NP-SAM: Implementing the Segment Anything Model for Easy Nanoparticle Segmentation in Electron Microscopy Images

Rasmus Larsen,1 Torben L. Villadsen,1 Jette K. Mathiesen,2 Kirsten M. Ø. Jensen,3 Espen D. Bøjesen1*

1: Interdisciplinary Nanoscience Center & Aarhus University Centre for Integrated Materials Research, Aarhus University, Gustav Wieds Vej 14, Aarhus C, 8000 Denmark
2: Surface Physics & Catalysis (SURFCAT), Department of Physics, Technical University of Denmark, Fysikvej 312, 2800 Kgs Lyngby, Denmark
3: Department of Chemistry, University of Copenhagen, 2100 Copenhagen Ø, Denmark
* Corresponding author: espen.bojesen@inano.au.dk - NP-SAM GitLab: https://gitlab.au.dk/disorder/np-sam.git

Abstract

Despite the numerous existing (semi)automated workflows for image segmentation of electron microscopy pictures of nanoparticles for statistical size and shape determination the prevalent approach to particle counting still is doing so in cumbersome manual fashion. Here, we present an easily implementable, low entry barrier workflow for nanoparticle segmentation, which eliminates the need for manual particle counting. It is based on the recently released segment anything model and widely distributed, well maintained, python libraries. We explore the impressive zero shot performance of the segment anything model and present approaches for subsequent filtering of outputs to minimize over and under segmentation on a range of different electron microscopy images of nanoparticles. Furthermore, we introduce a novel methodology for handling partial overlap between nanoparticles, which comprise one of the biggest obstacles for many automated segmentation algorithms. Our presented workflow is easily adaptable, and we encourage the community to further build on the work we present here.

Introduction

Nanoparticle (NP) properties are inherently linked to their shape, size, and defects. Clever synthesis approaches to control these parameters are being designed, which in turn requires reliable tools for characterizing NPs to guide the synthesis efforts.1 Methods to extract statistically meaningful ensemble average values of these NP characteristics from powder X-ray diffraction, X-ray total scattering, small angle X-ray scattering, dynamic light scattering and other “broad beam” approaches have been optimized over the course of many years.2-6 However, the statistical potency of these methodologies comes at a price – the attained information is limited to ensemble average values due to the large volumes (on the mm³ scale) sampled. Consequently, subtle details about multimodal size and shape distributions as well as correlation of such parameters with elemental variations or structural defects are lost. Very meticulous studies of individual NPs by aberration corrected (scanning) transmission electron microscopy (S)TEM conversely provide precise atomic level structural information on the single NP level.7, 8 Juxtaposing the statistical significance of the broad beam based methods, current (S)TEM studies are limited in their ability to provide the sought after insight for any statistically meaningful number of NPs in a sample. For instance, determining the yield of a NP synthesis with only two products to a precision of 1% requires description of more than 2000 NPs.9 Yet, often only 100-300 NP diameters are counted manually. This approach is slow and brings along potential pitfalls such as an unconscious bias towards counting “particles that fit the
story”. Recent efforts into automated NP classification constitute important steps towards full NP sample categorization for NP size and shape determination. Irrespective of the chosen method being TEM or STEM imaging, a major roadblock for expanding the automated characterization of NPs is the need for an initial image segmentation.

Automatic NP segmentation is relatively straightforward on (S)TEM images of well-separated NPs on a background producing distinctively different contrast the NPs. In such cases, widely implemented automated thresholding algorithms can be used to create a binary mask based on variations of image contrast. However, for samples with significantly varying contrast within one single image, due to, e.g., broad particle size distributions or particles on various supports, simple thresholding-based segmentation fails. In such cases, improved segmentation is achieved by using locally varying thresholding and watershedding. These methods are easy to use and are implemented in such tools as FIJI and scikit-image, which makes them the go-to approach for NP size and shape characterization for many materials scientists. However, both approaches are challenged in cases where NPs are physically touching or, even worse, overlapping. More advanced methods of segmentation exist, including template matching, edge detection, shape classification, image preprocessing, and unsupervised machine learning. Imressive results have been presented using these approaches, yet most of them still struggle when applied to “real samples” with broad size and shape distributions as well as NP-NP overlap. Several promising approaches make use of different flavors of supervised machine learning, including U-Net based convolutional networks and trainable Weka based segmentation. Likely, the lacking realization of the untapped potential of automated classification is due to the fact that the simplest approaches (e.g., local thresholding) fail for “real samples.” In addition, the more advanced approaches may present a steep learning curve and can be difficult to adapt and implement into workflows, while others require large, labelled datasets (currently not existing) to perform well. Therefore, lowering the entry bar for high quality ad-hoc automated segmentation of a broad range of (S)TEM images bears great potential facilitating automated statistical correlative analysis of relevant NP characteristics for applications, i.e., shape, size, composition, and polymorphism.

Recently, researchers from Meta AI Research introduced the Segment Anything project (SA) for image segmentation purposes. The resulting Segment Anything Model (SAM) and the corresponding dataset it was built upon (more than 1 billion masks resulting from 11 million images) have been made freely available for use under an Apache 2.0 license. SAM has already shown promising results in segmenting medical images and geospatial data, while it apparently is challenged by segmentation of camouflaged object detection. Here, we present our python-centric methodology, aptly named NP-SAM, that optimizes SAM based segmentation for the case of segmentation of NPs in (S)TEM images to obtain statistically sound values for NP characteristics from (S)TEM images.
Methodology – The NP-SAM Workflow

SAM reads an image (Figure 1A) as input and returns a list of $n$ python dictionaries that contain the corresponding binary masks, area, bounding boxes, and additional metadata. In NP-SAM, the binary masks are extracted and stacked to produce a 3D NumPy array with the shape $(x, y, n)$ (Figure 1B), where $x$ and $y$ correspond to the image pixels and $n$ to the number of masks output by SAM. SAM tends to segment adjacent objects in multiple ways, resulting in numerous output masks covering the same regions, either fully or partially. Analyzing the raw SAM based masks under the assumption that each mask represents a unique NP would thus result in redundant NP counts and inclusion of unphysically large areas due to NP overlap. NP-SAM corrects this issue by assigning a unique label to all pixels sharing the same combination of one or more masks. This step effectively results in a dimensional reduction of the 3D NumPy array to a 2D array of labels without any loss of information (Figure 1C). Furthermore, this procedure is the crucial step which facilitates postprocessing of the raw-SAM output and, thereby, the removal of questionable masks.

![Figure 1](image)

Figure 1  A) An EM image of NPs is provided as input for NP-SAM. B) SAM outputs a set of binary masks stored in a 3D NumPy array. Some masks overlap spatially. C) Every pixel that shares the same unique combination of mask is given the same label, resulting in a 2D image of labels. D) Some labels can be filtered, e.g. the yellow label in C) has a low solidity and is removed. E) Questionable labels that come from the overlap of many masks can be selected and handled appropriately. F) Finally, visualizations and statistics of the measured characteristics can be produced. More details on the procedure are provided in the SI.

The first step in NP-SAMS postprocessing is to use the scikit-image function regionprops to extract characteristics of the individual areas marked by the labels and store these values in a pandas DataFrame. Having unique characteristics associated with each label enables our implementation of...
a sequential set of conditional filters based on such parameters as NP area, NP solidity, or average HAADF intensity to improve the outcome of the “blind” segmentation. For example, NP-SAM may segment two adjacent particles into three different labels shown in Fig 1 C as mask 5 (grey), 6 (rosa), and 7 (yellow). Without any filtering, our approach would produce a “yellow edge” labelled area which mistakenly would be counted as a large peanut shaped hollow particle around the two particles. By using suitable conditional filters such areas can be removed. Furthermore, NP-SAM shows promising capabilities in identifying areas with particle overlap and thus enables the user to deal with such cases based on user defined criteria (Figure 1D & E). Additional details of the different types of filtering and a brief introduction of the procedure for implementing additional filters is provided in the SI. The final output of a NP-SAM based analysis of one image is a Pandas DataFrame containing all the particle characteristics the regionprops function provides. Multiple DataFrames can easily be combined to increase statistical sampling and produce histograms or similar visualizations based on such NP characteristics as shape and solidity.
Results

In the following, we will demonstrate the strengths and limitations of NP-SAM by showcasing a range of different cases often encountered in (S)TEM images of NPs. We will also discuss our efforts in speeding up NP-SAM to become a feasible tool for analysis of many images.

Case 1

Figure 2 A) HAADF image of PdCu NPs, B) labelled image of segmentation result from FIJI's default global thresholding, C) labelled image of segmentation result from FIJI's default LTWS, D) labelled image of segmentation result from our NP-SAM based workflow after applying appropriate filters (SI for details).

As a first test we explore NP-SAMs ability to automatically segment and provide analysis results for an “easy” case, i.e., a HAADF image of well separated PdCu NPs (Figure 2A). We compare NP-SAMs performance to two of the most used segmentation approaches for NP counting, i.e., global thresholding and local Phansalkar thresholding combined with watersheding (LTWS). To ease the comparison of the obtained segmented images throughout the entire manuscript, we label each segment with a unique label and color it accordingly. As seen in Figure 2B, the global thresholding fails the task of segmenting this image of relatively monodisperse, large (compared to the field of view), and well separated NPs. Especially, overlapping and touching particles cause under-segmentation. The approach of LTWS, on the other hand, can segment particles that are very close to each other as seen in Figure 2C. Finally, in Figure 2D, we show the outcome of using NP-SAM with applied filters based on HAADF image intensity and Euler characteristic number (SI for details). Based on a qualitative comparison, NP-SAM and LTWS perform equally well, and both outperform the global thresholding. In SI Figure 1A, we compare the obtained histograms from a manual segmentation and NP size measurement with that of SAM and get very good agreement between the two (especially considering the low number of particles).
Next, we test NP-SAMs ability to handle more complex contrast variations, NP proximity, and low degrees of overlap. Figure 3A shows a HAADF image of non-spherical AgCuIrPdPt NPs with varying intraparticle contrast due to core-shell structures. The NPs are lying close to each other with many of them touching. The image contains areas where two layers of NPs lie on top of each other. The latter will in many automated segmentation approaches either cause over- or under-segmentation. LTWS segments some of the particles correctly (Figure 3B). Yet, LTWS struggles when faced with NPs displaying large intraparticle intensity variation, e.g., the area marked with a red circle in Figure 3B. The HAADF intensity in the center of the particle causes the local thresholding to falsely mark the center as background and the following watershedding wrongly splits the particle into three distinct areas. Figure 3C shows the NP-SAM output after we have applied filters based on average NP HAADF intensity, average NP area, and average NP solidity (SI for details). Clearly, NP-SAM performs better than LTWS, yet some over-segmentation occurs due to particle overlap (see pink ring on Figure 3D). In Figure 3D, we present the labelled image after we have applied a filter that removes all potentially overlapping particles from the SAM segmentation result (SI for details). The employed approach will remove most of the potentially over-segmented areas at the cost of poorer counting statistics, a drawback which can be remedied simply by collecting more images from additional areas.

Intriguingly, it appears that our NP-SAM workflow enables identification of potentially problematic areas in an image, e.g., areas with particle overlap, at least in cases with moderate (single layer) overlap. This is a feature most automatic segmentation approaches lack, and we consider it to be a huge step forward in avoiding false outputs. We encourage the community to establish more elaborate approaches of dealing with this overlap than our very conservative way, i.e., deleting them. More intricate methodologies may allow for correct counting of slightly overlapping particles as well.
To explore the effect of sampling on the performance of NP-SAM, we test it on a HAADF image of silver NPs with a very broad size distribution (Figure 4A). LTWS (Figure 4B) performs reasonably well, especially when it comes to segmenting the smallest NPs, while still struggling with overlap. In Figure 4C, we show the output from NP-SAM using SAMs default sampling of 32x32 (SI for details). NP-SAM in this case ignores all the smallest NPs resulting in under segmentation. Increasing SAMs sampling from 32x32 to 128x128 significantly improves the number of small NPs detected by NP-SAM (Figure 4D). Nevertheless, a considerable fraction of the smallest particles is still not detected and increasing the sampling comes at the cost of increased calculation time. On our GPU setup, we experienced an approximate threefold increase in calculation time when changing from 32x32 to 128x128 sampling. Figure SI3 shows an image demonstrating the difference between the outcome of the two sampling grids and the remaining undetected NPs. The histograms shown in SI Figure 1C supports the notion of NP-SAM’s issues with detecting very small particles, i.e., particles corresponding to a diameter of ~10 pixels or below. A possible solution to this limitation would be to collect images at larger magnification, albeit at the cost of poorer NP counting statistics, which in turn could be remedied by collecting more images. In the SI we elaborate upon finding the optimal tradeoff between segmentation quality and computational time.
In this final example, we test NP-Sams performance when it comes to analyzing bright field (BF) TEM images (Figure 5A) and investigate different ways to speed up NP-SAM. As shown in Figure 5B, the NP-SAM workflow performs well on BF TEM images, albeit that areas with large amount of overlap cause incorrect segmentation (which can be dealt with, as explained earlier). Figure 5A shows a 2048x2048 pixel BF TEM image of PdCu particles. This image has four times the number of pixels compared to the 1024x1024 images analyzed in cases 1-3. Consequently, our computing setup had insufficient VRAM (8GB) to perform GPU-based segmentation. Nevertheless, CPU based segmentation was still possible, albeit at the cost of longer computation time (5-10 times slower than GPU based segmentation). The resulting labelled and filtered output is shown in Figure 5B (details in SI). We explore two easily implementable ways to enable GPU based segmentation. In Figure 5C, we show the outcome of the first approach, cropping the image into four images of 1024x1024 pixel dimensions. Each quadrant was individually processed and subsequently stitched back together. Despite speed improvements, the approach introduces unwanted boundaries in the image and complicated the process unnecessarily. We next tested the effect of binning the image by a factor of 2 to obtain a 1024x1024 image with poorer resolution. This enabled fast GPU processing resulting in a 15 times faster segmentation compared to using the CPU on the original 2048x2048 image. Comparing Figure 5D to the non-binned version in Figure 5B shows no discernable differences. Thus, in a case like this, i.e., image collected at sufficient magnification, binning is a viable option for speeding up calculations.

Our initial non-optimized version of NP-SAM used approximately 1.5 s per label and computational time scaled linearly with number of pixels in each image, rendering it unfeasible for high throughput analysis. However, by implementing some degree of parallelization and numba\textsuperscript{35} based optimization, a fourfold increase in speed was realized, i.e., \textasciitilde6 labels/s without any loss of quality of the output. Furthermore, we have implemented a variable “slice step size” parameter to increase performance further. This parameter allows NP-SAM to traverse the NumPy arrays in larger increments than 1. This means fewer elements must be compared and sorted. Impressive increases in computational speed are achieved, e.g., for a step size of 4, we experienced an additional 2.5-fold increase in speed. A beneficial side effect from increasing the “slice step size” is that fewer redundant masks are
produced and therefore fewer labels must be processed. Specifically, any labels smaller than the “slice step size” will be lost during the dimensionality reduction step in going from the 3D array to the 2D array. Consequently, if this “speed up process” is to be used, the smallest particles in an image must occupy an area larger than the selected step size to avoid under segmentation. Certainly, there are further ways to optimize our NP-SAM workflow. However, exploring these goes beyond the scope of this communication, and we strongly encourage the community to collaborate on doing so. More details on the specifics of the NP-SAM code are provided in the SI.

**Conclusion and Final Remarks**

Here, we presented NP-SAM, an easily implementable workflow for NP segmentation from (S)TEM images using SAM and several well-maintained python libraries. NP-SAM performs well when it comes to a wide variety of NP (S)TEM image segmentation tasks. NP-SAMs enables easy filtering of the raw SAM segmentation results on a case-by-case basis. Importantly, we present a straightforward way to remove potentially overlapping areas and hereby reduce over- and under-segmentation. We have taken initial steps to ensure that the computational demands of NP-SAM are kept at an level, which does not require excessive computational power. Moreover, we have sped up the workflow enabling going from an image to an output of NP characteristics in an easily indexable Pandas DataFrame format within minutes. Current limitations are especially encountered when segmenting images with a broad size distribution and NPs that only produce signals occupying very few pixels. In such cases, we suggest collecting additional images at higher magnifications. Automated NP characterization with (S)TEM clearly still requires careful sample preparation and a trained researcher to guide the analysis. Nevertheless, it is intriguing how well NP-SAM performs in many cases and we encourage the community to explore its use further. Specifically, combining various filtering approaches in different order may hold great potential. Moreover, by taking advantage of the plethora of signals produced during a STEM raster scan of a NP sample, we suggest that it may be possible to improve upon segmentation simply by feeding the signal with the greatest contrast variation between background and particle to SAM, e.g., STEM-EDX, STEM-EELS or STEM-BF signals. To expand the progress in the field, we happily share both the images used as well as the implemented workflows with the community.


**Acknowledgements**

The authors acknowledge the help from everyone in the DISORDER group for providing valuable feedback and testing NP-SAM. Krestine Hofstedt Rasmussen is thanked for valuable feedback on the manuscript. J.K.M. is grateful for funding from the VILLUM FONDEN through a Villum International Postdoc Grant (VIL41388). K.M.Ø.J. and J.M.K are grateful for funding from the Villum Foundation through a Villum Young Investigator Grant (VKR00015416). K.M.Ø.J. and J.M.K acknowledge support from the Danish National Research Foundation Center for High Entropy Alloy Catalysis (DNRF 149)

**Author Contributions**

E.D.B got the initial idea for the project. E.D.B. and R.L. performed the (S)TEM imaging. R.L. and T.V.L. developed the NP-SAM implementation. Choice of materials and design of the synthesis of the materials was performed by J.K.M. and K. M. Ø. J. The manuscript was written by E.D.B and R.L. and revised by E.D.B. with help from all other authors.

**Competing Interests**

The authors declare no competing interests.