JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

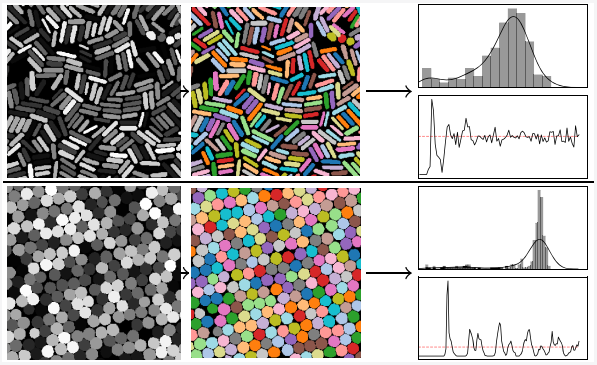
Bayesian Particle Instance Segmentation for Electron Microscopy Image Quantification

Batuhan Yildirim and Jacqueline M. Cole*

Cite This: J. Chem. Inf. Model. 2021, 61, 1136–1149 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Automating the analysis portion of materials characterization by electron microscopy (EM) has the potential to accelerate the process of scientific discovery. To this end, we present a Bayesian deep-learning model for semantic segmentation and localization of particle instances in EM images. These segmentations can subsequently be used to compute quantitative measures such as particle-size distributions, radial- distribution functions, average sizes, and aspect ratios of the particles in an image. Moreover, by making use of the epistemic uncertainty of our model, we obtain uncertainty estimates of its outputs and use these to filter out false-positive predictions and hence produce more accurate quantitative measures. We incorporate our method into the ImageDataExtractor package, as ImageDataExtractor 2.0, which affords a full pipeline to automatically extract particle information for large-scale data-driven materials discovery. Finally, we present and make publicly available the Electron Microscopy Particle Segmentation (EMPS) data set. This is the first human-labeled particle instance segmentation data set, consisting of 465 EM images and their corresponding semantic instance segmentation maps.

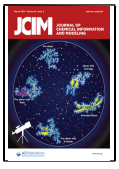


INTRODUCTION

The ability to automate analysis during characterization by electron microscopy (EM) is a desirable endeavor due to its capability to speed up particle analysis, as well as having the potential to collect data from EM images on a large scale. Measuring particle sizes from EM images has traditionally been performed using image-processing-based algorithms and this has proven an effective approach when doing so for a handful of images. However, the need to tune parameters to obtain accurate segmentations of particles in individual images makes this a highly manual approach on the small scale and a frail approach on the large scale. Learning-based segmentation methods that perform well in a variety of cases (given enough training data) have taken off in the past decade with the democratization of machine-learning methods in science. The robustness of these methods is well suited to automating data extraction and particle analysis from EM images, given that the need to tune parameters for edge cases can be eliminated by training on a sufficiently large and diverse set of labeled examples.

ImageDataExtractor[1] was the first of its kind to introduce a pipeline to extract quantitative particle measures from EM images in a high-throughput manner. Its authors used a series of image-processing methods such as thresholding, contour detection, and ellipse fitting to detect and locate particles in electron micrographs, subsequently performing particle analysis on the identified particles. They were able to automate the entire process, from the extraction of EM images from scientific literature using ChemDataExtractor,[2] to the measuring of scalebars in these images, achieving accurate particle and scalebar measurements. Similar to this work is that by Hiszpanski et al.,[3] where image-processing techniques were employed to extract morphology and particle size information of nanoparticles from scanning electron microscopy (SEM) and transmission electron microscopy (TEM) images. This was part of a larger project in which nanomaterial synthesis procedures and information were extracted automatically from text and images available in the published scientific literature. Other examples of purely image-processing-based methods include work by Groom et al.,[4] Meng et al.,[5] and as well as Mirzaei and Rafsanjani.[6] The former employs a thresholding-based segmentation approach, while the latter two use Hough transforms to segment particles following a series of preprocessing steps. Kim et al. measure particles in SEM images using a neural network to predict particle morphologies, followed by an application of the watershed segmentation

Received: December 18, 2020
Published: March 8, 2021



https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136–1149

algorithm to segment individual particles.[7] Like ImageDataExtractor, they automatically detect the scalebar to obtain the pixel-to-nanometer conversion. They then measure particles by the length of line segments passing through the centers of each particle. Slightly different, but related work, by Tatum et al.[8] is the Python library m2py, which performs semantic segmentation on scanning probe microscopy images, using a Gaussian mixture model (GMM) to identify a user-specified number of phases in the image. Once pixels belonging to the background and those belonging to particles/phases have been differentiated, m2py then performs instance segmentation using connected components labeling (CCL) or persistence watershed segmentation (PWS). They also show that instance segmentation can be performed directly using PWS, although by default this requires a single-channel image, meaning some form of dimensionality reduction is necessary for multichannel images. A commonality shared by these works (excluding m2py PWS and Kim et al.[7]) is that they perform a series of several steps to achieve segmentation. The drawback in this approach is that errors from the individual steps can accumulate to afford poorer results. Zhang et al.[9] perform instance segmentation in a single step and measure particles from the resulting segmentations. They employed a deep-learning approach to segment particles using a Mask R-CNN[10] instance-segmentation model, from which they were able to measure particles by performing edge fitting on the predicted segmentation maps. However, their model was trained on a small number of training examples (160), and their method is limited to spherical nanoparticles from TEM images, thereby lacking the diversity to be applied in a general sense for high-throughput use. Frei and Kruis[11] also employ a Mask R-CNN model to segment particles for analysis, although their work differs in that they do so for agglomerated/partially sintered particles. Due to the difficulty in manually labeling data of such occluded particles, they train their model on synthetic data and show that their method performs better than standard Hough transform-based segmentation and ImageJ.[12] Finally, Wu et al.[13] also utilize Mask R-CNN to perform particle analysis in similar ways.

In this work, we present a Bayesian deep representation learning model for semantic particle instance segmentation and show that the method is effective in accurately segmenting individual particles. Like Zhang et al.,[9] Frei and Kruis,[11] and Wu et al.,[13] we perform instance segmentation in a single step, thereby side stepping the problem of error accumulation. We train our model on the Electron Microscopy Particle Segmentation (EMPS) data set, a diverse collection of 465 electron micrographs and their corresponding human-labeled particle instance segmentation maps. While 465 samples may seem small as a training set for a deep-learning model, we show with model capacity experiments that we begin to reach performance plateaus with $n < 465$ training examples due to data augmentation. We show that our method performs significantly better than the previous image-processing approach, as well as the Mask R-CNN approach. We demonstrate the capability of our method to be used to perform particle analysis, by computing particle-size statistics, histograms, and radial- distribution functions from predicted segmentations. Finally, the method is integrated into ImageDataExtractor—a Python library for performing large-scale particle analysis on EM images.

SYSTEM OVERVIEW

Particle Instance Segmentation. *Basis of Model Formulation for Instance Segmentation.* Modern representation learning-based instance-segmentation methods can be classified into two categories: proposal-based and proposal-free. The former consists of those methods where region proposals in the form of bounding boxes are produced from an input image, with segmentation subsequently being performed in these proposed regions.[15−18] Since its inception, Mask R-CNN[10] has been the prevailing deep-learning-based instance-segmentation method that falls into the proposal-based category. Based on a detect-and-segment approach, objects are first detected using a region proposal network (RPN), which proposes several regions of interest (RoI) to potentially be segmented. Then, for each proposed RoI, Mask R-CNN produces a binary segmentation mask, alongside bounding box offsets and object classifications. This approach proved highly effective and Mask R-CNN continues to be the dominant approach in instance segmentation. Although a powerful method, de Brabandere et al.[19] highlighted the shortcomings of Mask R-CNN in cases where the objects being segmented overlap significantly. They argued, using a synthetic data set, that proposal-based methods fail to correctly segment overlapping instances in these cases, due to highly overlapping region proposal bounding boxes. Considering this evidence, along with the fact that highly overlapping nanoparticles are expected to be observed in EM images, this work takes a proposal-free approach for the task of particle instance segmentation.

In proposal-free methods,[19−24] every pixel in the input image is mapped to a pixel embedding, such that pixels belonging to the same object/instance are encouraged to be close together in pixel-embedding space, while pixels belonging to different instances are pushed apart. This is usually achieved by designing a loss function that encourages grouping/clustering of pixels based on the $x-y$ location and learned feature vectors of each pixel. Until recently, these methods suffered from their need to perform clustering using a separate algorithm during test time. This is disadvantageous as this can be slow on high-resolution images, and incorporating clustering into the loss function is likely to benefit the learned features of such a model for segmentation tasks. Neven et al.[20] proposed a solution to this problem by outputting a seed map alongside pixel embeddings. Pixels with high seed scores in the seed map exist close to their instance centroid in embedding space, while those with lower scores are further away. Using this seed map, it is possible at test time to identify potential instance centroids and assign pixels to instances by determining to which centroid their embeddings lay closest.

This work adopts the semantic instance segmentation methodology of Neven et al.[20] and recasts it in the framework of Bayesian deep learning. We set out to learn a function $f(X; \theta)$ that takes as input an EM image containing a set of particles and maps each pixel to an offset vector o_i, from which we obtain their pixel embeddings e_i. Each o_i is a two-dimensional (2D) vector that should point to the centroid of the particle instance, S_k, to which the pixel belongs. Pixel embeddings are subsequently obtained by adding this offset vector to the 2D coordinate of each pixel in the input image $e_i = o_i + x_i$, where x_i is the coordinate. By outputting offset vectors instead of embeddings directly, we avoid spatial equivariance by encoding the location of each pixel in the input image, thereby allowing

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

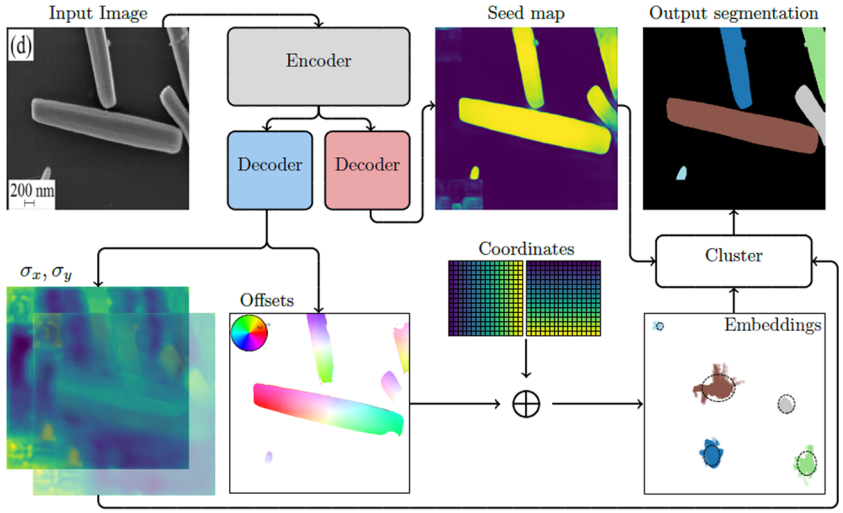


Figure 1. Pipeline for segmenting particle instances in EM images. An EM image is passed as input to the Bayesian particle instance segmentation (BPartIS) encoder to produce a latent representation of the input image. Standard deviations and offset vectors for each pixel are produced from this latent representation by the first decoder. Offset vectors are converted to spatially dependent pixel embeddings by adding the 2D coordinates of each pixel to each offset. The second decoder transforms the latent representation into a seed map, denoting which pixels are likely to be the centroid embeddings of each particle instance. The embeddings, standard deviations, and seed map are all used to cluster pixel embeddings to afford an output instance-segmentation map. The example used is an SEM image of ZnO microrods by Sarma and Sarma[14] reprinted from ref 14, Copyright (2017), with permission from Elsevier.

the model to learn position-relative embeddings. This is crucial, since spatial-equivariant embeddings would mean that pixels belonging to two distinct particles that are identical in appearance but exist in different parts of the image would have the same e_i values. As a result, we would not be able to assign the pixels belonging to these particles to two separate instances. Hence, our function outputs offset vectors for each pixel, which we transform into spatially dependent pixel embeddings. This process is illustrated in Figure 1.

When learning our function, we desire that the mapping carried out by f will have the following properties. In pixel-embedding space, pixels belonging to the same particle instance should be close together. Additionally, clusters of pixels belonging to different particle instances should be well separated from each other. We model this function as a convolutional neural network (CNN) and learn its parameters θ using a supervised loss function, which encourages clustering of pixel embeddings that belong to the same class (particle instance). At test time, we subsequently use this function to compute the pixel embeddings of an input image and cluster them to assign each pixel to a particle instance (or background). We now describe the loss function used to learn θ and how this function achieves clustering of particle-instance pixel embeddings. The design choices of the model architecture that enable the use of this loss function are also outlined.

Learning to Segment Particle Instances. For each particle instance, S_k, in the set of instances, S, we desire that each $e_i \in S_k$ lay close to the instance centroid of S_k. This way, we can assign pixels to particles in the input image based on to which instance centroid each pixel embedding is closest. During training, instance centroids, μ_k, are computed by taking the mean of the embeddings that belong to an instance.

$$\mu_k = \frac{1}{|S_k|} \sum_{e_j \in S_k} e_j \tag{1}$$

Hence, it would be possible to regress each embedding toward the centroid of the instance to which it belongs. In practice, however, minimizing a quadratic penalty to achieve this does not work so well when the objects being segmented vary in size. Larger particles are naturally comprised of a larger number of pixels in EM images, and it is likely that many of these pixels will exist far from the instance centroid in pixel-embedding space. These further pixel embeddings will dominate a loss function based on squared distance, leading to inferior performance.

Instead, this distance is converted into the probability of belonging to an instance, by placing an elliptical Gaussian kernel over the distance, $\phi_k(e_i)$

$$\phi_k(e_i) = \exp\left(-\frac{\|e_{ix} - \mu_{kx}\|^2}{2\sigma_{kx}^2} - \frac{\|e_{iy} - \mu_{ky}\|^2}{2\sigma_{ky}^2}\right) \tag{2}$$

By letting our model output standard deviations, σ_k, for each instance, larger σ_k values can be assigned to larger particle instances, meaning we overcome the previously mentioned issue of large particles dominating the loss function. Architecturally, we achieve this by outputting a σ_i for each pixel. We subsequently obtain σ_k by averaging each σ_i belonging to instance S_k. During training, we encourage all σ_i values that belong to an instance to be similar to each other by including a smoothness term in our loss function. Finally, to determine the assignment of pixels to particle instances, we assign e_i to instance S_k if $\phi_k(e_i) > 0.5$, and to the background or other instances otherwise.

Before defining the loss functions used to learn θ, there is an important architectural feature of the model that must be outlined. During training, we are able to obtain the centroids μ_k of each instance, since we train our model (supervised) using labeled data. We do so using the binary ground-truth segmentation maps as masks to obtain all pixel embeddings belonging to an instance, and average them. During test time, however, we do not have these centroids, and we must obtain

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

them to cluster around each μ_k. To achieve this, our model outputs a seed map (alongside offset vectors and sigmas), as described in the previous subsection and according to Neven et al.[20] Pixels with a high seed score in the seed map lay close to the instance centroid, while pixels with lower seed scores are further away. This way, at test time, we identify the pixel embeddings with the highest seed scores and use these as proxies for our centroids, $\hat{\mu}_k$. Then, for each $\hat{\mu}_k$, we can find all pixels whose $\phi_k(e_i) > 0.5$ and assign these to instance S_k. We train our model to output the seed map such that background pixels are regressed to zero, while foreground pixels (those belonging to particles) are regressed to the output of the elliptical Gaussian function $\phi_k(e_i)$.

The loss function has three terms (and three weighting coefficients λ_i) and is defined as follows

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{\text{Lovász}} + \lambda_2 \cdot \mathcal{L}_{\text{seed}} + \lambda_3 \cdot \mathcal{L}_{\text{smooth}} \tag{3}$$

The first term is the Lovász-hinge loss,[25] and its formulation is given by Berman and Blaschko.[26] Minimizing this term directly optimizes the mean intersection-over-union (IoU) between the predicted and ground-truth segmentation masks of the particle instances. This encourages pixel embeddings to be close to their corresponding instance centroids in pixel-embedding space. We minimize the Lovász-hinge loss between the predicted $\phi_k(e_i)$ and ground-truth instance masks for each instance. The second term is the seed map loss

$$\mathcal{L}_{\text{seed}} = \frac{1}{N} \sum_{i}^{N} \mathbf{1}_{\{s_i \in S_k\}} \, \|s_i - \phi_k(e_i)\|^2 + \mathbf{1}_{\{s_i \in \text{bg}\}} \, \|s_i - 0\|^2 \tag{4}$$

where s_i are the seed-map values and the indicator functions denote binary masks for the foreground (particles) and background. This term regresses all background seed-map values to 0, and all foreground seed-map values to $\phi_k(e_i)$. Finally, the smoothness term encourages all $\sigma_i \in \sigma_k$ values to be similar to each other

$$\mathcal{L}_{\text{smooth}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{\sigma_i \in S_k} \|\sigma_i - \sigma_k\|^2 \tag{5}$$

Bayesian Inference. *Basis of Bayesian Deep Learning Model Formulation.* While deep neural networks are typically optimized to learn point estimates of their parameters, in cases where it is necessary to reason about uncertainty, it is favorable to take the Bayesian approach and instead learn distributions over the parameters of such models.[27−29] In this paradigm, a prior distribution $p(\theta)$ is placed over the weights of the network. Given a data set with inputs $\mathbf{x} \in \mathbf{X}$ and targets $y \in \mathbf{Y}$, we seek to obtain the posterior distribution over the parameters using Bayes' theorem

$$p(\theta|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta)}{p(\mathbf{Y}|\mathbf{X})} \tag{6}$$

where $p(\mathbf{Y}|\mathbf{X},\theta)$ is the likelihood function and $p(\mathbf{Y}|\mathbf{X})$ is the marginal likelihood or normalizing distribution. If we were able to solve eq 6, we could obtain this posterior and perform inference by marginalizing over the parameter distribution

$$p(y^*|\mathbf{x}^*) = \int_\theta p(y^*|\mathbf{x}^*, \theta)p(\theta|\mathbf{X}, \mathbf{Y}) \, \mathrm{d}\theta \tag{7}$$

This amounts to evaluating a weighted average of the likelihood function for all possible θ, each weighted by its plausibility $p(\theta|\mathbf{X},\mathbf{Y})$.

In many cases, however, including the context of Bayesian neural networks (BNNs), performing inference this way is not possible, as the true posterior over the parameters cannot be evaluated analytically. Gal and Ghahramani[30] proposed using variational inference to approximate this posterior, where an approximating variational distribution $q(\theta)$ is employed to estimate the true posterior. The form of $q(\theta)$ proposed by the authors is a Bernoulli distribution, which they showed can be conveniently modeled using dropout,[31] a technique commonly used for regularization in neural networks. By minimizing the Kullback−Leibler (KL) divergence between $q(\theta)$ and $p(\theta|\mathbf{X},\mathbf{Y})$, the approximating distribution is encouraged to closely resemble the posterior. In practice, the KL divergence is implicitly minimized, since finding the optimal parameters of a dropout neural network is equivalent to finding the optimal variational parameters $\theta_i \sim q(\theta)$.[29] By training a neural network with dropout, we can perform inference by approximating the marginal over the variational distribution using Monte Carlo (MC) sampling. This is achieved by first activating dropout at test time and performing T forward passes on the same input to obtain T MC samples. The average of these samples yields the MC estimate

$$\begin{aligned} \mathbb{E}_{p(\theta|\mathcal{D})}[f(x;\, \theta)] &= \int_\theta f(x;\, \theta)p(\theta|\mathcal{D}) \, \mathrm{d}\theta \\ &\approx \int_\theta f(x;\, \theta)q(\theta) \, \mathrm{d}\theta \\ &\approx \frac{1}{T} \sum_{i=1}^{T} f(x;\, \theta_i), \;\; \theta_i \sim q(\theta) \end{aligned} \tag{8}$$

where $f(x;\theta)$, the Bayesian neural network, is the likelihood function. The result of this is our MC estimate $p_{\text{MC}}(y|x)$. Thus, by implicitly minimizing the KL divergence between the posterior and the approximating distribution during training, it is possible to perform inference using a MC estimator to obtain predictions at test time.

Method Application of Particle Segmentation by Bayesian Inference. Once our model is trained, we predict particle instance segmentation maps from unseen EM images as follows. First, to perform inference, we output T MC samples by passing the same input through our model T times with dropout turned on, each time with different parameters due to the randomness induced by dropout. An individual MC sample (the output of a single MC estimator) consists of offset vectors (which we convert into pixel embeddings e_i) and sigmas for each pixel in the input image, as well as a seed map denoting which pixels are likely to belong to particles. We obtain the final MC estimates from these outputs by averaging each component from the individual MC samples, yielding the MC estimates of embedding, sigma, and seed maps. Finally, we use these to cluster pixels and obtain the final instance-segmentation maps.

To cluster pixels, we obtain the maximum value in the MC seed map to find the pixel that is most likely to be the centroid embedding μ_k of the first particle instance. Once we know the location of this maximal value, we obtain the σ_k value of its instance from the MC sigma map. Armed with μ_k and σ_k, we compute $\phi_k(e_i)$ for each pixel embedding, as in eq 2, and take all cases where $\phi_k(e_i) > 0.5$ to be belonging to the first particle

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

instance. We then obtain a binary mask of all pixels belonging to the first instance and use this to set all seed values in the seed map of this instance to 0. We can then find the next maximal seed-map value belonging to the next particle instance and repeat this until all particle instances have been identified.

Quantifying Uncertainty. *Statistical Basis of Uncertainty Quantification.* The Bayesian approach has the advantage that it allows us to measure the uncertainties of the estimates of a model. The ability to measure the uncertainty of machine-learning-based operational pipelines can be crucial in informing the decisions that these systems make, as well as improving the quality of predictions. In this work, we make use of the uncertainties during the particle instance segmentation process, given that our end goal is to compute quantitative measures. Uncertain segmentations in our output segmentation maps are treated accordingly, such that they do not affect the quality of the quantitative measures that we compute. Specifically, any artifacts of the background, scalebars, or subpart labels that are incorrectly segmented as particles generally have high uncertainty and can easily be discarded and ignored in further computations.

Uncertainty can be distinguished into two distinct types: aleatoric and epistemic.[32] Aleatoric uncertainty refers to the intrinsic noise in observations/data. In the case of human-labeled segmentation maps, a manifestation of this uncertainty would be due to errors in the edges of objects, where a few pixels of the background may have been included in the segmentation mask of an object due to human error. This type of uncertainty could theoretically be reduced if data labeling/collection could be done perfectly. However, this is generally not possible, and reduction in aleatoric uncertainty cannot be achieved by collecting more data. Epistemic uncertainty can be viewed as a model's lack of knowledge, where its parameters are not well constrained to deal with out-of-distribution inputs. In these regions of input space, the posterior over the parameters is broad, resulting in high epistemic uncertainty. When performing Bayesian inference, this property of epistemic uncertainty allows one to discern whether the inputs provided to a model are out-of-distribution (with respect to the data the model was trained on), since such inputs will produce high epistemic uncertainties. This property can be used for outlier detection[33] or may serve as an indication that the model would benefit from being trained with more data, since unlike aleatoric uncertainty, epistemic uncertainty can be reduced given more data.

Several methods have been proposed in the literature to compute uncertainty with Bayesian neural networks, which perform inference using MC sampling by dropout. One of these methods simply uses the entropy of the predictive distribution $H[p(y|\mathbf{x})]$.[34] Another simple method is to use the variance of the MC samples as a form of uncertainty.[30,35] In this case, the idea is that the MC estimate of a model with low uncertainty will have low variance, and the model will confidently predict the same output more often than not, given an input and some paramaters sampled from an approximating distribution. Conversely, if the model is not paramaterized well to deal with an input, the individual MC samples will vary drastically and the variance will be high. While both straightforward to compute, the entropy of the predictive distribution does not distinguish model uncertainty from data uncertainty, while the variance is an approximation of model uncertainty. A more sophisticated method, and the one which we used in this work, is uncertainty based on the conditional mutual information (MI) between the output and the parameters.[36,37] MI is a measure of the reduction in uncertainty in a random variable, after gaining knowledge of another. In the case of $\mathrm{MI}(y; \theta|\mathbf{x})$, it measures the reduction in uncertainty of the parameters θ, given that we observe y; a high MI in this case indicates that knowing the true value of y would result in a large reduction in uncertainty, meaning that there is high uncertainty in θ at the onset. We express MI as the difference between: (1) the predictive entropy (total uncertainty) of the MC estimate, which acts as a proxy for the uncertainty in the marginal estimate of the model; and (2) the expected entropy (data uncertainty) that describes the average uncertainty of the individual MC samples, which make up the MC estimate.

$$\underbrace{\mathrm{MI}(y;\theta|\mathbf{x})}_{\text{epistemic uncertainty}} = \overbrace{\underbrace{\mathrm{H}[p_{MC}(y|\mathbf{x})]}_{\text{total uncertainty}}}^{\text{predictive entropy}} - \overbrace{\underbrace{\mathbb{E}_{\theta\sim p(\theta|\mathcal{D})}\mathrm{H}[p(y|\mathbf{x},\theta)]}_{\text{aleatoric uncertainty}}}^{\text{expected entropy}} \tag{9}$$

This way, we decompose the uncertainty into its epistemic and aleatoric components.[38,39] In the case where the predictive entropy is similar to the expected entropy, the MI (and hence the epistemic uncertainty) is low, and we know that most of the uncertainty is aleatoric (due to data). In contrast, if the predictive entropy is high while the expected entropy is low, we know that epistemic uncertainty dominates the total uncertainty.

Method Application of Uncertainty Quantification. We are interested in modeling when an input lies in a region of data space that the parameters are not conditioned to deal with. Since data uncertainty is inevitable, we would like to explicitly model this and remove its contribution to the total uncertainty to obtain the model uncertainty. Smith and Gal[37] showed that MI of y and θ is a measure of a model's epistemic uncertainty, since being uncertain about an input x implies that if the model knew the true label y at that point, then information would be gained. In contrast, if our model parameters are well-conditioned to deal with x, we would gain little information in learning the true value of y. We therefore model the epistemic uncertainty in our outputs using MI, expressed in eq 9. By activating dropout at test time, we use the dropout approximation to obtain an MC estimate of the predictive distribution given an input $\mathbf{x}$

$$p_{\mathrm{MC}}(y|\mathbf{x}) \approx p_{\theta\sim p(\theta|\mathcal{D})}(y|\mathbf{x}) \approx \frac{1}{T}\sum_{i=1}^{T} p_{\theta_i\sim q(\theta)}(y|\mathbf{x},\theta_i) \tag{10}$$

where $p_{\theta_i\sim q(\theta)}(y|\mathbf{x},\theta_i)$ is a single MC sample—the output of a single MC estimator (our model with some weights randomly turned off by dropout). The entropy of $p_{\mathrm{MC}}(y|\mathbf{x})$ gives us the predictive entropy (total uncertainty). We compute the expected entropy (aleatoric uncertainty) by taking the mean of the entropies of the individual MC samples. Putting these two together, we approximate the MI and hence the epistemic uncertainty by

$$\mathrm{MI}(y;\theta|\mathbf{x}) \approx H[p_{\mathrm{MC}}(y|\mathbf{x})] - \frac{1}{T}\sum_{i=1}^{T} H[p_{\theta_i\sim q(\theta)}(y|\mathbf{x},\theta_i)] \tag{11}$$

Recall that for each input image, our method outputs a seed map (alongside offset vectors and sigmas), which is an estimate of the probability $\phi_k(e_i)$ of each pixel belonging to a particle in the image. Thus, we can output T MC seed map samples and

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

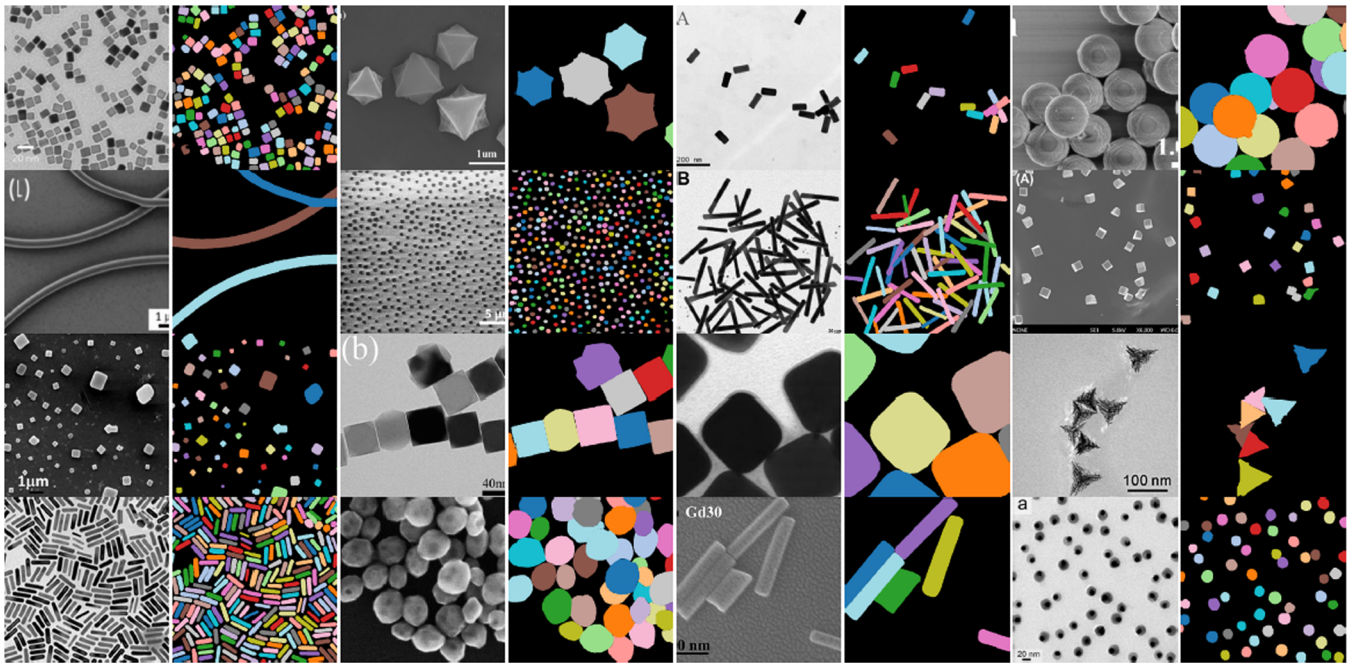


Figure 2. Sample images and corresponding instance-segmentation maps from the EMPS data set. Particle instances are denoted by the colored regions in the segmentation maps. Images going downwards then right are: Falcaro et al. reprinted from ref 51, Copyright (2016); Jiang et al. reprinted from ref 52, Copyright (2017); Navas and Soni reprinted from ref 53, Copyright (2016); Meng et al. reprinted from ref 54, Copyright (2017); Li et al. reprinted from ref 55, Copyright (2018); Balling et al. reprinted from ref 56, Copyright (2018); Yang et al. reprinted from ref 57, Copyright (2017); Distaso et al. reprinted from ref 58, Copyright (2017); He et al. reprinted from ref 59, Copyright (2019); Roy et al. reprinted from ref 60, Copyright (2017); Wu et al. reprinted from ref 61, Copyright (2020); Wu et al. reprinted from ref 62, Copyright (2017); Shang et al. reprinted from ref 63, Copyright (2020); Liu et al. reprinted from ref 64, Copyright (2017); Wang et al. reprinted from ref 65, Copyright (2017); and Wang et al. reprinted from ref 66, Copyright (2020). All with permission from Elsevier.

directly compute the predictive and expected entropies from these probabilities, followed by the conditional MI. This serves as the measure of epistemic uncertainty in our model outputs.

Uncertainty Filtering. We use the uncertainties yielded by Bayesian inference to significantly improve the predictions of our model by identifying potential false positives and removing them. We simply compute the MI as mentioned in the previous subsection and use this as a measure of the epistemic uncertainty of our model given some input. From our initial particle-instance predictions, we can use the segmentation mask of each instance individually to obtain all uncertainty values that belong to an instance. By averaging the uncertainties for a predicted instance, we obtain the total epistemic uncertainty of that prediction. If this uncertainty is greater than some threshold t_u, we label that prediction as a false positive and remove it from the final particle instance segmentation map. We use $t_u = 0.0125$ in this work for a model trained on 366 images from the EMPS training set. This value was found empirically—the details and results of this experiment can be found in the Results and Discussion section. The value of t_u will likely have to be tuned for models that are trained on greater or fewer data than this.

Reducing Overfitting. Deep neural networks typically require several thousands of training examples to be able to approximate a function well. Nevertheless, several methods can be employed to circumvent this limitation when fewer data are available for the task at hand. Since the EMPS data set consists of a relatively small number of training samples for the context of deep learning, it is necessary to explore methods to overcome this limitation. These include simpler methods such as regularization and data augmentation,[40] as well as more sophisticated methods such as transfer learning when sufficiently similar labeled data are available, and unsupervised pretraining when a large number of unlabeled data are available. The latter includes methods such as autoencoders trained to reconstruct their inputs,[41] contrastive predictive coding,[42,43] and information maximization.[44]

We found that data augmentation alone was enough to prevent overfitting, and more sophisticated methods were not necessary. When exploring the latter methods, we attempted unsupervised pretraining by image reconstruction (as well as data augmentation), due to its simplicity and to the availability of the SEM data set curated by Aversa et al.[45] The SEM data set is an ideal candidate for two reasons: it consists of a diverse set of around 22 000 SEM images (without segmentation labels), around the order of magnitude needed to effectively train deep neural networks for vision tasks; the objects and textures found in these data are highly likely to share common characteristics with what is found in the EMPS data set. We trained a convolutional autoencoder to reconstruct the images from the SEM data set and used the encoder and decoder from this model to initialize our segmentation model (also an encoder−decoder architecture). Ultimately, we found that a model trained with a combination of unsupervised pretraining and data augmentation performed ever so slightly worse than one trained with data augmentation alone.

Thus, we opted for data augmentation without unsupervised pretraining. Augmentations used were: horizontal flips; vertical flips; random rotations; random color jitters (brightness ±0.3, contrast ±0.3, saturation ±0.3, hue ±0.3); random crops of half the original image resolution which were scaled up to the original resolution using bicubic interpolation for images, and

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

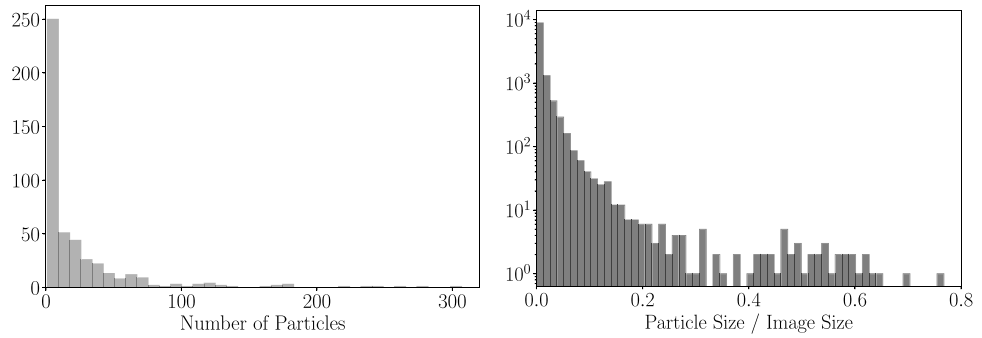


Figure 3. Image and particle-instance statistics from the EMPS data set. Left: number of particles per image. Right: particle-instance size as a percentage of image size (log *y*-scale).

nearest-neighbor interpolation for ground-truth segmentation maps.

Implementation Details. *Architecture.* The underlying architecture of our model is an ERFNet[46] with two decoders (one for offsets and sigmas, and one for seed maps) as per Neven et al.[20] Offset vectors output by the first decoder are bounded between $[-1, 1]$ using a tanh activation function, and the coordinate map added to this to obtain pixel embeddings ranges from $[0, 1]$ in both x and y dimensions. Instead of outputting σ values directly, we output $\log\left(\frac{1}{2\sigma^2}\right)$ and use an exponential function to obtain $\frac{1}{2\sigma^2}$. A sigmoid activation function is applied to the output of the second decoder to obtain seed maps, since it is trained to produce probabilities $\phi_k(e_i)$ for foreground (particle) pixels and 0 for background pixels. We slightly modified the originally devised ERFNet[46] in the following ways. We replaced all 2D transposed convolutions (deconvolutions) in the decoders with bilinear interpolation upsampling layers + standard 2D convolutions, since we initially observed checkerboard artifacts in our predictions as a result of the original transposed convolutions.[47] Additionally, we replaced all rectified linear unit (ReLU) activations[48] with exponential linear units (ELUs),[49] as we found that this slightly improved performance.

Training. We trained our model for 300 epochs using the Adam[50] variant of stochastic gradient descent with a learning rate of 0.0003. Weighting factors used in the loss function (as shown in eq 3) are $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 10$. The training was performed on an NVIDIA Tesla V100 graphics processing unit (GPU) in Google Colaboratory, where a single epoch (training and validation) took roughly 70 s.

Electron Microscopy Particle Segmentation (EMPS) Data Set. A bespoke Electron Microscopy Particle Segmentation (EMPS) data set was constructed to serve as the training data for this work. It consists of 465 electron micrographs and their corresponding human-labeled ground-truth semantic instance segmentation maps, as well as the coordinates of the polygons drawn around each particle to construct the segmentation maps. Figure 2 shows 16 sample images and their segmentation maps and portrays qualitatively the diversity of particle sizes, shapes, textures, densities, and (grayscale) colors that exist in the data set. Although not relevant for computing quantitative measures, we included several images where particles overlap each other with varying degrees of overlap, as this is common in the electron micrographs of nanoparticles. The third EM image in the second row of Figure 2 is an example of highly overlapping particles, while the particles in the fourth EM image of the third row show minor overlap.

All images in the EMPS data set were mined from published scientific literature using the data-mining application programming interface (API) of Elsevier. We first used the Article Retrieval API to obtain the digital object identifiers (DOIs) of articles published between the years 2015 and 2020, which had the possibility of containing EM images. This was achieved using the search query "SEM−TEM−scanning electron microscopy−transmission electron microscopy." Next, using the Object Retrieval API, we iterated through the figures in these articles and obtained images at high resolution from any figure which contained one or more of the acronyms or phrases from our search query. This resulted in 34 091 images of figures, from which 788 were manually determined as suitable and set aside for postprocessing. It was often the case that EM images were part of a panel of several images in these figures. Thus, the EM images were cropped from these 788 figures, resulting in 962 potential images to be labeled (many figures contained several relevant EM images). We annotated 465 of these images using the VGG Image Annotator (VIA).[67,68] This consisted of drawing polygons around each individual particle in each image. Once the annotation process was completed, we finally assigned pixels to particle instances in each image by finding all pixels that were encapsulated by the polygon of each particle.

Figure 3 presents some statistics of the images and particles in the EMPS data set. It is evident that most images contain fewer particles, and only a few images contain many particles (Figure 3, left). Similarly with particle sizes, most particles in the data set are small, with the number of large particles dropping significantly as particle size increases (Figure 3, right).

RESULTS AND DISCUSSION

Technical Validation. *Particle Instance Segmentation.* We evaluate the performance of our Bayesian particle instance segmentation model (hereafter known as BPartIS) using a number of metrics and compare it to several similar algorithms as benchmarks. These include ImageDataExtractor,[1] a Gaussian mixture model + connected component labeling[8] and Mask R-CNN[10] (with a ResNet-101 backbone[69]). To quantify the improvement in performance afforded by our Bayesian formulation, we also report the performance of the discriminative version of our method, as well as the Bayesian version for direct comparison. The EMPS data set was split into training and test sets consisting of 366 and 99 samples, respectively, where the former was used for training and

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

validation, and the latter was used to evaluate our model, as well as the benchmark algorithms. Since there are cases where several EM images from a single publication exist in the EMPS data, the data were split such that images from the same publication were never spread across both the training and test set to avoid data leakage, i.e., multiple images from the same publication only appear in the training or the test set, but not both. We used average precision (AP) as defined by Hariharan et al.[70] for our main evaluation metric. Given an input image, our model outputs a set of predictions for individual particle instances. We compare each of the individual predictions to all ground-truth particle instances. If the predicted particle instance segmentation mask and a ground-truth instance mask have an intersection-over-union (IoU) greater than some threshold t, we count this as a true positive. If a prediction has an IoU less than t with all ground-truth instance masks (there is no match), we count this as a false positive. Duplicates are defined as several predictions having an IoU greater than t for a single ground-truth instance. In this case, we designate the prediction with the highest IoU as a true positive, and the rest as false positives. Finally, a false negative is defined as any ground-truth instance which has not been matched with any predicted instance. We created a precision−recall curve by computing precision and recall at varying thresholds t and took the area under this curve as the average precision (AP). We also report AP_{50} and AP_{75}, which are simply the precision values at $t = 0.5$ and 0.75. AP_{75} indicates how well the model performs under stricter true positive conditions and rewards methods with better localization. In addition to AP, we also report the mean IoU for all predicted particles

$$\mathrm{IoU}(s_i, s'_j) = \begin{cases} \dfrac{s_i \cap s'_j}{s_i \cup s'_j}, & \text{if true positive} \\ 0, & \text{if false positive} \end{cases} \tag{12}$$

where s'_j is the segmentation mask of the jth predicted particle instance and s_i is the ith ground-truth segmentation mask.

The results in Table 1 show that BPartIS (Bayesian + filter) outperforms all of the benchmark methods on these metrics

Table 1. Comparing the Performance of BPartIS (Our Method) to Similar Algorithms

method	AP	AP_{50}	AP_{75}	mean IoU
ImageDataExtractor	0.327	0.320	0.189	0.249
GMM + CCL	0.411	0.289	0.215	0.236
Mask R-CNN	0.506	0.668	0.638	0.621
BPartIS (discriminative)	0.560	0.786	0.738	0.712
BPartIS (Bayesian)	0.590	0.823	0.771	0.745
BPartIS (Bayesian + filter)	0.632	0.928	0.874	0.844

and significantly outperforms the image-processing and non-deep-learning-based particle segmentation methods. While all recent EM image quantification methods that employ learning-based models for segmentation use Mask R-CNN, these results suggest that proposal-free methods such as ours have greater potential for the task of particle instance segmentation. This is due to the inductive biases of proposal-free methods (pixel-wise clustering in a latent space) being more suitable for the domain of electron microscopy images, where objects can appear densely packed and overlapping. The results show that BPartIS (Bayesian), which performs prediction by Bayesian inference (without uncertainty filtering), results in a slight improvement over BPartIS (discriminative). This can be attributed to the fact that Bayesian inference does two things which result in improved performance: (a) a Bayesian model marginalizes over a posterior parameter distribution in which larger probability is assigned to more probable parameters given the observed data; (b) Bayesian models are ensemble methods; statistical ensembles are well-established as methods for improving prediction accuracy compared to single-estimator models. The significant increase in performance afforded by the Bayesian formulation combined with uncertainty filtering (BPartIS (Bayesian + filter)) is shown quantitatively in Table 1, and qualitatively in Figure 4, where the increase in precision is visually evident. Initially, regions such as scalebars and background textures are incorrectly segmented by BPartIS as particles (column 3 in Figure 4) but with high uncertainty (column 4 in Figure 4). Using the uncertainties of the individual predicted particle instances, we are able to filter out these false positives by their high uncertainties and achieve much higher AP, AP_{50}, AP_{75}, and mean IoU scores than the discriminative version (as well as all other benchmarks). The improvement in AP afforded by BPartIS (Bayesian + filter) relative to Mask R-CNN is evident in Figure 5, where the false positives predicted by Mask R-CNN do not appear in the segmentation maps predicted by BPartIS (Bayesian + filter). The results in Table 1 and Figures 4 and 5 suggest that the most significant gains in performance can be attributed to the ability of BPartIS (Bayesian + filter) to filter out false positives using uncertainties.

Analysis of the Uncertainty Threshold. We analyzed the effect of changing the uncertainty threshold on the performance of BPartIS (Bayesian + filter). To do so, we performed a gridsearch over 100 t_u values and examined the performance of the model on the AP, AP_{50}, AP_{75}, and mean IoU metrics. The results are shown in Figure 6. When the uncertainty threshold is low, all metrics become zero. Since BPartIS is unable to predict the segmentation mask of a particle with complete certainty, all predicted segmentations have some nonzero uncertainty, resulting in all particles being filtered out as false positives at low threshold values. This baseline uncertainty was found to be 7.8×10^{-4}. Increasing t_u beyond this resulted in a steady increase in all metrics with AP peaking at $t_u = 0.01244$ and the other metrics peaking around $t_u = 0.008$ (t_u ranges between $[0.0, \ln 2]$). Beyond these peaks, the performance drops slightly until around $t_u = 0.09$, leveling off as t_u increases beyond this for all metrics. At these postpeak values of t_u, the drop in performance can be attributed to particles needing to have relatively high uncertainties to be filtered out, resulting in a higher false-positive rate. As a result of these findings, the default uncertainty threshold was selected to be $t_u = 0.0125$.

Model Capacity Experiments. We investigated the capacity of BPartIS to improve as a function of the number of training examples. This is of interest due to the relatively small size of the EMPS data set. We trained several models with data augmentation using $N \in \{50, 100, 150, 200, 250, 300, 366\}$ training examples in each case and evaluated each model on the same test set of 99 images. Figure 7 shows the performance of each model, where we initially observe a steady increase in performance by increasing the number of training samples. However, the increase in performance begins to level off after 200/250 training samples, and increasing the size of the training set beyond this does not improve performance

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

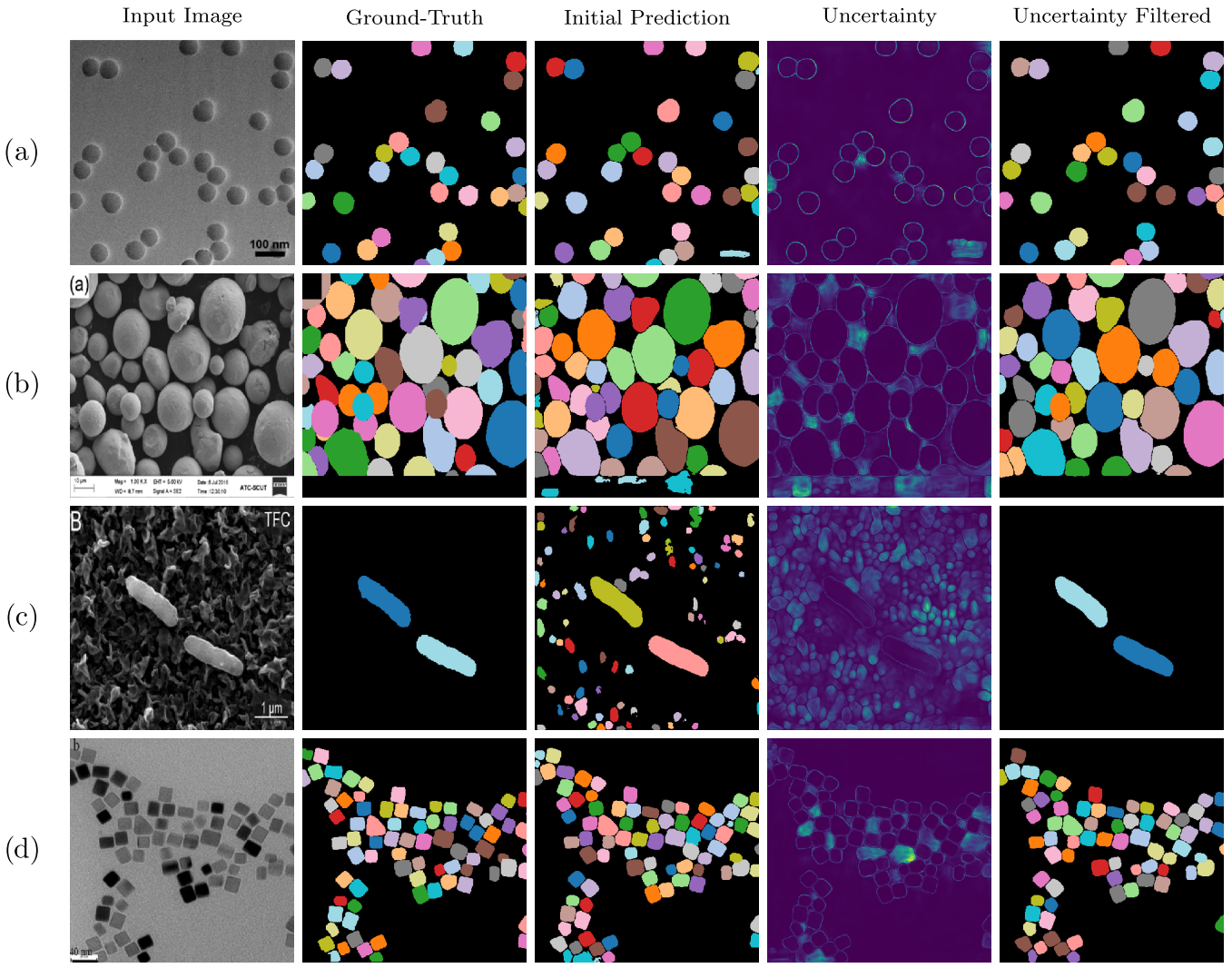


Figure 4. Qualitative results of performing Bayesian inference and uncertainty filtering with BPartIS on four examples from the EMPS test set. Predicted instance-segmentation maps and their corresponding uncertainty maps are shown, as well as the uncertainty-filtered final output. Notice how regions such as scalebars, text, and background textures are initially identified as particles with high uncertainty. These are subsequently removed to produce the uncertainty-filtered output, by removing all predicted instances with an uncertainty above some threshold t_u. (a) TEM of functionalized silica nanoparticles by Sun et al.[71] reprinted from ref 71, Copyright (2019); (b) SEM of grade 300 maraging steel powders by Tan et al.[72] reprinted from ref 72, Copyright (2017); (c) SEM of bacterial cells by Faria et al.[73] reprinted from ref 73, Copyright (2017); and (d) TEM of Pd cubic nanoparticles by Shah et al.[74] reprinted from ref 74, Copyright (2017). All with permission from Elsevier.

significantly on the test set. These empirical results suggest that the use of data augmentation has allowed us to train an effective particle instance segmentation model with a relatively small number of training examples, and increasing the number of training samples further is unlikely to increase the performance enough to warrant further time and labor-intensive manual data labeling.

Demonstration of Model in Automatic Particle Analysis. Our method can afford accurate and precise particle segmentation in an automated operational pipeline that measures and quantifies particles from EM images. Quantitative measurements that can be extracted from such a pipeline include average particle size, particle-size distributions, aspect ratios, and in some cases radial-distribution functions. As such, we present two case studies in which we employ BPartIS on EM images not present in the training or test sets and produce relevant quantitative measures of output. Although many works automate scalebar measurements to convert pixels into the relevant units (nm, μm), for the purpose of this demonstration, we measure them manually since in this work we are evaluating the particle segmentation capability of an automated particle measurement system. Besides, the strength of the BPartIS model is such that it has been incorporated into the ImageDataExtractor[1] Python library, replacing the existing particle-detection and quantitative-analysis modules in v1.0, to form ImageDataExtractor v2.0. Therein, it uses the original automated scalebar-measurement modules. The particle analysis is illustrated in Figure 8. First, an input image is passed through the BPartIS model to obtain the initial instance-segmentation predictions and uncertainty map. The predictions are then postprocessed by uncertainty filtering to remove false-positive regions that may have incorrectly been classified as particles. Since the aim is to accurately measure particles, it is necessary to discard all predictions that exist on the border of the image, where it is likely that some proportion of these particles exist outside of the image. If these partially visible particles are not discarded, they will bias the aggregate measures or distribution of particle sizes if measured alongside

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

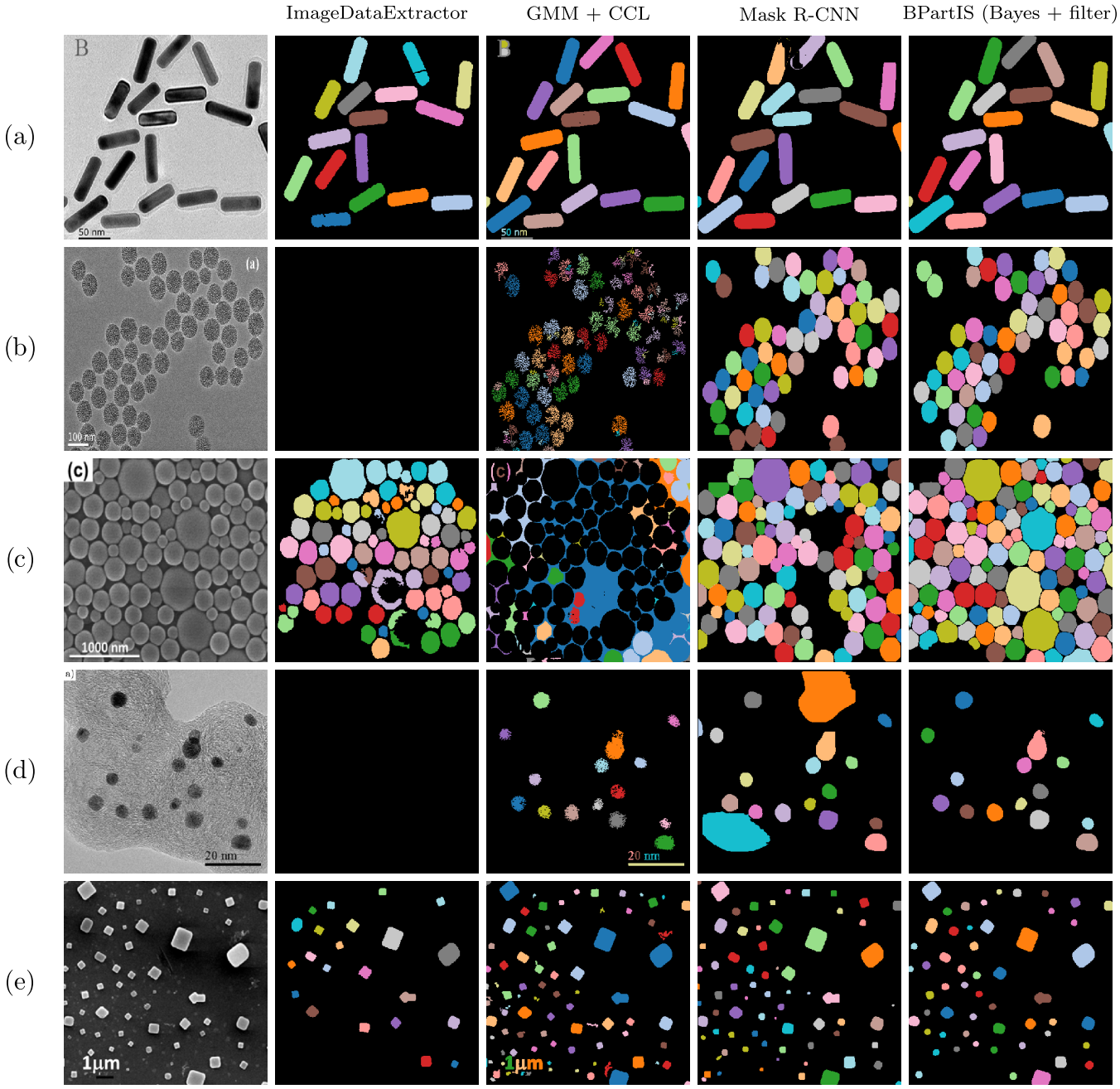


Figure 5. Qualitative comparison of BPartIS (Bayesian with uncertainty filtering) with other methods: ImageDataExtractor,[1] m2py,[8] and Mask R-CNN.[10] All five images are from the EMPS test set. (a) TEM of Au nanorods by He et al.[59] reprinted from ref 59, Copyright (2019); (b) TEM of dendritic-like mesoporous silica by Chen et al.[75] reprinted from ref 75, Copyright (2020); (c) SEM of polydisperse polystyrene spheres by Zheng et al.[76] reprinted from ref 76, Copyright (2020); (d) TEM of Pt_3Co nanoparticles by Rasouli et al.[77] reprinted from ref 77, Copyright (2017); and (e) SEM of Pd nanocrystals by Navas et al.[53] reprinted from ref 53, Copyright (2016). All with permission from Elsevier.

the valid particles. Thus, we find all particles that intersect with the image borders and remove these from the set of particles to measure. Finally, we end up with a set of particles from which we can compute quantitative measures.

To measure each particle, we iterate through all of the predicted instance-segmentation masks and compute the area covered by each mask in pixels-squared. These measurements can subsequently be converted into relevant units by the conversion factor computed from scalebar measurements. From these, aggregate-size measures and distributions can be calculated. We show two examples in Figure 8 where we computed particle-size distributions and radial-distribution functions (using rdfpy[78]) from two images that are not part of the EMPS data set.

Quantitative Evaluation of Particle Analysis. Although the accuracy of the quantitative measures we compute can be extrapolated from the accuracy of the segmentation model, we performed an assessment of their accuracy by comparing particle analyses derived from BPartIS predictions with those derived from ground-truth segmentation maps from the EMPS test set. These include particle-size histograms, aspect-ratio histograms, and radial-distribution functions. For this analysis, we selected all images from the test set which contained greater than or equal to 30 particles, which resulted in 32 images. For

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

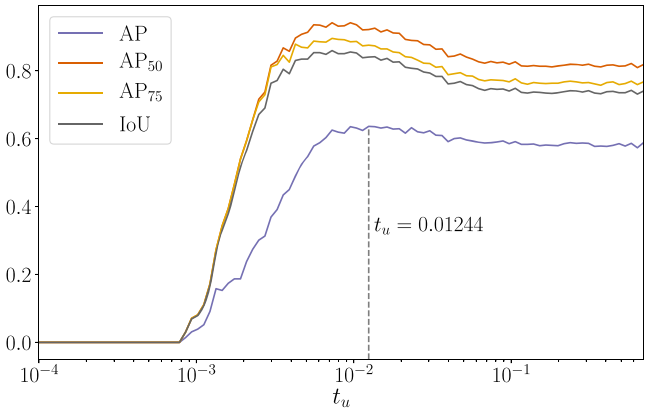


Figure 6. Metrics as a function of uncertainty threshold (t_u).

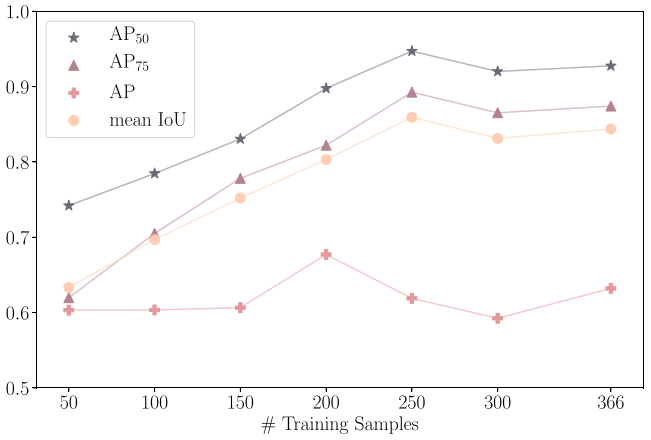


Figure 7. Metrics as a function of the number of training samples.

particle-size and aspect-ratio histograms, we compared predicted to ground-truth histograms using the histogram cosine distance metric: $d_{\cos} = 1 - \sum_{i=1}^{N} \frac{h_1(i)h_2(i)}{\| h_1 \| \| h_2 \|}$, where h_1 and h_2 are the histograms being compared and $h_1(i)$ is the count of ith bin in h_1. We compared predicted and ground-truth radial-distribution functions using the root-mean-square error (RMSE) metric: $\|g_{\text{pred}}(r) - g_{\text{gt}}(r)\|_2$, where $g_{\text{pred}}(r)$ and $g_{\text{gt}}(r)$ are the predicted and ground-truth radial-distribution functions, respectively. We computed these metrics for each of the 32 (prediction, ground-truth) pairs and averaged across all pairs. Average cosine distances for particle-size and aspect-ratio histograms were 0.02111 and 0.03261, respectively. The average RMSE for radial-distribution functions was found to be 0.63646. These values suggest that quantitative measures computed from BPartIS predictions can provide a faithful representation of the true underlying particle statistics present in an EM image.

CONCLUSIONS

We have presented a Bayesian deep-learning methodology to segment particle instances in EM images for quantitative analysis. By comparing the performance of BPartIS with other similar methods, we showed that our method performs better than an image-processing method, a machine-learning-based method, and a proposal-based deep-learning method for particle instance segmentation. Our model was trained on a human-labeled particle instance segmentation data set consisting of 465 images and ground-truth segmentation maps, which we make publicly available at https://imagedataextractor.org/evaluation. We have demonstrated the ability of BPartIS to be used in quantitative particle analysis, by computing particle-size distributions and radial-

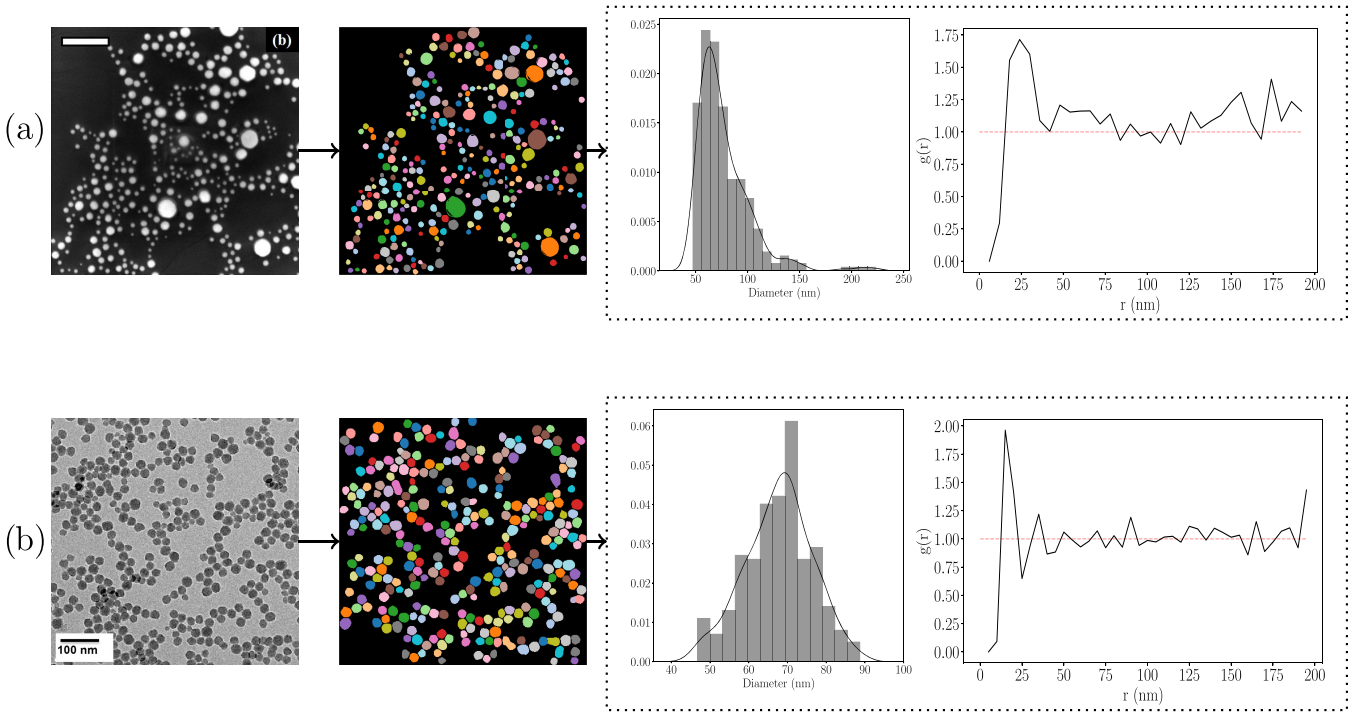


Figure 8. Example particle-size distributions and radial-distribution functions computed from BPartIS predictions of images not present in the EMPS data set. (a) SEM of Au@SiO_2 core–shell nanoparticles by Gundanna et al.[79] reprinted from ref 79, Copyright (2020); (b) TEM of ERM FD 304 colloidal SiO_2 nanoparticles by Dazon et al.[80] reprinted from ref 80, Copyright (2019). All with permission from Elsevier.

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

distribution functions from EM images found in scientific literature. The code and implementation of BPartIS can be found at https://github.com/by256/bpartis, where we provide scripts to reproduce our results. The BPartIS model replaces the particle-detection and quantitative-analysis steps of ImageDataExtractor[1] to afford ImageDataExtractor v2.0.

AUTHOR INFORMATION

Corresponding Author

Jacqueline M. Cole − *Cavendish Laboratory, Department of Physics, University of Cambridge, Cambridge CB3 0HE, U.K.; ISIS Neutron and Muon Source, STFC Rutherford Appleton Laboratory, Didcot, Oxfordshire OX11 OQX, U.K.; Research Complex at Harwell, Rutherford Appleton Laboratory, Didcot, Oxfordshire OX11 OQX, U.K.; Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, U.K.;* orcid.org/0000-0002-1552-8743; Email: jmc61@cam.ac.uk

Author

Batuhan Yildirim − *Cavendish Laboratory, Department of Physics, University of Cambridge, Cambridge CB3 0HE, U.K.; ISIS Neutron and Muon Source, STFC Rutherford Appleton Laboratory, Didcot, Oxfordshire OX11 OQX, U.K.; Research Complex at Harwell, Rutherford Appleton Laboratory, Didcot, Oxfordshire OX11 OQX, U.K.;* orcid.org/0000-0001-7741-0516

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c01455

Notes

The authors declare no competing financial interest.

ImageDataExtractor 2.0 and BPartIS are released under the MIT license and are available to download from https://imagedataextractor.org, where an interactive demo, user guide, code examples, and full API documentation are available. The Electron Microscopy Particle Segmentation (EMPS) data set is available at https://imagedataextractor.org/evaluation, which includes 465 images and ground-truth segmentation maps.

ACKNOWLEDGMENTS

J.M.C. is grateful for the BASF/Royal Academy of Engineering Research Chair in Data-Driven Molecular Engineering of Functional Materials. J.M.C. is also indebted to the Science and Technology Facilities Council (STFC) via the ISIS Neutron and Muon Source, who partly support this Research Chair and provide PhD studentship support (for B.Y.).

REFERENCES

(1) 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. PMID: 31714792.

(2) Swain, M. C.; Cole, J. M. ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. *J. Chem. Inf. Model.* **2016**, *56*, 1894−1904. PMID: 27669338.

(3) Hiszpanski, A. M.; Gallagher, B.; Chellappan, K.; Li, P.; Liu, S.; Kim, H.; Han, J.; Kailkhura, B.; Buttler, D. J.; Han, T. Y.-J. Nanomaterial Synthesis Insights from Machine Learning of Scientific Articles by Extracting, Structuring, and Visualizing Knowledge. *J. Chem. Inf. Model.* **2020**, *60*, 2876−2887. PMID: 32286818.

(4) Groom, D.; Yu, K.; Rasouli, S.; Polarinakis, J.; Bovik, A.; Ferreira, P. Automatic Segmentation of Inorganic Nanoparticles in BF TEM Micrographs. *Ultramicroscopy* **2018**, *194*, 25−34.

(5) Meng, Y.; Zhang, Z.; Yin, H.; Ma, T. Automatic Detection of Particle Size Distribution by Image Analysis Based on Local Adaptive Canny Edge Detection and Modified Circular Hough Transform. *Micron* **2018**, *106*, 34−41.

(6) Mirzaei, M.; Rafsanjani, H. K. An Automatic Algorithm for Determination of the Nanoparticles from TEM Images using Circular Hough Transform. *Micron* **2017**, *96*, 86−95.

(7) Kim, H.; Han, J.; Han, T. Y.-J. Machine Vision-Driven Automatic Recognition of Particle Size and Morphology in SEM Images. *Nanoscale* **2020**, *12*, 19461−19469.

(8) Tatum, W. K.; Torrejon, D.; O'Neil, P.; Onorato, J. W.; Resing, A. B.; Holliday, S.; Flagg, L. Q.; Ginger, D. S.; Luscombe, C. K. Generalizable Framework for Algorithmic Interpretation of Thin Film Morphologies in Scanning Probe Images. *J. Chem. Inf. Model.* **2020**, *60*, 3387−3397.

(9) Zhang, F.; Zhang, Q.; Xiao, Z.; Wu, J.; Liu, Y. Spherical Nanoparticle Parameter Measurement Method Based on Mask R-CNN Segmentation and Edge Fitting. *Pattern Recognit.* **2019**, 205−212.

(10) He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *International Conference on Computer Vision*, arXiv:1703.06870. arXiv.org e-Print archive. https://arxiv.org/abs/1703.06870 (submitted on Mar 20, 2017).

(11) Frei, M.; Kruis, F. Image-based Size Analysis of Agglomerated and Partially Sintered Particles via Convolutional Neural Networks. *Powder Technol.* **2020**, *360*, 324−336.

(12) Rueden, C. T.; Schindelin, J.; Hiner, M. C.; DeZonia, B. E.; Walter, A. E.; Arena, E. T.; Eliceiri, K. W. ImageJ2: ImageJ for the Next Generation of Scientific Image Data. *BMC Bioinf.* **2017**, *18*, No. 529.

(13) Wu, Y.; Lin, M.; Rohani, S. Particle characterization with on-line imaging and neural network image analysis. *Chem. Eng. Res. Des.* **2020**, *157*, 114−125.

(14) Sarma, B.; Sarma, B. K. Fabrication of Ag/ZnO Heterostructure and the Role of Surface Coverage of ZnO Microrods by Ag n=Nanoparticles on the Photophysical and Photocatalytic Properties of the Metal-Semiconductor System. *Appl. Surf. Sci.* **2017**, *410*, 557−565.

(15) Dai, J.; He, K.; Sun, J. In *Instance-Aware Semantic Segmentation via Multi-task Network Cascades*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2016; pp 3150−3158.

(16) Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. In *Fully Convolutional Instance-Aware Semantic Segmentation*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2017; pp 4438−4446.

(17) Pinheiro, P. O.; Collobert, R.; Dollár, P. In *Learning to Segment Object Candidates*, NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, December 2015; pp 1990−1998.

(18) Pinheiro, P. O.; Lin, T.; Collobert, R.; Dollár, P. In *Learning to Refine Object Segments*, European Conference on Computer Vision; Computer Vision−ECCV 2016, 2016; pp 75−91.

(19) de Brabandere, B.; Neven, D.; van Gool, L. In *Semantic Instance Segmentation with a Discriminative Loss Function*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2017.

(20) Neven, D.; Brabandere, D. B.; Proesmans, M.; Gool, V. L. In *Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019; pp 8837−8845.

(21) Newell, A.; Huang, Z.; Deng, J. In *Associative Embedding: End-to-End Learning for Joint Detection and Grouping*, Advances in Neural Information Processing Systems 30 (NIPS 2017), 2017; pp 2277−2287.

(22) Kong, S.; Fowlkes, C. In *Recurrent Pixel Embedding for Instance Grouping*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018; pp 9018−9028.
(23) Fathi, A.; Wojna, Z.; Rathod, V.; Wang, P.; Song, H. O.; Guadarrama, S.; Murphy, K. Semantic Instance Segmentation via Deep Metric Learning, arXiv.abs/1703.10277. arXiv.org e-Print archive. https://arxiv.org/abs/1703.10277 (accessed March 30, 2017).
(24) Novotný, D.; Albanie, S.; Larlus, D.; Vedaldi, A. In *Semi-convolutional Operators for Instance Segmentation*, Proceedings of the European Conference on Computer Vision, 2018; pp 89−105.
(25) Yu, J.; Blaschko, M. B. In *Learning Submodular Losses with the Lovász Hinge*, International Conference on Machine Learning, July 2015; pp 1623−1631.
(26) Berman, M.; Blaschko, M. B. In *Optimization of the Jaccard Index for Image Segmentation with the Lovász Hinge*, International Conference on Machine Learning, 2015; pp 1623−1631.
(27) MacKay, D. J. C. A Practical Bayesian Framework for Backpropagation Networks. *Neural Comput.* **1992**, *4*, 448−472.
(28) Neal, R. M. *Bayesian Learning for Neural Networks*; Springer-Verlag, 1996.
(29) Gal, Y. Uncertainty in Deep Learning. Ph.D. Thesis, University of Cambridge: United Kingdom, 2016.
(30) Gal, Y.; Ghahramani, Z. In *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*, International Conference on Machine Learning, 2016; pp 1050−1059.
(31) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929−1958.
(32) Kendall, A.; Gal, Y. In *What Uncertainties do We Need in Bayesian Deep Learning for Computer Vision?*, NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing, 2017; pp 5574−5584.
(33) Daxberger, E. A.; Hernández-Lobato, J. M. Bayesian Variational Autoencoders for Unsupervised Out-of-Distribution Detection. arXiv:1912.05651. arXiv.org e-Print archive. https://arxiv.org/abs/1912.05651 (submitted on Dec 11, 2019).
(34) Shannon, C. E. A Mathematical Theory of Communication. *Bell Syst. Tech. J* **1948**, *27*, 379−423.
(35) Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. *Br. Mach. Vis. Conf.* **2017**, 57.1−57.12.
(36) Houlsby, N.; Huszár, F.; Ghahramani, Z.; Lengyel, M. Bayesian Active Learning for Classification and Preference Learning, arXiv:1112.5745.arXiv. org e-Print archive. https://arxiv.org/abs/1112.5745 (submitted on Dec 24, 2011).
(37) Smith, L.; Gal, Y. Understanding Measures of Uncertainty for Adversarial Example Detection. *Uncertain. Artif. Intell.* **2018**.
(38) Malinin, A.; Mlodozeniec, B.; Gales, M. In *Ensemble Distribution Distillation*, International Conference Learning Representation, 2020; pp 1−22.
(39) Depeweg, S.; Hernandez-Lobato, J.-M.; Doshi-Velez, F.; Udluft, S. In *Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning*, Proceedings of Machine Learning Research, 2018; pp 1184−1193.
(40) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. In *ImageNet Classification with Deep Convolutional Neural Networks*, Advanced Neural Information Processing Systems, 2012; pp 1097−1105.
(41) Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.-A.; Vincent, P.; Bengio, S. Why Does Unsupervised Pre-Training Help Deep Learning. *J. Mach. Learn. Res.* **2010**, *11*, 625−660.
(42) van den Oord, A.; Li, Y.; Vinyals, O. In Representation Learning with Contrastive Predictive Coding, arXiv:1807.03748. arXiv.org e-Print archive. https://arxiv.org/abs/1807.03748 (submitted on Jul 10, 2019).
(43) Hénaff, O. J.; Srinivas, A.; Fauw, J. D.; Razavi, A.; Doersch, C.; Eslami, S. M. A.; van den Oord, A. In *Data-Efficient Image Recognition with Contrastive Predictive Coding*, International Journal on Machine Learning, 2020.
(44) Hjelm, D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. In *Learning Deep Representations by Mutual Information Estimation and Maximization*, International Conference on Learning Representations, 2019.
(45) Aversa, R.; Modarres, M.; Cozzini, S.; Ciancio, R.; Chiusole, A. The First Annotated Set of Scanning Electron Microscopy Images for Nanoscience. *Sci. Data* **2018**, *5*, No. 180172.
(46) Romera, E.; Álvarez, J. M.; Bergasa, L. M.; Arroyo, R. In *ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation*, IEEE Transactions on Intelligent Transportation Systems, 2018; pp 263−272.
(47) Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and Checkerboard Artifacts *Distill* 2016, *1*, DOI: 10.23915/distill.00003.
(48) Nair, V.; Hinton, G. E. In *Rectified Linear Units Improve Restricted Boltzmann Machines*, Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010; pp 807−814.
(49) Clevert, D.; Unterthiner, T.; Hochreiter, S. In *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*, International Conference on Learning Representations, 2016.
(50) Kingma, D. P.; Ba, J. In *Adam: A Method for Stochastic Optimization*, International Conference on Learning Representations, 2015.
(51) Falcaro, P.; Ricco, R.; Yazdi, A.; Imaz, I.; Furukawa, S.; Maspoch, D.; Ameloot, R.; Evans, J. D.; Doonan, C. J. Application of Metal and Metal Oxide Nanoparticles@MOFs. *Coord. Chem. Rev.* **2016**, *307*, 237−254.
(52) Jiang, S.; He, W.; Landfester, K.; Crespy, D.; Mylon, S. E. The Structure of Fibers Produced by Colloid-Electrospinning Depends on the Aggregation State of Particles in the Electrospinning Feed. *Polymer* **2017**, *127*, 101−105.
(53) Navas, M.; Soni, R. Bromide (Br) Ion-Mediated Synthesis of Anisotropic Palladium Nanocrystals by Laser Ablation. *Appl. Surf. Sci.* **2016**, *390*, 718−727.
(54) Meng, X.; Shibayama, T.; Yu, R.; Ishioka, J.; Watanabe, S. Ion Beam Surface Nanostructuring of Noble Metal Films with Localized Surface Plasmon Excitation. *Curr. Opin. Solid State Mater. Sci.* **2017**, *21*, 177−188.
(55) Li, W.; Wu, X.; Li, S.; Tang, W.; Chen, Y. Magnetic Porous Fe_3O_4/Carbon Octahedra Derived from Iron-Based Metal-Organic Framework as Heterogeneous Fenton-like Catalyst. *Appl. Surf. Sci.* **2018**, *436*, 252−262.
(56) Balling, P.; et al. Improving the Efficiency of Solar Cells by Upconverting Sunlight using Field Enhancement from Optimized Nano Structures. *Opt. Mater.* **2018**, *83*, 279−289.
(57) Yang, J.; Kou, Q.; Liu, Y.; Wang, D.; Lu, Z.; Chen, L.; Zhang, Y.; Wang, Y.; Zhang, Y.; Han, D.; Xing, S. G. Effects of amount of benzyl ether and reaction time on the shape and magnetic properties of Fe_3O_4 nanocrystals. *Powder Technol.* **2017**, *319*, 53−59.
(58) Distaso, M.; Apeleo Zubiri, B.; Mohtasebi, A.; Inayat, A.; Dudák, M.; Kočí, P.; Butz, B.; Klupp Taylor, R.; Schwieger, W.; Spiecker, E.; Peukert, W. Three-Dimensional and Quantitative Reconstruction of Non-Accessible Internal Porosity in Hematite Nanoreactors using 360 Electron Tomography. *Microporous Mesoporous Mater.* **2017**, *246*, 207−214.
(59) He, Z.; Cai, Y.; Yang, Z.; Li, P.; Lei, H.; Liu, W.; Liu, Y. A Dual-Signal Readout Enzyme-Free Immunosensor Based on Hybridization Chain Reaction-Assisted Formation of Copper Nanoparticles for the Detection of Microcystin-LR. *Biosens. Bioelectron.* **2019**, *126*, 151−159.
(60) Roy, E.; Patra, S.; Saha, S.; Kumar, D.; Madhuri, R.; Sharma, P. K. Shape Effect on the Fabrication of Imprinted Nanoparticles: Comparison Between Spherical-, Rod-, Hexagonal-, and Flower-Shaped Nanoparticles. *Chem. Eng. J.* **2017**, *321*, 195−206.
(61) Wu, J.; Zhang, L.; Huang, F.; Ji, X.; Dai, H.; Wu, W. Surface Enhanced Raman Scattering Substrate for the Detection of Explosives: Construction Strategy and Dimensional Effect. *J. Hazard. Mater.* **2020**, *387*, No. 121714.

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149

(62) Wu, Y.; Ji, Y.; Xu, J.; Liu, J.; Lin, Z.; Zhao, Y.; Sun, Y.; Xu, L.; Chen, K. Crystalline Phase and Morphology Controlling to Enhance the Up-Conversion Emission from NaYF4:Yb,Er Nanocrystals. *Acta Mater.* **2017**, *131*, 373−379.
(63) Shang, B.; Wang, Y.; Peng, B.; Deng, Z. Bioinspired Polydopamine Coating as a Versatile Platform for Synthesizing Asymmetric Janus Particles at an Air-Water Interface. *Appl. Surf. Sci.* **2020**, *509*, No. 145360.
(64) Liu, P.; Zhang, M.; Xie, S.; Wang, S.; Cheng, W.; Cheng, F. Non-Enzymatic Glucose Biosensor Based on Palladium-Copper Oxide Nanocomposites Synthesized via Galvanic Replacement Reaction. *Sens. Actuators, B* **2017**, *253*, 552−558.
(65) Wang, Y.; Yang, J.; Liu, H.; Wang, X.; Zhou, Z.; Huang, Q.; Song, D.; Cai, X.; Li, L.; Lin, K.; Xiao, J.; Liu, P.; Zhang, Q.; Cheng, Y. Osteotropic Peptide-Mediated Bone Targeting for Photothermal Treatment of Bone Tumors. *Biomaterials* **2017**, *114*, 97−105.
(66) Wang, Y.; Li, Z.; Hu, Y.; Liu, J.; Guo, M.; Wei, H.; Zheng, S.; Jiang, T.; Sun, X.; Ma, Z.; Sun, Y.; Besenbacher, F.; Chen, C.; Yu, M. Photothermal conversion-coordinated Fenton-like and photocatalytic reactions of $Cu_{2-x}Se$-Au Janus nanoparticles for tri-combination antitumor therapy. *Biomaterials* **2020**, *255*, No. 120167.
(67) Dutta, A.; Zisserman, A. In *The VIA Annotation Software for Images, Audio and Video*, Proceedings of the ACM International Conference on Multimedia ACM Multimedia, 2019; pp 2276−2279.
(68) Dutta, A.; Gupta, A.; Zissermann, A. VGG Image Annotator (VIA). Version: 2.0.10, 2016, http://www.robots.ox.ac.uk/~vgg/software/via/ (accessed on April 2020).
(69) He, K.; Zhang, X.; Ren, S.; Sun, J. In *Deep Residual Learning for Image Recognition*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016; pp 770−778.
(70) Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. In *Simultaneous Detection and Segmentation*, European Conference on Computer Vision: Computer Vision−ECCV, 2014; pp 297−312.
(71) Sun, G.; Ge, H.; Luo, J.; Liu, R. Highly Wear-Resistant UV-Curing Antibacterial Coatings via Nanoparticle Self-Migration to the Top Surface. *Prog. Org. Coat.* **2019**, *135*, 19−26.
(72) Tan, C.; Zhou, K.; Ma, W.; Zhang, P.; Liu, M.; Kuang, T. Microstructural Evolution, Nanoprecipitation Behavior and Mechanical Properties of Selective Laser Melted High-Performance Grade 300 Maraging Steel. *Mater. Des.* **2017**, *134*, 23−34.
(73) Faria, A. F.; Liu, C.; Xie, M.; Perreault, F.; Nghiem, L. D.; Ma, J.; Elimelech, M. Thin-film Composite Forward Osmosis Membranes Functionalized with Graphene Oxide-Silver Nanocomposites for Biofouling Control. *J. Membr. Sci.* **2017**, *525*, 146−156.
(74) Shah, M.; Zhang, F.; Ahmad, A. Catalytic Conversion of Substituted and Un-Substituted Cyclohexanone into Corresponding Enones and Phenols by Nanocatalysts Under Acid or Base-Free Reaction Conditions. *Appl. Catal., A* **2017**, *531*, 161−168.
(75) Chen, Y.; Mu, Z.; Wang, W.; Chen, A. Development of mesoporous SiO_2/CeO_2 core/shell nanoparticles with tunable structures for non-damage and efficient polishing. *Ceram. Int.* **2020**, *46*, 4670−4678.
(76) Zheng, X.; Zhang, Z.; Meng, S.; Wang, Y.; Li, D. Regulating Charge Transfer Over 3D Au/ZnO Hybrid Inverse Opal Toward Efficiently Photocatalytic Degradation of Bisphenol A and Photoelectrochemical Water Splitting. *Chem. Eng. J.* **2020**, *393*, No. 124676.
(77) Rasouli, S.; Ortiz Godoy, R.; Yang, Z.; Gummalla, M.; Ball, S.; Myers, D.; Ferreira, P. Surface area loss mechanisms of Pt_3Co nanocatalysts in proton exchange membrane fuel cells. *J. Power Sources* **2017**, *343*, 571−579.
(78) Yildirim, B. rdfpy: a Python Library for Fast Computation of 2D and 3D radial-distribution functions, 2020https://doi.org/10.5281/zenodo.4298486.
(79) Gundanna, S. K.; Mitra, A.; Bhatta, L. K.; Bhatta, U. M. SEM study of site-specific thermal behavior of $Au@SiO_2$ core−shell nanostructures under inert and air atmospheres. *Nano-Struct. Nano-Objects* **2020**, *23*, No. 100521.
(80) Dazon, C.; Maxit, B.; Witschger, O. Comparison Between a Low-Voltage Benchtop Electron Microscope and Conventional TEM for Number Size Distribution of Nearly Spherical Shape Constituent Particles of Nanomaterial Powders and Colloids. *Micron* **2019**, *116*, 124−129.

https://dx.doi.org/10.1021/acs.jcim.0c01455
J. Chem. Inf. Model. 2021, 61, 1136−1149