ACS
NANOSCIENCE Au
AN OPEN ACCESS JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/nanoau

Article

UTILE-Gen: Automated Image Analysis in Nanoscience Using Synthetic Dataset Generator and Deep Learning

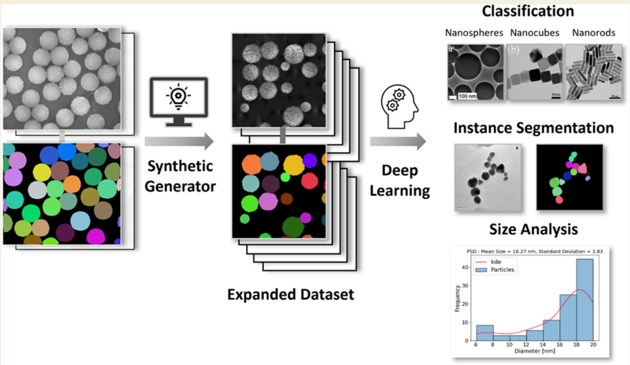
André Colliard-Granero,* Jenia Jitsev, Michael H. Eikerling, Kourosh Malek, and Mohammad J. Eslamibidgoli*

Cite This: ACS Nanosci. Au 2023, 3, 398−407

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: This work presents the development and implementation of a deep learning-based workflow for autonomous image analysis in nanoscience. A versatile, agnostic, and configurable tool was developed to generate instance-segmented imaging datasets of nanoparticles. The synthetic generator tool employs domain randomization to expand the image/mask pairs dataset for training supervised deep learning models. The approach eliminates tedious manual annotation and allows training of high-performance models for microscopy image analysis based on convolutional neural networks. We demonstrate how the expanded training set can significantly improve the performance of the classification and instance segmentation models for a variety of nanoparticle shapes, ranging from spherical-, cubic-, to rod-shaped nanoparticles. Finally, the trained models were deployed in a cloud-based analytics platform for the autonomous particle analysis of microscopy images.



KEYWORDS: nanoscience, deep learning, convolutional neural networks, synthetic data, domain randomization, particle analysis

■ INTRODUCTION

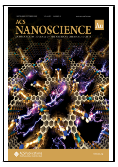
The development of deep learning (DL) algorithms to automate image processing and image analysis is one of the most rapidly growing fields in computer vision (CV).[1] Convolutional neural networks (ConvNets) are among the state-of-the-art methodologies that outperform classical algorithms in a wide variety of image recognition tasks, including verification/identification, classification, object detection, segmentation, image reconstruction, denoising, colorization, or style transfer.[2,3] ConvNets have also been applied to solve various problems in materials research,[4] such as high-throughput classification of microscopy images of catalyst materials and their particle size distribution analysis,[5] detecting defects in nanofibrous materials,[6] and high-resolution synchrotron tomography by denoising reconstructed images of internal material structures.[7]

Instead of performing several handcrafted preprocessing steps, DL models—trained on representative datasets—can learn to extract complex features directly from the raw micrographs, enabling automation in the analysis. Furthermore, well-defined performance metrics validate the reliability of such models, providing robust quantitative standards. However, one of the major challenges for supervised model training tasks is the availability of labeled data. This issue is exacerbated in highly specialized domains, such as nanoscience, where the lack of annotated datasets hinders the building of autonomous analytics tools.[8]

Electron microscopy (EM) is one of the most utilized characterization techniques in nanoscience.[9] Based on the image analysis, nanoparticle (NP) images can be characterized by a variety of parameters, including shape, size, or spatial distribution. After collecting samples and imaging data, researchers often perform particle measurements manually using image analysis software such as Fiji.[10] This approach, however, is tedious and inefficient for high-throughput or real-time analysis due to its slow, laborious, and highly specialized manual procedure.

Semiautomated methods based on classical computer vision algorithms were integrated into the image analysis software, but they are often applicable for simple cases where particles are well segregated or monodispersed. In these methods, conventional procedures like Otsu's binarization and Canny edge detection were adopted for detecting the particle

Received: May 24, 2023
Revised: July 20, 2023
Accepted: July 20, 2023
Published: August 2, 2023



boundaries.[11] Other techniques have also been developed based on the template matching strategy, which assigns a score to each probable particle location in the image.[12,13] Such methods, however, are not often applicable to analyze complex cases with different particle shapes, distributions, or heterogeneous image backgrounds. Furthermore, two-dimensional (2D) mapping complicates semiautomated thresholding techniques for segmenting overlapping and crowded particle systems.

The application of DL algorithms for the analysis of EM data has been thoroughly reviewed in a recent article.[14] For example, the U-Net architecture[15] with StarDist formulation for loss function[16] were trained to automate the particle size distribution analysis of electrocatalyst materials, where various shape, texture, and patterns are generated between the overlapping catalyst NPs and the support material.[17] DL models were also used for real-time segmentation of NPs in liquid phase EM movies to statistically examine the diffusion, reactivity, and assembly kinetics of cube-, prism-, and rod-shaped colloidal NPs.[18] Another example is the ImageDataExtractor software which uses Bayesian DL to segment and quantify NPs of different morphologies.[19]

Supervised DL algorithms for image classification and segmentation require a large amount of annotated and high-quality data for training. While the manual annotation of the regions of interest (ROIs) in the image is time-consuming, various approaches were reported for the synthetic dataset generation (Figure 1). For DL-based image synthesis, generative adversarial networks (GANs) are prevalent.[20] Notably, Zhang et al. developed the DatasetGAN,[21] which generates semantic segmentation masks in addition to synthetic images. The training of GANs is, however, computationally expensive, and they also require large datasets. Other reported techniques are based on data augmentation methods like domain randomization (DR). Applying DR on 2D images is based on cropping of ROI from the original image and randomly pasting them on a similar background after applying geometrical transformations like flipping, rotating, or resizing. Simultaneously, segmentation masks can be created. These simple procedures can provide effective datasets for training DL models and are also adaptable to various systems.

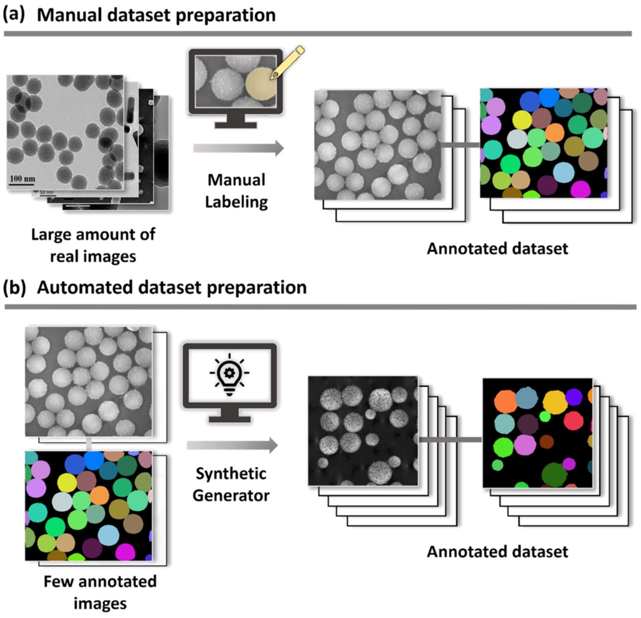


Figure 1. Dataset preparation for supervised model training tasks. Time-consuming, manually performed single-particle annotation in (a) is compared with automated synthetic dataset generation with expanded datasets and instance segmentation masks in (b).[27]

Even though the synthetic images using DR look less realistic than those generated with GANs, there have been several reports of the effective use of such methods for training the DL models. For example, the "Cut, Paste, and Learn" method was proposed by Dwibedi et al.[22] for object detection. Toda et al.[23] applied DR for segmented dataset generation of seeds; here, a pool of different seeds was created from real images and were pasted randomly on a similar background, while the segmentation masks of seeds are also created. Kharin generated 3D shapes of nanoparticles from extracted textures of real EM data,[24] with the particles positioned onto a 3D environment and backgrounds extracted from the micrographs. The obtained annotated dataset was employed to train a model for particle detection.

Likewise, Polyanichenko et al.[25] generated synthetic data based on 3D models of metal−organic frameworks and trained a model to detect and analyze such structures in real time. Due to the extreme complexity and diversity of nanoparticle systems in terms of shape, size and spatial distribution, various textures, or occlusions, it is essential to develop a versatile, system-agnostic, and configurable program that can generate high-quality annotated datasets for DL model training.

In this paper, we first demonstrate a DR-based tool for generating synthetic images of nanoparticle systems with varying shapes along with their instance segmentation masks. The input can be as small as a single image/mask pair, from which the program extracts the background and ROI regions and applies various transformations to generate diverse and customized datasets for DL model training. The generated dataset was used for supervised learning tasks including nanoparticle classification and nanoparticle segmentation. Next, CV-based tools were developed to extract the size of the NPs and to generate particle size distribution plots. The trained models were further deployed into our cloud-based platform, Virtual Minds (ViMi) Labs,[26] which offers an interface for high-throughput image analysis for functional energy materials.

■ METHODS

Figure 2 demonstrates the general workflow for the automated nanoparticle analysis by utilizing our DL-based approach. It involves synthetic image generation to expand the annotated training datasets, supervised learning for image classification and particle segmentation, automated size measurement on predicted ROI, and statistics and visualization of results, followed by model deployment into the imaging platform.

As input to generate the synthetic images, we used the open access Electron Microscopy Particle Segmentation (EMPS) dataset.[27] It consists of 465 micrographs and their corresponding manually annotated labels for segmentation model training. The dataset contains diverse particle shapes, sizes, textures, and distributions. To apply the methodical workflow in Figure 2 to various particle systems, we inspected the EMPS dataset based on the shape of the particles and collected the images in 3 different classes, namely, the spherical NPs, nanocubes, and nanorods.

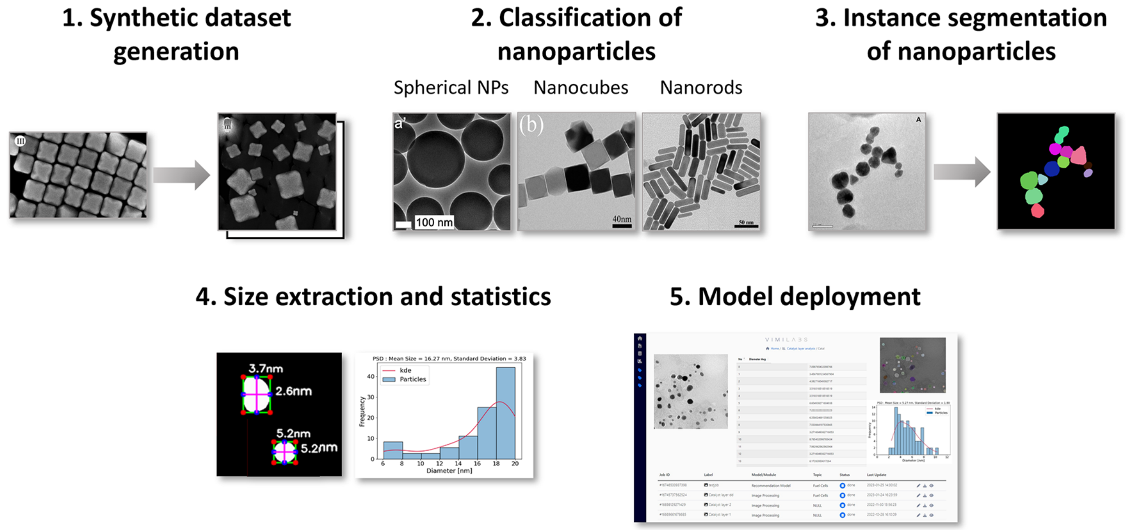

1. Synthetic dataset generation
2. Classification of nanoparticles
Spherical NPs
Nanocubes
Nanorods
3. Instance segmentation of nanoparticles
100 nm
40nm
50 nm
4. Size extraction and statistics
5. Model deployment
3.7nm
2.6nm
5.2nm
5.2nm
PSD : Mean Size = 16.27 nm, Standard Deviation = 3.83
kde
Particles
Frequency
Diameter [nm]

Figure 2. Methodical pipeline for deep learning-based analysis of nanoparticles. The pipeline consists of five main steps. (1) Synthetic dataset generation: a small number of annotated images are utilized to generate a synthetic dataset with the UTILE-Gen tool. (2) Shape classification: a classification model is trained to identify the shape of nanoparticles in images. (3) Instance segmentation model training: instance segmentation models are trained using the synthetic dataset. (4) Particle size and shape extraction and statistics: the predicted masks from the segmentation models are used to extract the size and analyze the shape of individual particles based thereon a statistical analysis is performed. (5) Web app deployment: the pipeline is deployed as a web application, offering access to users for nanoparticle analyses.[27]

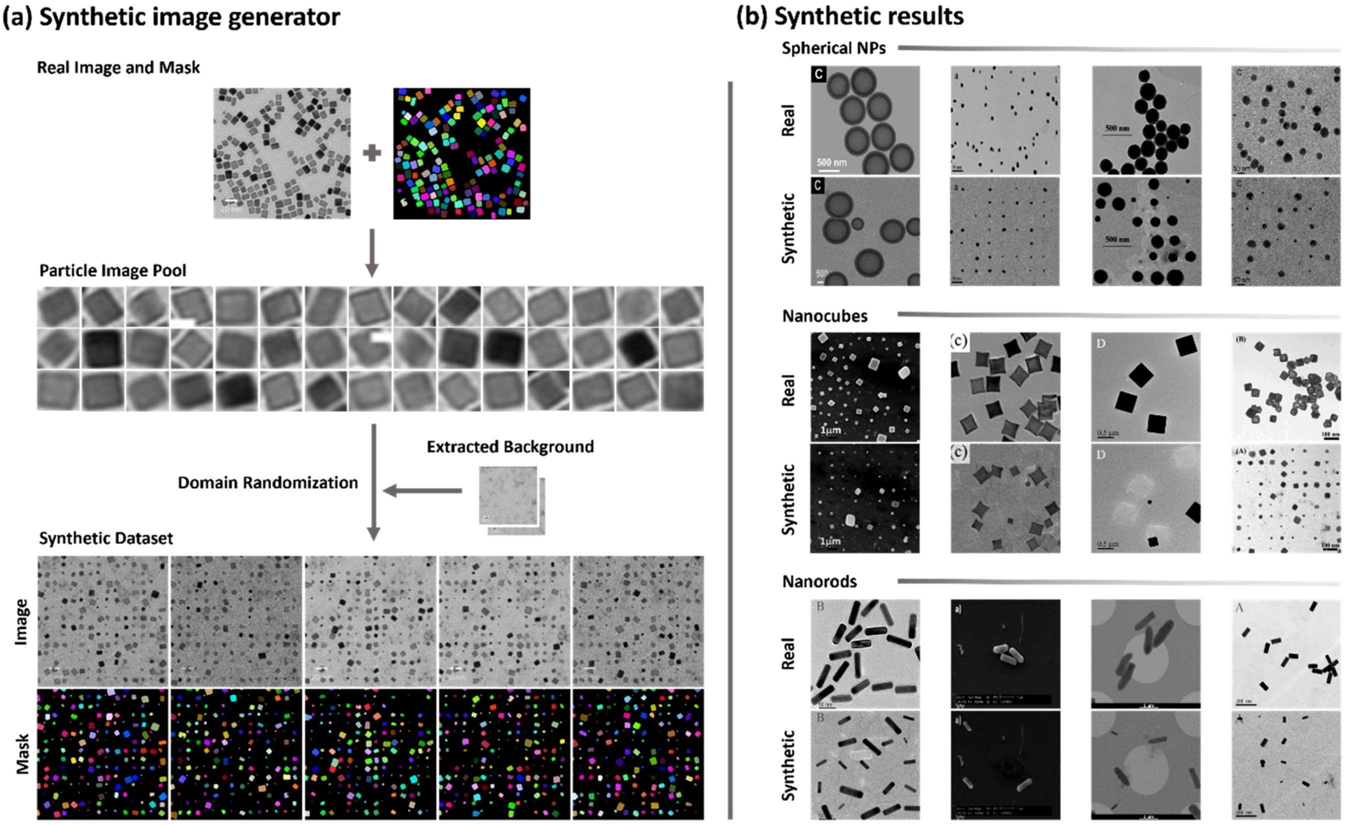


Figure 3. (a) Schematic pipeline for synthesizing images based on image/mask pairs. (b) Examples of real images from the EMPS dataset in three different classes of nanoparticles,[27] along with the generated synthetic image using the tool developed in this work.

Synthetic Image Generator

As shown in Figure 3a, our algorithm takes the EM image and its segmentation mask as the input, automatically extracts the ROIs and background, and saves them into their pools. Thereafter, the software applies customized random transformations to locate the ROIs on the background; simultaneously, it generates the corresponding masks for the synthetic image through the same transformations.

For the first step, ImageJ API and the PymageJ library were used to locate the position of ROI from the annotated mask and to crop the ROI into individual image files for creating the particle image pool.[23] Chopped ROI due to overlapping 2D view or when located at the boundaries of the real image affects the quality of synthetic data, and

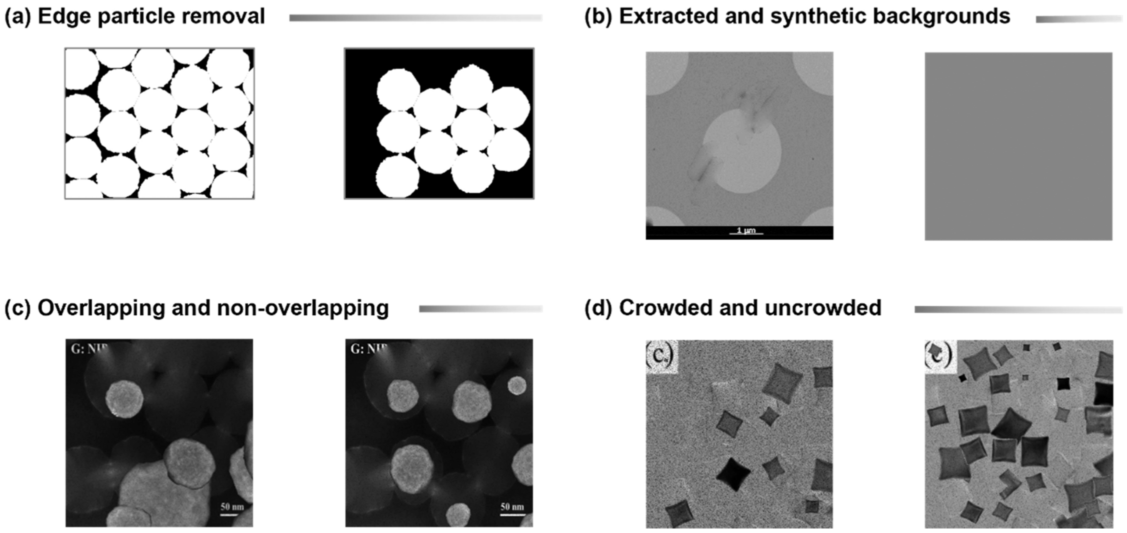

(a) Edge particle removal
(b) Extracted and synthetic backgrounds
1 μm
(c) Overlapping and non-overlapping
G: NI
50 nm
G: NI
50 nm
(d) Crowded and uncrowded
(c)
(e)

Figure 4. Schematic of synthetic image generator tool capabilities. (a) Edge particle removal: artifact reduction by removing edge particles. (b) Inhomogeneous background extraction and synthesis: extraction or creation of new inhomogeneous backgrounds. (c) Synthetic particle arrangements: representation of different arrangements for synthetic particles. (d) Custom crowding selection: feature for selecting the level of crowding in synthetic images.

therefore, they need to be removed from the ROI pool. To address the former issue, the size of individual ROI was quantified in pixels and a tunable parameter was set in the program to eliminate all particles below a certain size. Here, the particles with less than 60% of the average size of ROIs were removed from the pool to remove the truncated overlapping particles. As shown in Figure 4a, in order to filter the chopped particles at the boundaries from the pool, the pixel values from the border were inspected and the ROI with values detected at the boundaries of the image was removed from the mask. This process could prevent the extraction of chopped boundary particles.

To generate the background canvas, as shown in Figure 4b, two different methods were implemented in the software to address the cases with homogeneous and inhomogeneous background textures. The first one employs the inPaint() function from the OpenCV library where the annotated objects in the mask can be removed from the real image and substituted by a neutral hue based on the surrounding pixels of the object. The result generates an empty canvas and keeps the features of the original image such as the background inhomogeneities, or the scale bar. The second method identifies the background from the real mask and computes the average background pixel's intensities from the corresponding real image. The mean intensity can then be used to create a canvas for the synthetic images. The latter is more suitable for systems with homogeneous backgrounds.

Next, a series of hyperparameters were specified to customize the synthetic images including options for creating the overlapping vs nonoverlapping ROIs (Figure 4c) or specifying the number of particles in the image (Figure 4d). To generate the image/mask pairs, first, a background is randomly taken from the background pool along with a black frame for the mask. Next, a particle is randomly picked from the particle pool, rotated, and rescaled within a specified range to introduce variations. Then, the particle is pasted onto the background, and a particle filled with a unique color is placed into the mask canvas in the same location. For overlapping systems, the ROIs are randomly pasted into the canvas, while for nonoverlapping instances, the canvas is divided into a grid based on the number of particles and populated afterward to avoid overlapping. This process can be repeated until the required particle count per image is obtained. Next, salt and pepper noise and Gaussian blur filter are added for smoothing and enhancing the realism of the images. Finally, the image/mask pairs are converted to TIFF files and saved in a folder structure for deep learning model training. The output size of the generated images and masks is set by default to 1024 × 1024 pixels.

Table 1 summarizes the hyperparameters in the program to generate the customized dataset. To prepare the dataset for supervised learning tasks in this work, the number of particles per image was selected as almost the same number as in the original image to avoid over- and undercrowded images. Salt and pepper noise and Gaussian blur were set to 0.1 and 0.3, respectively, and the range scaling factor was set at 0.3 and 2 for the lower and upper limits, respectively.

Table 1. Description of the Input Parameters for the Creation of Customized Datasets with the Synthetic Dataset Generator Tool

hyperparameter	input	description
dataset size	integer	number of generated synthetic image/mask pairs
particles per image	integer range	random number inside the range of objects per image
overlapping particles	boolean	if true, the objects are pasted randomly in a grid with minimal overlap
synthetic backgrounds	boolean	if true, the mean background intensity is extracted, and a synthetic background is added to the pool
salt-pepper noise	upper-limit float	random SP-noise between 0 and the given limit is applied to every image
gaussian blur	upper-limit float	random GB between 0 and the given limit is applied to every image

Supervised Model Training

The synthetic data was used to train ConvNet-based image classification and instance segmentation models. In the typical classification architectures, an image is provided as input and the probability distribution over the different predefined classes is predicted as output. The image is a 3D tensor (column × row × channel) and the ConvNet applies a set of convolutions defined via learnable kernels (filters or feature detectors) by sliding over spatial locations of the image to generate the transformed representations or activation maps. The activation maps have smaller spatial dimensions and larger depth in the channel dimension. Each generated activation map channel learns to respond to certain visual features of the image—from low- to high-level features as it goes deeper into the convolutional layers.

The convolution operations are typically followed by applying activation functions and pooling operations. Activations functions

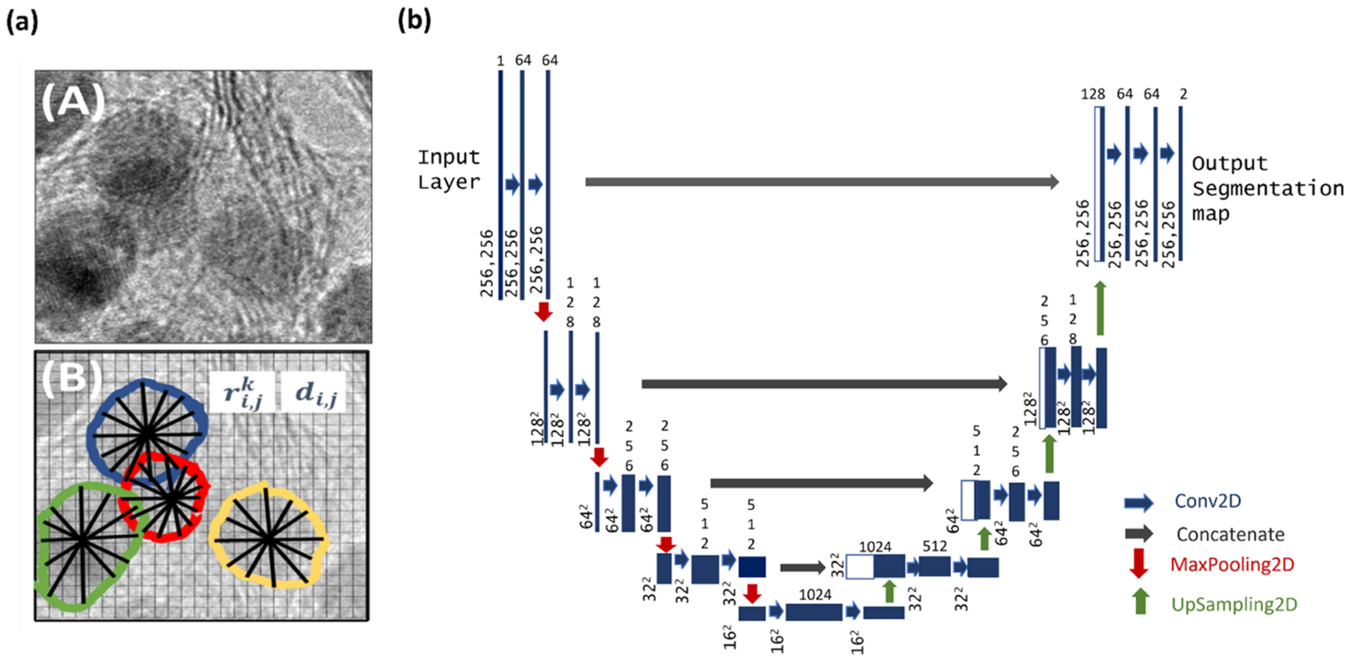


Figure 5. (a) Sample EM image depicting overlapping particles. The StarDist model[16] captures star-convex polygons for each particle via radial distances ($r_{i,j}$) and object probabilities ($d_{i,j}$) for the pixel i,j. (b) The U-Net architecture.[15,16]

such as Rectified Linear Units (ReLU) introduce nonlinearities into the layer-wise signal transformation. After an image is passed through a series of convolutional layers and the fully connected layer, the classifier can be applied to the 1D feature vector to generate the probability distribution over each class and the one with the highest probability is mapped as the class label for the image. During the model training, the gradient of the loss function is computed for each weight in the convolutional kernels using backpropagation, and the weights are updated to minimize the loss function.

To classify the images into different nanoparticle systems, we used transfer learning and fine-tuned a pretrained model on our synthetic dataset. A pretrained model can be initially trained on a large generic natural image dataset and then be used to extract useful generic features from the image. In our work, we employed the Xception[28] model pretrained on the ImageNet dataset.[29] For classification, the synthetic dataset of nanoparticles was generated using 179 real images in the three different classes—spherical-, cubic-, and rod-shaped. The dataset was split into two sets for training, including 126 images from which 750 images were created synthetically, and 53 images for validation.

For the instance segmentation task, the U-Net architecture with StarDist loss function was chosen (Figure 5).[15−17] Standard architectures, for instance segmentation, involve fully convolutional networks, which first perform convolution and down-sampling operations to extract the features (encoder), followed by up-sampling and transpose convolution operations (decoder) until the starting input size is reached. The U-Net model, in addition to the described data flow, concatenates the up-sampled decoder features with the corresponding ones from the encoder (Figure 5b). The concatenation of the encoder and decoder feature maps across all resolution levels enables an improved recognition of object boundaries and edges, leading to an increased performance and more accurate output segmentation maps. The StarDist add-on to the loss function of the U-Net backbone overcomes the typical issues for the dense prediction of merged bordering particles. In StarDist, each pixel from the detected ROI is parameterized by two values, radial distances, and object probabilities (Figure 5a). For the radial distance, a star-convex polygon is fitted from the pixel position to the edges of the particle and the object probability value is calculated from the shortest distance of the pixel to the edge of the particle. Non-maximum suppression (NMS) technique is then applied to eliminate the overlapping detections with lower object probabilities leading to segmentation of individual overlapping instances.

Standard augmentation techniques[30] were employed to expand the training set through basic image manipulations such as cropping images into patches and augmenting by rotation, varying intensity, or applying Gaussian blur. The instance segmentation models were trained using the default hyperparameter in the StarDist implementation of U-Net with the number of epochs and steps per epoch set as 200 and 100, respectively. The performance of the trained model was analyzed using the Intersection over Union (IoU) threshold. The IoU measures the number of pixels common between the manual annotated masks, the so-called ground truth, and prediction masks, obtained by the model, divided by the total number of existing pixels in both masks. A true-positive (TP) represents the case if a prediction−target mask pair for nanoparticles has an IoU score that exceeds 0.5; a true-negative (TN) is that for background. On the other hand, a false-positive (FP) indicates a predicted NP mask with no associated ground truth mask, and a false-negative (FN) indicates a ground truth NP mask with no associated predicted mask. This way, the metrics of accuracy, precision, recall, and $F1$ score are defined as

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \times Precision \times Recall/(Precision + Recall)$$

Given the segmentation maps obtained from the model, each ROI was cropped into an individual image. Next, we employed the OpenCV library to draw a bounding box around the particles and measured the pixels along with the X and Y directions. Real distances can then be calculated from the pixel values with image calibration. Finally, the obtained sizes for the X and Y axis were used to generate the histograms for the particle size analysis and other statistics such as perimeter, area, solidity, roundness, extent, and aspect ratios.

■ RESULTS AND DISCUSSION

For classifying nanoparticles, the comprehensive process of fine-tuning and assessing the pretrained model after dividing

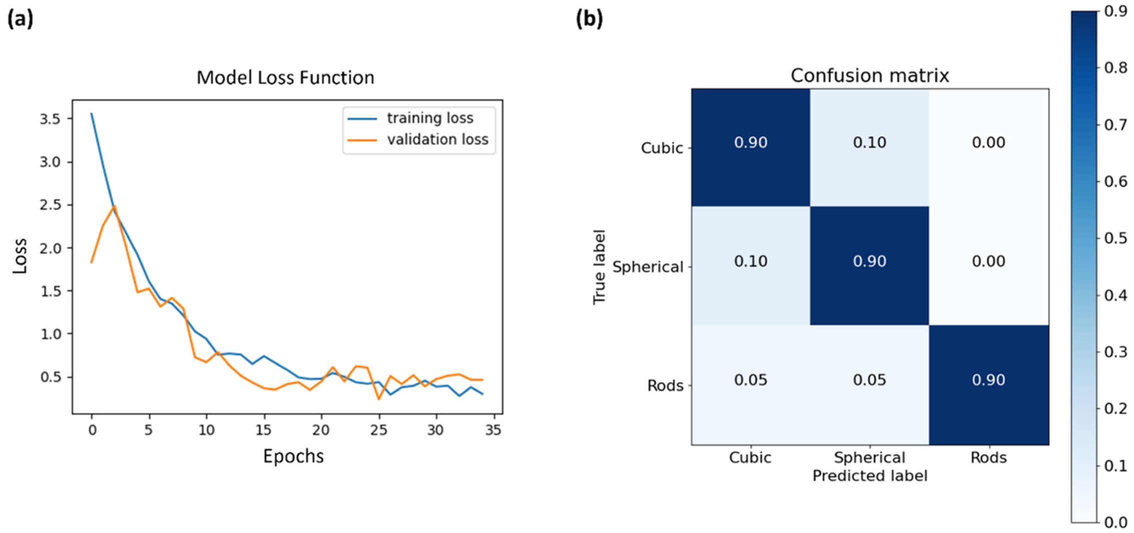


Figure 6. (a) Learning curves for fine-tuning the pretrained Xception model for the classification of nanoparticles into three classes: cubic-, spherical-, and rod-shaped particles. (b) Normalized confusion matrix on the test set.

the data involves selecting the network architecture, initializing the weights from a pretrained model to improve the performance, and selecting the optimal learning rate to determine how large are the steps toward the optimal minimum and regularization strength. These process steps are required to minimize the loss function, which computes how large the error between the manual annotated mask and the predicted mask by the model is. Afterward, we evaluate the model using the validation set. Figure 6a shows the learning curves (validation loss vs epoch number) of the pretrained Xception model fine-tuned on our dataset. The training and validation loss curves displayed no signs of overfitting/underfitting during the model training, On the validation set, the model achieved an accuracy of 91%. Figure 6b shows the normalized confusion matrix for the classifier on the test set, which was not used as input for the synthetic dataset generator. The x-axis shows the predicted labels, and the y-axis shows the true labels. The classification accuracy of ~90% can also be seen in the confusion matrix.

In this study, we compared state-of-the-art models and augmentation techniques by assessing the impact of synthetic data on the prediction performance of models in cases with limited data. The Xception model and the ConvNeXtBase model,[31] a state-of-the-art deep learning architecture, were trained on two datasets: one consisting of 15 real images per class and mixed dataset consisting of 15 real images per class and 5 synthetic images per real image. Furthermore, we evaluated the use of advanced augmentation techniques, such as RandAugment and MixUp,[32,33] to determine their effects on model performance in comparison to the base models.

Table 2 compares the classification accuracies from the models trained on real images, as well as mixed real and synthetic images using various augmentation techniques. The Xception model demonstrated improved accuracy from 30 to 64% when advanced augmentation methods were applied on real images,[32,33] while the best performance was achieved by combining synthetic and real images from 68 to 75%. On the other hand, the classification accuracy of more advanced architectures like ConvNeXtBase network showed a slight decrease in performance when subjected to the use of synthetic data, saturating at 85%, while the employment of state-of-the-art augmentations could slightly increase the performance to 89% on our dataset.

Table 2. Comparison of Model Accuracy Using Real Data, Mixed Real and Synthetic Data, and Various Augmentation Techniques for the Xception and ConvNeXtBase Networks

	model accuracy [%]	
model	standard augmentation real/Mix	MixUp + RandAug Real/Mix
Xception	30/68	64/75
ConvNeXtBase	87/87	89/85

For the instance segmentation task, the images in each class were split into approximately 80% for the training set and 20% for the validation set. Next, using the synthetic image generator tool, the training set was expanded by 500%. Afterward, the U-Net model was trained on the real images as well as the expanded training set. Table 3 summarizes the performance of the model for the two cases by calculating the metrics for precision, recall, and *F*1 score. For all particle categories, up to 13% improvements in the performance metrics were obtained using synthetic data. Figure 7 demonstrates a few examples of the segmentation results on the test set, for the model trained on the expanded dataset, and for the three different particle categories.

Table 3. Performance Comparison of Segmentation Model Trained on Only Real Data versus the Mixture of Real and Synthetic Data

dataset	precision [%] real/mix	recall [%] real/mix	*F*1 [%] real/mix
nanocubes	89/90	72/78	79/84
nanoparticles	81/94	85/92	84/93
nanorods	86/92	87/93	87/93

To further analyze the utility of the synthetic image generator tool, further experimentations were conducted by starting from the worst-case scenario where only one annotated image is available. Even though in this extreme case, training a DL model is not relevant, by generating synthetic data, we can enable automation in the image analysis.

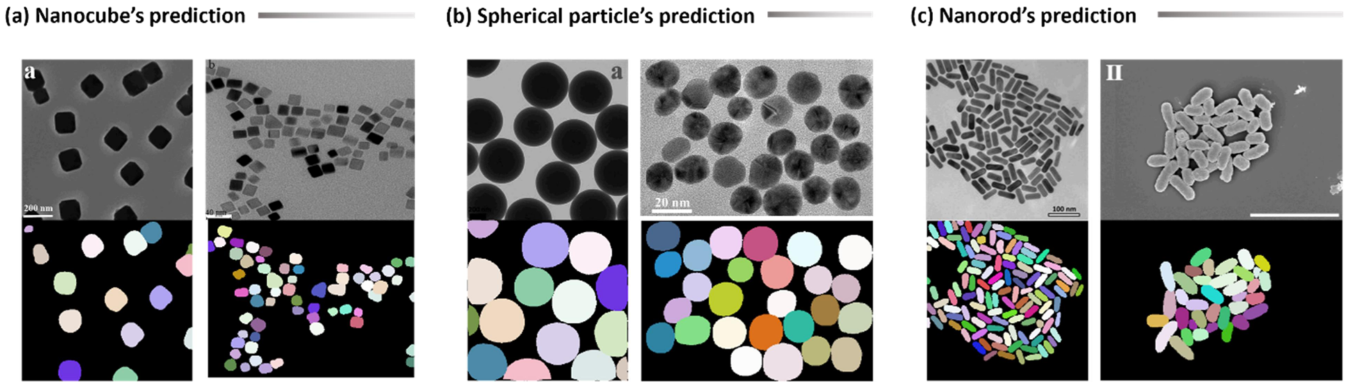

(a) Nanocube's prediction
(b) Spherical particle's prediction
(c) Nanorod's prediction
a
b
200 nm
a
20 nm
II
100 nm

Figure 7. Prediction of segmentation model on test images for various nanoparticle classes: (a) nanocubes, (b) spherical nanoparticles, and (c) nanorods.[27]

To test this, one image was randomly selected starting from three different cases of black NPs, gray NPs, and ordered nanorods, and a synthetic dataset of each image was generated consisting of 50 synthetic image/mask pairs. Afterward, for each class, a model was trained on synthetic data. The remaining images in each class were used for validation of the results. As summarized in Table 4, acceptable performances were obtained for the models only trained on the synthetic data in a most difficult scenario where only one annotated real image/mask pair is available. This utility of the tool means that with just a minimal manual annotation effort, we can obtain an acceptable performance of the DL model for the segmentation of various nanoparticle systems.

Table 4. Instance Segmentation Performance Metrics of Three Use-Cases for Models Trained on 50 Synthetic Images Generated from One Real Image

dataset	precision [%]	recall [%]	*F*1 [%]	validation set size
gray NPs	88	63	73	39
black NPs	91	93	92	62
ordered nanorods	80	75	77	8

We further evaluated the performance improvements by varying the number of real image/mask pairs as input for the synthetic image generator. For this test, the largest class dataset was employed, which consists of 93 images of spherical NPs, which were split into 30 images for training and 63 images for validation. From the 30 real image/mask pairs for training, four datasets consisting of 5, 10, 20, and 30 images were created on which the synthetic image generator tool was applied to expand the training set by 500%. Table 5 shows the *F*1 score obtained from this experimentation. Consistently, by increasing the number of real images, the *F*1 score increases from 36% for 5 images as the training set to 88% for 30 real images. Expanding the training set by generating 5 synthetic images per real image, the *F*1 score is significantly improved from 36% for 5 real images to 74% on the corresponding expanded training set. Likewise, the *F*1 score was improved from 63, 77, and 88% when the model is trained on 10, 20, and 30 real image/mask pairs to 86, 88, and 92% on the expanded dataset, respectively.

Table 5. *F*1 Score Comparison of Segmentation Models Trained on Different Numbers of Input Real Image/Mask and of the Expanded Synthetic Dataset

*F*1- score [%]	5 real images	10 real images	20 real images	30 real images
0 synth images per real image	36	63	77	88
5 synth images per real image	74	86	88	92

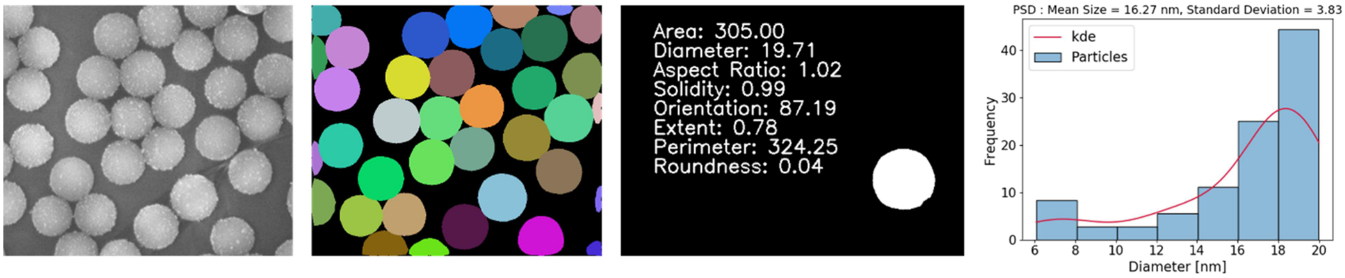
Following the overall workflow to automate the analysis as shown in Figure 2, particle sizes, areas, aspect ratios, solidities, orientations, extents, perimeters, and roundness can be extracted in step 4 with a separately developed CV tool.

The area of the region of interest (ROI) is quantified by the count of nonzero pixels, representing the occupied space by the ROI. The equivalent diameter is calculated based on the area, representing the diameter of a circle with the same area as the ROI.

The aspect ratio of the ROI is determined as the ratio of the width to the height of the individual object of interest, providing an understanding of the shape of the ROI. Solidity measures the compactness of the ROI, determined by the ratio of the ROI's contour area to the convex hull area. The convex hull refers to the smallest convex polygon that can completely enclose the contour. Orientation describes the angle at which the ROI is tilted. It is determined by fitting an ellipse to the contour and reporting the angle of this fitted ellipse. Extent quantifies the ratio of the contour area of the ROI to its bounding rectangle area, indicating how much of the bounding rectangle the ROI occupies. The perimeter is calculated as the arc length of the contour of the ROI, signifying the length of its boundary. Roundness, or circularity, is a measure of how closely the shape of the ROI resembles a perfect circle. It is computed as the ratio of 4π times the area to the square of the perimeter.

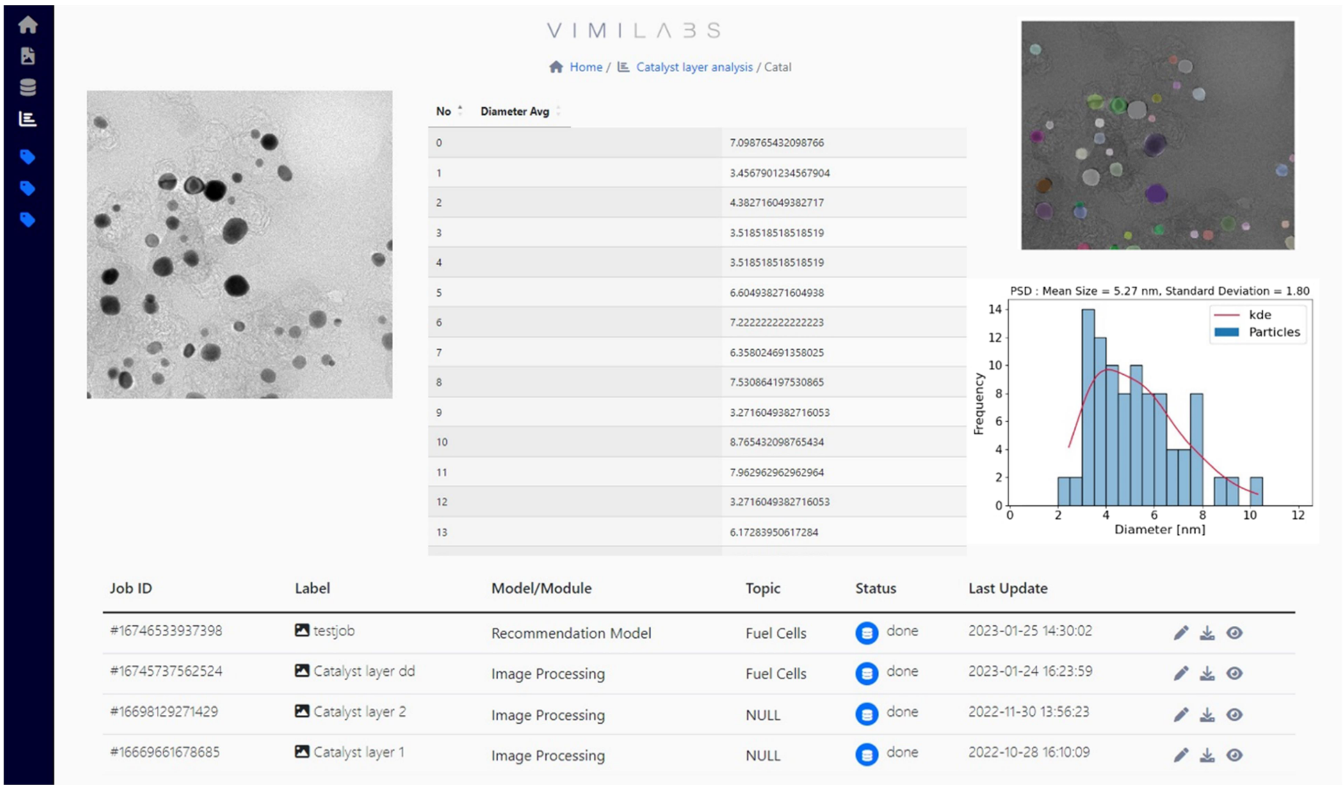
From this result, the particle size histograms can be generated using typical visualization tools with appropriate binning size and fitting function, as exemplified in Figure 8.

The developed DL-based workflow described in this paper was partly deployed into the Virtual Minds Labs (ViMi Labs, vimi.ai). As shown in Figure 9, the imaging module of ViMi labs currently allows the users to upload bulk EM images of electrocatalyst materials and obtain the particle size distribution (PSD) and other distributions plots in a noticeably short amount of time (~10 s processing for each image), while the manual or semiautomated PSD analysis can take up to several hours depending on the number of particles per image. To fit the obtained histogram, the kernel density estimation method and log-normal distribution were tested, giving the kernel



Area: 305.00
Diameter: 19.71
Aspect Ratio: 1.02
Solidity: 0.99
Orientation: 87.19
Extent: 0.78
Perimeter: 324.25
Roundness: 0.04

Figure 8. Automating particle size distribution analysis using the deep learning-based workflow based on instance segmentation and measurements of nanoparticles in EM micrographs.[27]



VIMILABS
Home / Catalyst layer analysis / Catal
PSD : Mean Size = 5.27 nm, Standard Deviation = 1.80
kde
Particles
Frequency
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
Diameter [nm]

No	Diameter Avg
0	7.098765432098766
1	3.4567901234567904
2	4.382716049382717
3	3.518518518518519
4	3.518518518518519
5	6.604938271604938
6	7.222222222222223
7	6.358024691358025
8	7.530864197530865
9	3.2716049382716053
10	8.765432098765434
11	7.962962962962964
12	3.2716049382716053
13	6.17283950617284

Job ID	Label	Model/Module	Topic	Status	Last Update
#16746533937398	testjob	Recommendation Model	Fuel Cells	done	2023-01-25 14:30:02
#16745737562524	Catalyst layer dd	Image Processing	Fuel Cells	done	2023-01-24 16:23:59
#16698129271429	Catalyst layer 2	Image Processing	NULL	done	2022-11-30 13:56:23
#16669661678685	Catalyst layer 1	Image Processing	NULL	done	2022-10-28 16:10:09

Figure 9. Example of the utilization of the trained models for particle size distribution analysis deployed in the ViMi labs platform.[26]

density estimation method a superior fit. However, the user of the code will have the option to choose among fitting methods and obtain the best estimate of the underlying probability distribution. The platform provides a visual representation of the results, including the uploaded image, individual measurements of the objects of interest, particle analysis, and mask of the model for prediction verification. Results can also be downloaded as a .csv file.

■ CONCLUSIONS

This work presented a multistep DL-based methodical workflow to automate microscopy image analysis in nanoscience. It involves generating synthetic annotated imaging data for nanoparticles with varying shapes using domain randomization. Afterward, we trained DL models on the augmented data for classification and instance segmentation, followed by computer vision-based automated measurement of particles, and finally deployment of the model into our cloud-based web application (vimi.ai).

Our synthetic image generator tool is customizable and nanoparticle shape agnostic, which outputs image/mask pairs for image segmentation model training tasks. The software can create a large amount of data employing domain randomization and provides several hyperparameters such as noise, blurring, scaling factor, and number or distribution of the particles which enables the application for various use-cases of interest. The data augmentation enabled by the synthetic image generator tool was extensively evaluated by comparing the performances of various trained deep learning models. Accurate particle classification and segmentation can pave the way for the accelerated characterization of nanomaterials. Future work should focus on developing real-time image analysis tools and deployment into web, mobile, or desktop applications.

■ ASSOCIATED CONTENT

Data Availability Statement

The software for the synthetic image generator tool is available at https://github.com/andyco98/UTILE-Gen.

AUTHOR INFORMATION

Corresponding Authors

André Colliard-Granero − *Theory and Computation of Energy Materials (IEK-13), Institute of Energy and Climate Research, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany; Centre for Advanced Simulation and Analytics (CASA), Simulation and Data Science Lab for Energy Materials (SDL-EM), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany; Chair of Theory and Computation of Energy Materials, Faculty of Georesources and Materials Engineering, RWTH Aachen University, 52062 Aachen, Germany;* orcid.org/0000-0002-4615-3710; Email: a.colliard@fz-juelich.de

Mohammad J. Eslamibidgoli − *Theory and Computation of Energy Materials (IEK-13), Institute of Energy and Climate Research, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany; Centre for Advanced Simulation and Analytics (CASA), Simulation and Data Science Lab for Energy Materials (SDL-EM), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany;* orcid.org/0000-0002-5057-2993; Email: m.eslamibidgoli@fz-juelich.de

Authors

Jenia Jitsev − *Centre for Advanced Simulation and Analytics (CASA), Simulation and Data Science Lab for Energy Materials (SDL-EM), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany; Jülich Supercomputing Center, Forschungszentrum Jülich, 52425 Jülich, Germany*

Michael H. Eikerling − *Theory and Computation of Energy Materials (IEK-13), Institute of Energy and Climate Research, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany; Centre for Advanced Simulation and Analytics (CASA), Simulation and Data Science Lab for Energy Materials (SDL-EM), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany; Chair of Theory and Computation of Energy Materials, Faculty of Georesources and Materials Engineering, RWTH Aachen University, 52062 Aachen, Germany;* orcid.org/0000-0002-0764-8948

Kourosh Malek − *Theory and Computation of Energy Materials (IEK-13), Institute of Energy and Climate Research, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany; Centre for Advanced Simulation and Analytics (CASA), Simulation and Data Science Lab for Energy Materials (SDL-EM), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany;* orcid.org/0000-0002-3021-0813

Complete contact information is available at: https://pubs.acs.org/10.1021/acsnanoscienceau.3c00020

Author Contributions

A.C.-G.: Conceptualization, methodology, software, visualization, writing—original draft, writing—review & editing. J.J.: methodology, supervision, writing—review & editing. M.H.E.: supervision, writing—review & editing. K.M.: supervision, writing—review & editing. M.J.E.: Conceptualization, methodology, supervision, visualization, writing—original draft, writing—review & editing. CRediT: **André Colliard-Granero** conceptualization (equal), methodology (equal), software (equal), visualization (equal), writing-original draft (equal), writing-review & editing (equal); **Jenia Jitsev** methodology (equal), supervision (equal), writing-review & editing (equal); **Michael H. Eikerling** supervision (equal), writing-review & editing (equal); **Kourosh Malek** supervision (equal), writing-review & editing (equal); **Mohammad Javad Eslamibidgoli** conceptualization (equal), methodology (equal), supervision (equal), visualization (equal), writing-original draft (equal), writing-review & editing (equal).

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Federal Ministry of Science and Education (BMBF) under the German−Canadian Materials Acceleration Centre (GC-MAC) grant number 01DM21001A. They also gratefully acknowledge the Gauss Centre for Supercomputing e.V. (http://www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS[34] at Jülich Supercomputing Centre (JSC) and the support of the Helmholtz Imaging Platform HIP (a platform of the Helmholtz Incubator on Information and Data Science).

REFERENCES

(1) Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vision* **2020**, *128*, 261−318.

(2) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proc. - IEEE Conf., Comput. Vision Pattern Recognit.* **2016**, 770−778.

(3) Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M. S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27−48.

(4) von Chamier, L.; Laine, R. F.; Jukkala, J.; Spahn, C.; Krentzel, D.; Nehme, E.; Lerche, M.; Hernández-Pérez, S.; Mattila, P. K.; Karinou, E.; et al. Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* **2021**, *12*, No. 2276.

(5) Eslamibidgoli, M. J.; Tipp, F. P.; Jitsev, J.; Jankovic, J.; Eikerling, M. H.; Malek, K. Convolutional neural networks for high throughput screening of catalyst layer inks for polymer electrolyte fuel cells. *RSC Adv.* **2021**, *11*, 32126−32134.

(6) Napoletano, P.; Piccoli, F.; Schettini, R. Anomaly detection in nanofibrous materials by CNN-based self-similarity. *Sensors* **2018**, *18*, 209.

(7) Hendriksen, A. A.; Bührer, M.; Leone, L.; Merlini, M.; Vigano, N.; Pelt, D. M.; Marone, F.; Di Michiel, M.; Batenburg, K. J. Deep denoising for multi-dimensional synchrotron X-ray tomography without high-quality reference data. *Sci. Rep.* **2021**, *11*, No. 11895.

(8) Shorten, C.; Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, No. 60.

(9) Su, D. Advanced electron microscopy characterization of nanomaterials for catalysis. *Green Energy Environ.* **2017**, *2*, 70−83.

(10) Schneider, C. A.; Rasband, W. S.; Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **2012**, *9*, 671−675.

(11) Phromsuwan, U.; Sirisathitkul, C.; Sirisathitkul, Y.; Uyyanonvara, B.; Muneesawang, P. Application of image processing to determine size distribution of magnetic nanoparticles. *J. Mag.* **2013**, *18*, 311−316.

(12) Nicholson, W. V.; Glaeser, R. M. Automatic particle detection in electron microscopy. *J. Struct. Biol.* **2001**, *133*, 90−101.

(13) Ranzato, M.; Taylor, P.; House, J. M.; Flagan, R.; LeCun, Y.; Perona, P. Automatic recognition of biological particles in microscopic images. *Pattern Recognit. Lett.* **2007**, *28*, 31−39.

(14) Lyu, Z.; Yao, L.; Chen, W.; Kalutantirige, F. C.; Chen, Q. Electron Microscopy Studies of Soft Nanomaterials. *Chem. Rev.* **2023**, *123*, 4051−4145.

(15) Ronneberger, O.; Fischer, P.; Brox, T. In *U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing*

and Computer-Assisted Intervention--MICCAI 2015, 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III18, 2015; pp 234−241.

(16) Schmidt, U.; Weigert, M.; Broaddus, C.; Myers, G. Cell Detection with Star-Convex Polygons. *Med. Imgage Comput. Comput. Assisted Interventions − MICCAI* **2018**, 265−273.

(17) Colliard-Granero, A.; Batool, M.; Jankovic, J.; Jitsev, J.; Eikerling, M. H.; Malek, K.; Eslamibidgoli, M. J. Deep learning for the automation of particle analysis in catalyst layers for polymer electrolyte fuel cells. *Nanoscale* **2021**, *14*, 10−18.

(18) Yao, L.; Ou, Z.; Luo, B.; Xu, C.; Chen, Q. Machine Learning to Reveal Nanoparticle Dynamics from Liquid-Phase TEM Videos. *ACS Cent. Sci.* **2020**, *6*, 1421−1430.

(19) Mukaddem, K. T.; Beard, E. J.; Yildirim, B.; Cole, J. M. ImageDataExtractor: A Tool To Extract and Quantify Data from Microscopy Images. *J. Chem. Inf. Model.* **2020**, *60*, 2492−2509.

(20) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Proc. -Int. Conf. Neural Info. Process. Syst.* **2014**, *2*, 2672−2680.

(21) Zhang, Y.; Ling, H.; Gao, J.; Yin, K.; Lafleche, J.-F.; Barriuso, A.; Torralba, A.; Fidler, S. Datasetgan: Efficient labeled data factory with minimal human effort. *Proc. -IEEE/CVF Conf. Comput. Vision Pattern Recognit.* **2021**, 10145−10155.

(22) Dwibedi, D.; Misra, I.; Hebert, M. Cut, paste and learn: Surprisingly easy synthesis for instance detection. *Proc. -IEEE Int. Conf. Comput. Vision* **2017**, 1301−1310.

(23) Toda, Y.; Okura, F.; Ito, J.; Okada, S.; Kinoshita, T.; Tsuji, H.; Saisho, D. Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. *Commun. Biol.* **2020**, *3*, No. 173.

(24) Kharin, A. Y. Deep learning for scanning electron microscopy: Synthetic data for the nanoparticles detection. *Ultramicroscopy* **2020**, *219*, No. 113125.

(25) Polyanichenko, D. S.; Chernov, A. V.; Kartashov, O. O.; Alexandrov, A. A.; Butova, V. V.; Butakova, M. A. Intelligent Detection of the Nanomaterials Spatial Structure with Synthetic Electron Microscopy Images. *In 2022 XXV Int. Conf. Soft Comput. Meas. (SCM)* **2022**, 254−258.

(26) Malek, A.; Eslamibidgoli, M. J.; Mokhtari, M.; Wang, Q.; Eikerling, M. H.; Malek, K. Virtual materials intelligence for design and discovery of advanced electrocatalysts. *Chemphyschem* **2019**, *20*, 2946−2955.

(27) Yildirim, B.; Cole, J. M. Bayesian Particle Instance Segmentation for Electron Microscopy Image Quantification. *J. Chem. Inf. Model.* **2021**, *61*, 1136−1149.

(28) Chollet, F. Xception: Deep learning with depthwise separable convolutions. *Proc. - IEEE Conf., Comput. Vision Pattern Recognit.* **2017**, 1251−1258.

(29) Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. *Proc. - IEEE Conf., Comput. Vision Pattern Recognit.* **2009**, 248−255.

(30) Shorten, C.; Khoshgoftaar, T. M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60.

(31) Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. *Proc. - IEEE/CVF Conf., Comput. Vision Pattern Recognit.* **2022**, 11976−11986.

(32) Cubuk, E. D.; Zoph, B.; Shlens, J.; Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. *Proc. - IEEE/CVF Conf., Comput. Vision Pattern Recognit. workshops* **2020**, 702−703.

(33) Zhang, H.; Cisse, M.; Dauphin, Y. N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* preprint arXiv:1710.09412, 25/10/2017 (accessed 12/07/2023).

(34) Krause, D. JUWELS: Modular Tier-0/1 supercomputer at the Jülich supercomputing centre. *J. Large-Scale Res. Facil.* **2019**, *5*, 135.