any training. Anaconda 4.10.3 with Python 3.9.7 and Tensorflow/Keras 2.8.0 were used to implement the neural networks. OpenCV Version 4.5.3 was used to calculate particle properties.

Additional material containing the training files and images to train the GAN as well as the UNet model for image analysis are available at [www.github.com/Dajadan/SEMcycleGAN](http://www.github.com/Dajadan/SEMcycleGAN).

### Acknowledgements

The authors thank Mrs. Aikaterini Karatzia and Dr. Mateusz Olejnik for provision of their images. For SEM operation and sample preparation, the authors are grateful to Tobias Bochmann, Ursula Giebel, and Dr. Kateryna Loza.

Open Access funding enabled and organized by Projekt DEAL.

### Conflict of Interest

The authors declare no conflict of interest.

### Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

### Keywords

artificial intelligence, deep learning, electron microscopy, generative adversarial networks, nanoparticles, particle size

Received: January 3, 2023

Revised: February 23, 2023

Published online: April 6, 2023

- [1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, *Int. J. Comput. Vision* **2015**, 115, 211.
- [2] K. He, X. Zhang, S. Ren, J. Sun, in *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2016**, pp. 770–778.
- [3] J. Long, E. Shelhamer, T. Darrell, *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ, June 2015, pp. 3431–3440.
- [4] O. Ronneberger, P. Fischer, T. Brox, *arXiv* **2015**, 1505.04597.
- [5] A. Krizhevsky, I. Sutskever, G. E. Hinton, in *Proc. of the 25th Int. Conf. on Neural Information Processing Systems - Volume 1*, Curran Associates Inc., Lake Tahoe, Nevada **2012**, pp. 1097–1105.
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Adv. Neural Inf. Process. Syst.* **2014**, 27, 2672.
- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Commun. ACM* **2020**, 63, 139.
- [8] S. Tyagi, D. Yadav, *Arch. Comput. Methods Eng.* **2022**, 29, 2685.
- [9] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, Berlin **2018**, pp. 3–11.
- [10] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, J. Liang, *IEEE Trans. Med. Imaging* **2020**, 39, 1856.
- [11] B. Ruhle, J. F. Krumrey, V. D. Hodoroaba, *Sci. Rep.* **2021**, 11, 4942.
- [12] H. Fissan, S. Ristig, H. Kaminski, C. Asbach, M. Epple, *Anal. Methods* **2014**, 6, 7324.
- [13] M. Miernicki, T. Hofmann, I. Eisenberger, F. von der Kammer, A. Praetorius, *Nat. Nanotechnol.* **2019**, 14, 208.
- [14] F. Caputo, J. Clogston, L. Calzolari, M. Rösslein, A. Prina-Mello, *J. Control. Release* **2019**, 299, 31.
- [15] E. A. Holm, R. Cohn, N. Gao, A. R. Kitahara, T. P. Matson, B. Lei, S. R. Yarasi, *Metall. Mater. Trans. A* **2020**, 51, 5985.
- [16] J. Schmidt, M. R. G. Marques, S. Botti, M. A. L. Marques, *npj Comput. Mater.* **2019**, 5, 83.
- [17] J. Bals, K. Loza, P. Epple, T. Kircher, M. Epple, *Materialwiss. Werkstofftech.* **2022**, 53, 270.
- [18] M. H. Modarres, R. Aversa, S. Cozzini, R. Ciancio, A. Leto, G. P. Brandino, *Sci. Rep.* **2017**, 7, 13282.
- [19] J. Timoshenko, C. J. Wrasman, M. Luneau, T. Shirman, M. Cargnello, S. R. Bare, J. Aizenberg, C. M. Friend, A. I. Frenkel, *Nano Lett.* **2018**, 19, 520.



- [20] A. B. Oktay, A. Gurses, *Micron* **2019**, 120, 113.
- [21] J. M. Ede, R. Beanland, *Sci. Rep.* **2020**, 10, 8332.
- [22] M. Ilett, J. Wills, P. Rees, S. Sharma, S. Micklethwaite, A. Brown, R. Brydson, N. Hondow, *J. Microsc.* **2020**, 279, 177.
- [23] H. Kim, J. Han, T. Y. J. Han, *Nanoscale* **2020**, 12, 19461.
- [24] B. Lee, S. Yoon, J. W. Lee, Y. Kim, J. Chang, J. Yun, J. C. Ro, J. S. Lee, J. H. Lee, *ACS Nano* **2020**, 14, 17125.
- [25] K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. L. Billinge, E. Holm, S. P. Ong, C. Wolverton, *npj Comput. Mater.* **2022**, 8, 59.
- [26] Y. Xia, X. Xia, H. C. Peng, *J. Am. Chem. Soc.* **2015**, 137, 7947.
- [27] Y. Xia, X. Xia, Y. Wang, S. Xie, *MRS Bull.* **2013**, 38, 335.
- [28] N. Mac Fhionnlaoich, S. Guldin, *Chem. Mater.* **2020**, 32, 3701.
- [29] L. Boselli, H. Lopez, W. Zhang, Q. Cai, V. A. Giannone, J. Li, A. Moura, J. M. de Araujo, J. Cookman, V. Castagnola, Y. Yan, K. A. Dawson, *Commun. Mater.* **2020**, 1, 35.
- [30] E. J. Braham, R. D. Davidson, M. Al-Hashimi, R. Arroyave, S. Banerjee, *Dalton Trans.* **2020**, 49, 11480.
- [31] H. Tao, T. Wu, M. Aldeghi, T. C. Wu, A. Aspuru-Guzik, E. Kumacheva, *Nat. Rev. Mater.* **2021**, 6, 701.
- [32] E. M. Williamson, Z. Sun, L. Mora-Tamez, R. L. Brutchey, *Chem. Mater.* **2022**, 34, 9823.
- [33] W. Zhang, H. Lopez, L. Boselli, P. Bigini, A. Perez-Potti, Z. Xie, V. Castagnola, Q. Cai, C. P. Silveira, J. M. de Araujo, L. Talamini, N. Panini, G. Ristagno, M. B. Violatto, S. Devineau, M. P. Monopoli, M. Salmona, V. A. Giannone, S. Lara, K. A. Dawson, Y. Yan, *ACS Nano* **2022**, 16, 1547.
- [34] Community, B. O. Blender, 3.3.0; Blender Online Community, **2022**, <http://www.blender.org>.
- [35] J. Bals, M. Epple, *RSC Adv.* **2023**, 13, 2795.
- [36] J. Y. Zhu, T. Park, P. Isola, A. A. Efros, in *IEEE CVPR 2017*, IEEE, Piscataway, NJ **2017**, pp. 2223–2232.
- [37] M. Prasad, D. Joy, *Microsc. Microanal.* **2003**, 9, 982.

# DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT  
DUISBURG  
ESSEN

*Offen im Denken*

ub | universitäts  
bibliothek

This text is made available via DuEPublico, the institutional repository of the University of Duisburg-Essen. This version may eventually differ from another version distributed by a commercial publisher.

**DOI:** 10.1002/aisy.202300004

**URN:** urn:nbn:de:hbz:465-20231219-163436-2



This work may be used under a Creative Commons Attribution 4.0 License (CC BY 4.0).