Supplementary Information

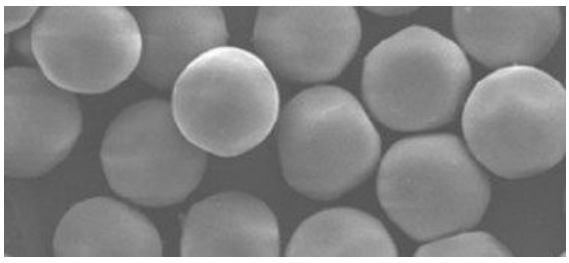
A deep learned nanowire segmentation model using synthetic data augmentation

Binbin Lin[1,*], Nima Emami[1], David A. Santos[2], Yuting Luo[2], Sarbajit Banerjee[2,*], Bai-Xiang Xu[1,*]

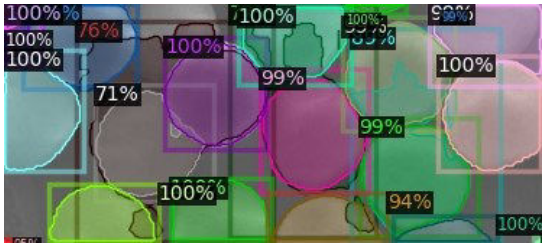
[1]Institute of Materials Science, Technische Universität Darmstadt, 64287 Darmstadt, Germany
[2]Department of Chemistry, Texas A&M University, TX 77843-3255, USA
*Corresponding author: b.lin@mfm.tu-darmstadt.de,
banerjee@chem.tamu.edu, xu@mfm.tu-darmstadt.de

Supplementary Note 1: Additional information on the extension of the deep learning model to account further instance classes

In the present work, the model presented leverages a pre-trained machine learning model that is trained on the large-scale COCO-Dataset with classes and labels of common daily objects. Based on the pretrained weights (for which the algorithm already has learned some general sense of features of objects with various shapes), we fine-tune and optimize the network by introducing the information of optical density-based synthetic nanowire data. This way effectively adds additional constraints in the network training and reinforces the model to learn the newly introduced training data of nanowires. Adding a new class of objects is relatively cheap in terms of computing time, especially with our given synthetic data generation workflow. The nanowires in real images, as in STXM and SEM images, are not exactly regular and straight nanowires, compared to the synthetic dataset. However, the trained algorithm is capable of segmenting non-regular instances, demonstrating the functionality of the present approach. Below we further show a test case for non-nanowire objects with our interactive web application: We see that the trained algorithm can detect spherical objects, by segmentation foreground objects and differentiating the particle boundaries. The results are, of course, not sufficiently good in the overlapped regions, since the model was not trained on spherical particles and their overlaps. One could, therefore, include a training dataset of spherical particles for the model to learn a second class, namely, e.g., nanospheres to enhance the prediction of spherical objects.



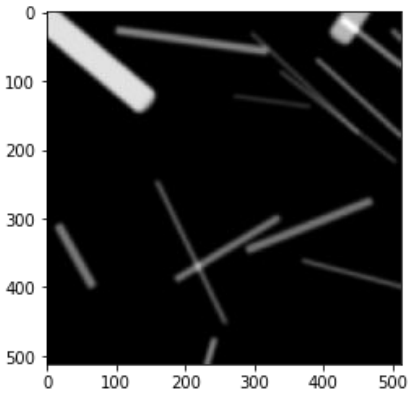
Supplementary Figure 1. Example SEM image of spherical particles with touching boundaries. We note here that the overlapped areas are not visible as in our optical density-based image dataset.



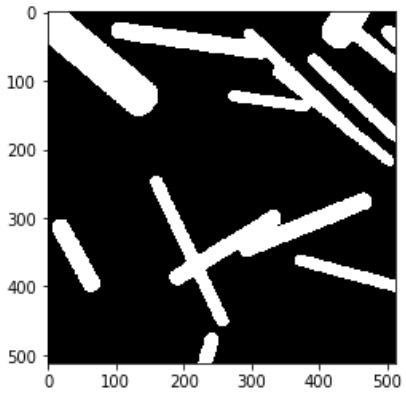
Supplementary Figure 2. Instance segmented image using the current model based on the synthetic dataset of nanowires. The results are not sufficiently good in the overlapped regions; However, the shapes of spheres are well captured. Including further training dataset of spherical objects in the synthetic training dataset can be instrumental to achieve a better performance, especially at the touching boundaries.

Supplementary Note 2: Exemplary segmentation results using Watershed algorithm

The watershed algorithm usually performs well at instance segmentation of similar **touching** objects, where the overlaps are not strictly visible unlike in our optical density-based images. Furthermore, one must firstly segment the foreground information, accompanying several steps of noise removals. Connected component labeling and distance transformation mapping have to be applied to provide initial marker areas (basins) for the watershed algorithm. Such steps can heavily involve manual tuning for complex microscopy images, counteracting efficient image analysis steps. Heavy manual preprocessing steps are spared in our approach by providing images containing different nanowires, that are labeled within noisy images, which can contain different sets of noise sources that are introduced in a synthetic manner. To provide some demonstrative cases using the watershed algorithm, we show below how a marker (connected-components-labeling) based watershed algorithm works on one of our relatively simple synthetic images following the OpenCV implementation in supplementary reference 1. The first step demonstrates the thresholding and segmentation of the foreground object.

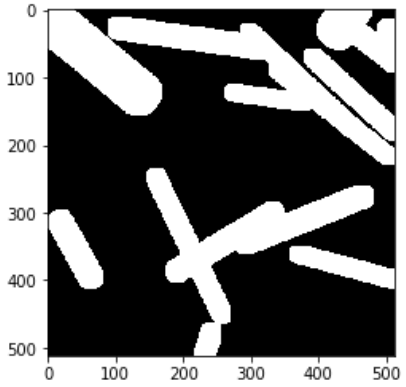


Supplementary Figure 3. Example of a simple synthetic image from our synthetic dataset with only few overlaps without any noises for testing purpose of the Watershed algorithm.

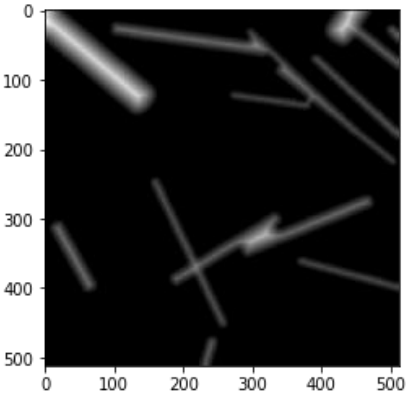


Supplementary Figure 4. Binary thresholding of the foreground nanowires from Supplementary Figure 3.

Further steps include dilate operation of thresholded image for further boundary marking, and distance transformation of foreground image.

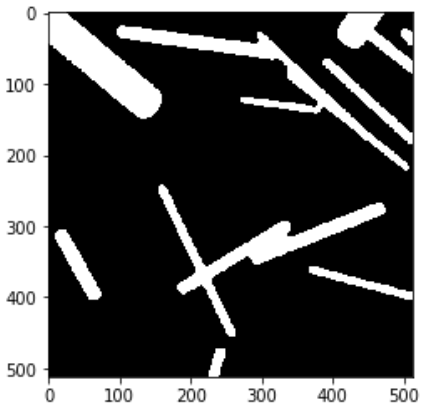


Supplementary Figure 5. Dilation of the thresholded image from Supplementary Figure 4. Dilation operation adds further pixels on the object boundary, which enlarges the object.

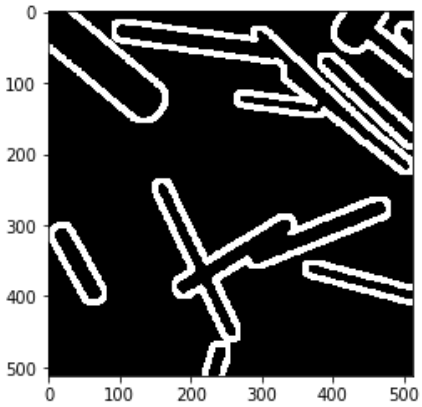


Supplementary Figure 6. Distance transformation of the previous image. Distance transformation generally maps the distance of the foreground object to its surrounding background.

Then we threshold the distance transformed image to extract the sure foreground image and subtract it with the dilated image to get the particle boundaries.

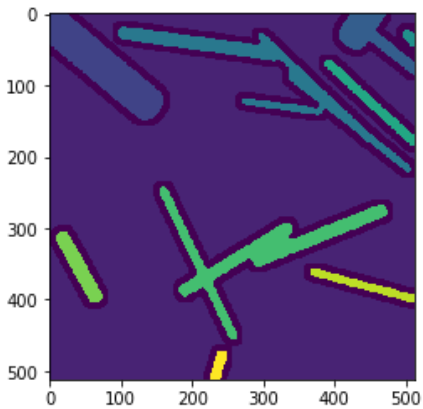


Supplementary Figure 7. Threshold of the distance transformed image to extract the sure foreground image as basis.

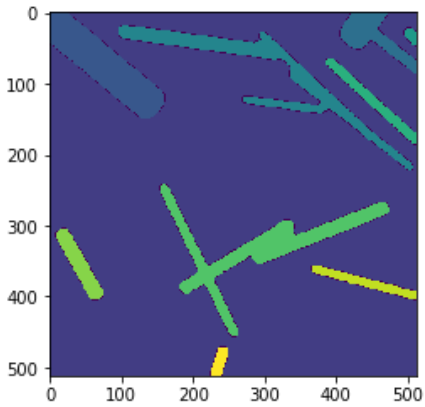


Supplementary Figure 8. Particle boundary extraction by subtract the sure foreground image in Supplementary Figure 7 from dilated image in Supplementary Figure 5.

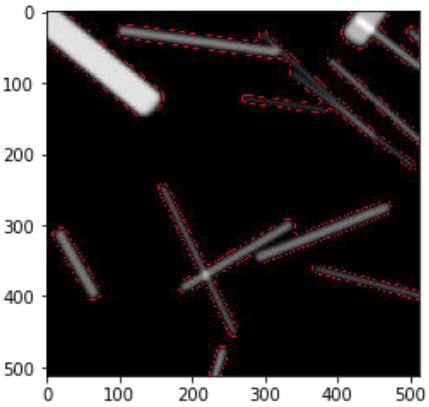
We further performed the connected component labeling algorithm to the sure foreground image and the obtained information served as a marker for the watershed algorithm. The obtained results are shown below:



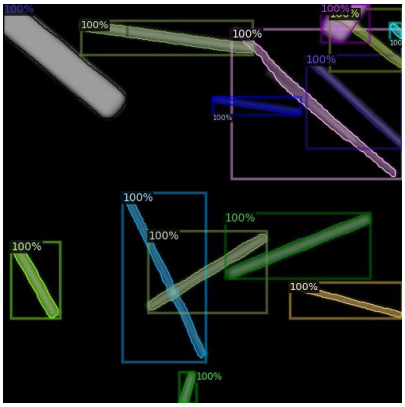
Supplementary Figure 9. Connected-Component-Labeling is performed for Supplementary Figure 8. The algorithm basically finds and marks the individually closed particle boundaries and assigns them consecutive numbers to determine different instances.



Supplementary Figure 10. Final mask image after application of Watershed algorithm with individual instances masked with different color scheme.



Supplementary Figure 11. Final boundaries of individually marked instances.



Supplementary Figure 12. Our model prediction with bounding boxes and masks in color. All overlapped nanowires are successfully detected and segmented as individual instances.

Looking at the results using the aforementioned procedure with a Watershed algorithm, it is clear that using a watershed algorithm -that relies on distance map as markers- is not sufficient at detecting overlapping nanowires, while it performs well for isolated ones. The issue lies in the pixels in the boundary regions, which are not clearly defined for the Watershed through the distance map. This could lead to false segmentation in complex microscopy images, where the boundaries can be blurred by limited spatial resolution, particle defects, and overlapping regions. Our synthetic data model can detect overlapped objects (for which it is trained to do so), whereas traditional algorithms

such as thresholding + connected component analysis are unable to accomplish this objective. The Watershed algorithm can indeed separate touching structures; however, it tends to overestimate/underestimate the boundary in real microscopy images and is prone to noise contamination, with additional preprocessing steps that need to be carried out in advance. However, it must be mentioned that further markers can be used for the watershed algorithm, including manual annotation, which might improve the segmentation results presented above.

Supplementary Note 3: Training results with a small set of real images included

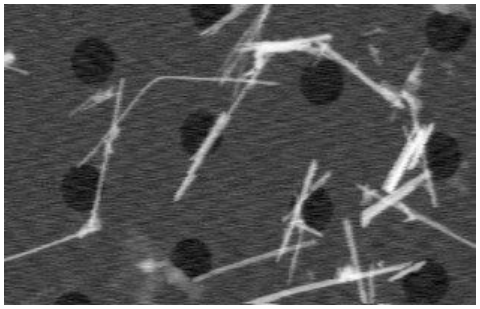
A model has been trained utilizing experimentally generated images. One of the main limitations arises from the limited number of images that can be generated with sufficient variability to train a model. Since the experimental data are limited and hard to label, we included 36 images derived from one real ptychography image, which was cropped in 9 smaller images and augmented with 3 times of 90% rotation each step. This dataset was included additionally into the training dataset size of 500 images for our model training. Adding a small amount of experimental data to the synthetic data in the training stage resulted in a negligible change in terms of AP performance within the large number of training dataset sizes. The difference in AP was within the margin of error (The AP dropped around 0.25), therefore we did not include any of the real images into the training dataset, instead, we used a pure synthetic data set to train our model.

Supplementary Note 4: Training results with smaller learning rates

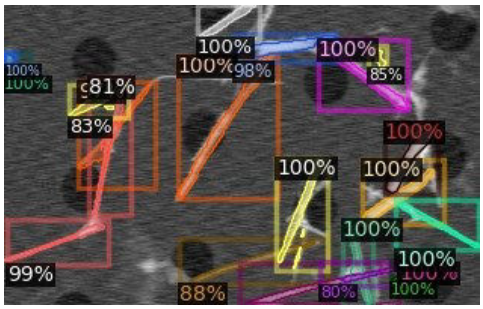
In the present work, the model presented leverages a pre-trained machine learning model that is trained on the large-scale COCO-Dataset with classes and labels of common daily objects. Based on the pre-trained weights, we optimize the network by introducing the information of optical density-based synthetic nanowire data. Such a way effectively adds additional constraints in the network training and reinforces the model to learn the newly introduced training data of nanowires. Therefore, in contrast to training a new network from scratch, where an extremely large dataset would be needed, we chose a comparably larger learning rate of 0.01, 0.02, 0.03 for a certain number of epochs to see how the model reacts to our dataset. We found that with a learning rate of 0.02 and the corresponding epoch numbers and training dataset size, the model performs optimally in our case. To test the model how it reacts to smaller learning rates, we further additionally trained two models with learning rates of 0.005 and 0.001 while keeping other hyperparameters fixed. The resulting AP BOX/SEGM decreased roughly 3 for each synthetic, X-ray ptychography and STXM image and can be found in the supplementary table 1 of hyperparameter study results. The default hyperparameter setting of model #7, has been used to vary the learning rate of 0.005 and 0.001.

Supplementary Note 5: Further statistical results for exemplary image samples

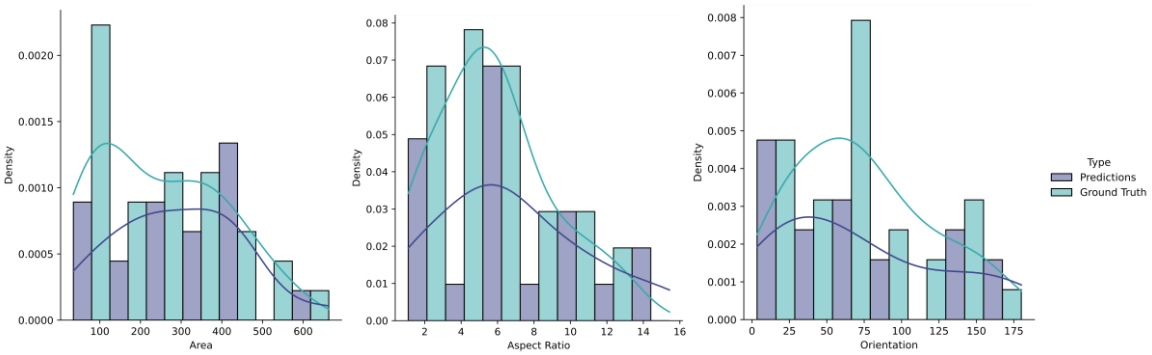
In the following, more STXM and SEM segmentation samples are presented along with the corresponding statistics. This is facilitated using our developed web-based interactive segmentation tool.



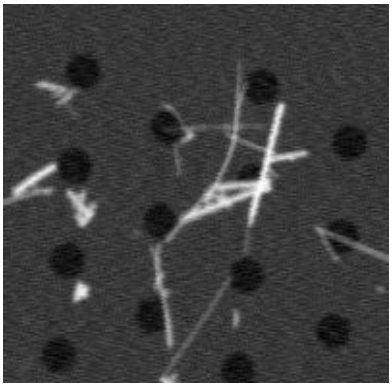
Supplementary Figure 13. A sample STXM image on substrate.



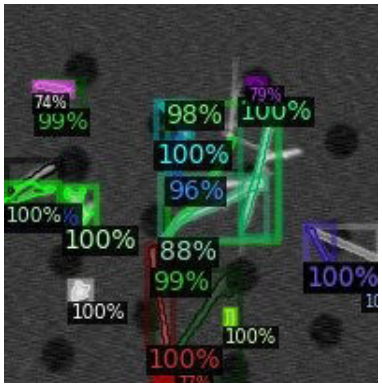
Supplementary Figure 14. Model prediction results on the sample image; The model results here are even insensitive to the substrate with black holes in the background and can predict the nanowires, including overlaps across the substrate holes. The long, curved nanowire, as well as some densely connected nanowires have not been well segmented, leading therefore to statistical deviations as shown in the following statistics.



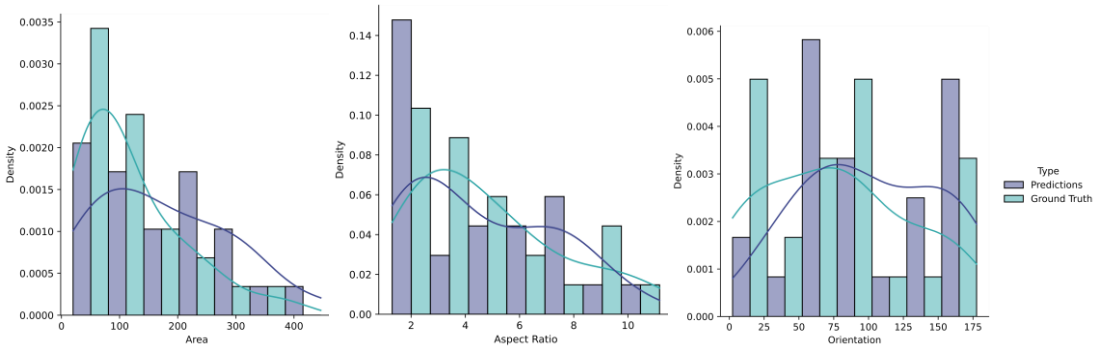
Supplementary Figure 15. Statistics from the exemplary image with histogram and KDE curves as functions of particle area size, aspect ratio, and relative orientation to horizontal axis.



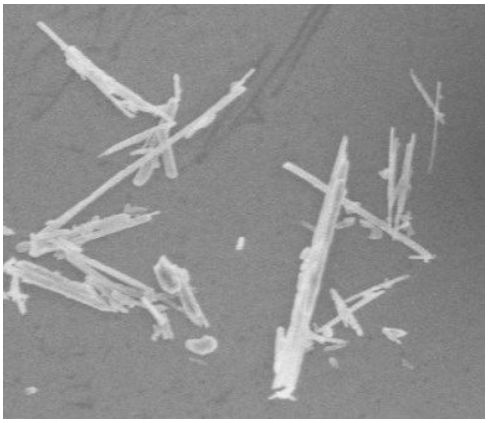
Supplementary Figure 16. Another sample STXM image on substrate.



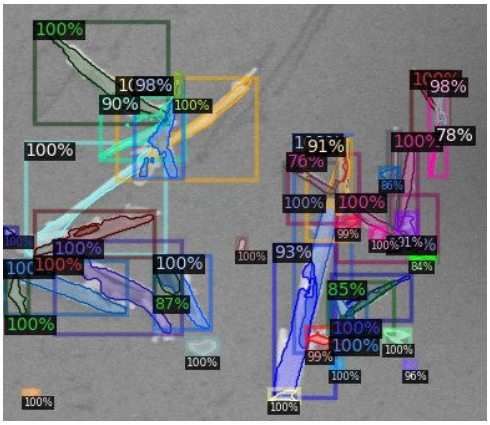
Supplementary Figure 17. Model prediction results on the sample image; The model results here are again insensitive to the substrate with black holes in the background and can predict the nanowires, including overlaps across the substrate holes. The model predicts this image to a great extent. The results are in qualitative and quantitative agreement. Little inconsistency still exists for small, densely nanowires in the left regions, where multiple particles are predicted as one. However due to the low resolution of the image, visual evaluation is difficult and will not be further elaborated.



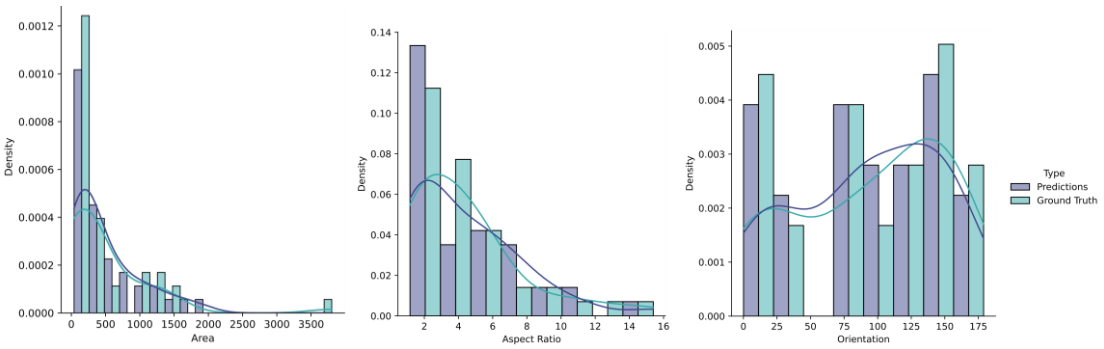
Supplementary Figure 18. Statistics from the exemplary image with histogram and KDE curves as functions of particle area size, aspect ratio, and relative orientation to horizontal axis. Higher density in the ground truth is observed in the respective number ranges due to manual separations of smaller nanowires, where the model predicts multiple nanowires as one single instance, presumably due to the comparably lower resolution of the image.



Supplementary Figure 19. A sample SEM image.



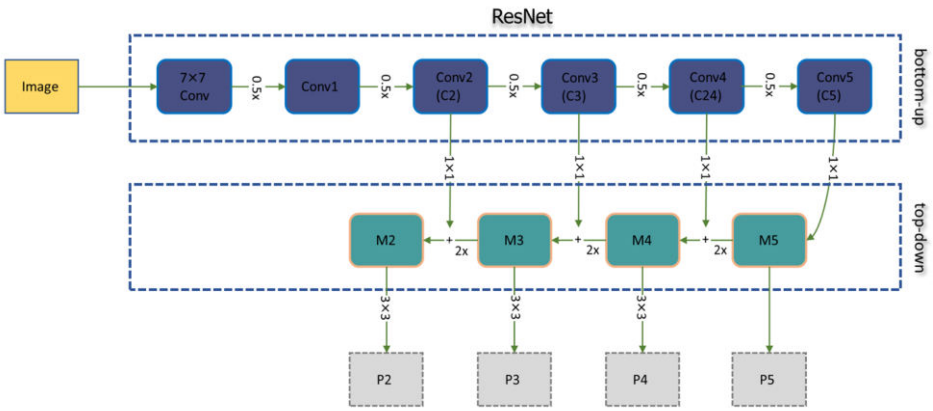
Supplementary Figure 20. Model prediction for the sample image; upon visual assessment, multiple nanowires are attached and melted together. This complicates the model prediction. The isolated, as well as cross orientated nanowires are well captured, whereas the attached nanowires are segmented as single instances and to less satisfaction.



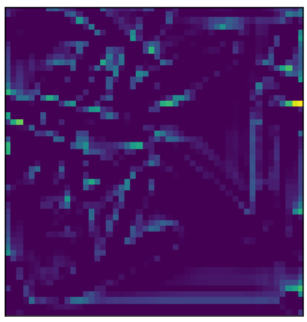
Supplementary Figure 21. Statistics from the SEM image. The statistical results show good qualitative, and quantitative agreement for the SEM image, but smoothed out in presence of many nanowires. Underestimation of the model leads to fewer number of nanowires thus leading to minor deviation in the statistics.

Supplementary Note 6: Details on the FPN Backbone

One of the main problems in object detection is detecting objects at different scales mainly for small objects. Image pyramids [2] were introduced by using the image at different scales to detect objects. This method comes with a significant cost in time and memory for end-to-end training. An alternative for this method is to use a pyramid of feature maps rather than the image, which enables object detection in a wide range of scales. These feature pyramids are considered scale-invariant since the object's scale is offset by shifting the level of the pyramid [3]. FPN is then constructed of two pathways, a bottom-up and a top-down pathway [3]. The bottom-up pathway is the well-known ConvNets, known for feature extractions, which encode the image to low-resolution feature maps with increasing semantic features and detecting high-level structures when going up (see supplementary figure 22). This part constructs the backbone of FPN which is usually different variants of Residual networks (ResNet) [4]. The top-down path constructs layers with higher spatial resolutions while combining the enriched semantic features of the bottom-up layers to lead final feature maps for further instance segmentation.



Supplementary Figure 22. General architecture of feature pyramid network with ResNet as backbone, redrawn based on [5]; The bottom-up path constructs layer with decreasing spatial scale (illustrated with 0.5x) with increasing semantic feature from C1-C5 layer, the top-down path M5-M2 constructs layers with increasing spatial resolution (illustrated with 2x) and combines the low-resolution, but high semantic features from bottom-up path (1x1) to create final feature maps (3x3), more details can be found in [3]



Supplementary Figure 23. A feature map from P3 level shown after the operation.

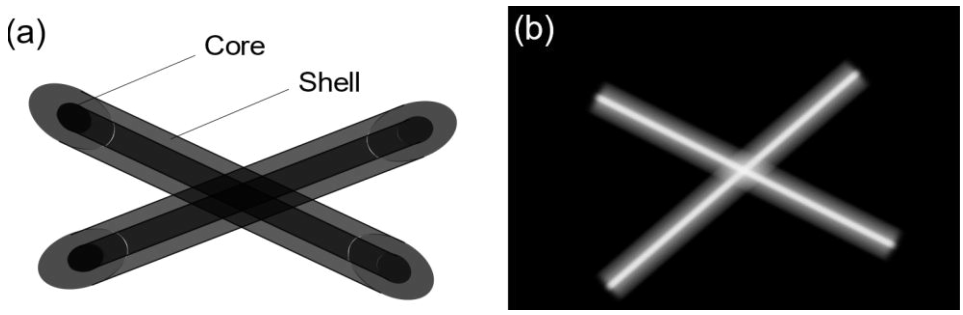
Supplementary Note 7: Additional information on model training/prediction speed

The time for generation is, after building the workflow, to a neglectable extent. The ML algorithm is purely trained on this dataset with a pre-trained network structure and backbone and took 5 to 10 hours (depending on the epoch numbers and dataset size) on four A100 Tesla cards.

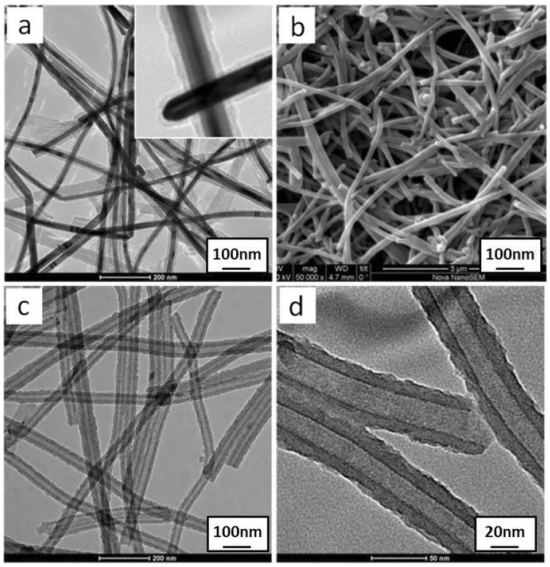
Regarding the speed performance, for comparison, ground truths are generated by human labeling, taking approx. tens of seconds for one nanowire, depending on the resolution of the image and coarseness of polygons (i.e, for a circular object, numbers of straight lines that are used to approximate the object) drawn on the instance object. The steps within human labeling using MakeSense.AI include uploading the desired image and drawing polygons around the instance. If multiple instances are available, multiple polygons are independently drawn. Ideally, the number of polygons drawn is equal to the particles visible in the image. After masking the instances, polygons are exported for further post-processing as determination of area size, orientation, or aspect ratios. The above steps would result in tens of seconds up to minutes for one image. Regarding this point, ML-method outperforms human segmentation, especially when it comes to large-scale, multi-particle, and low contrast images. The current inference speed, run on a CPU (AMD ryzen 3700x) is 3.4 seconds and on a GPU (RTX 2070 super) is 1.2 seconds plus post-processing steps of 1-2 seconds.

Supplementary Note 8: Additional information on extension of the current model to segmentation of coaxial or hierarchical structures

In principle, the framework is not bound to any specific structures if the structures can be created or synthesized. Based on the example of a STXM image, which maps the nanowires into optical intensity values, the synthetic image can be generated by creating coaxial, or hierarchical structures, for example, a core-shell nanowire of different materials, see illustration in the Supplementary Figure 23. Performing the optical density compression step as mentioned in the manuscript, we believe that our approach can be applied to real STXM, SEM images of coaxial structures as in Supplementary figure 25 [6] as well.



Supplementary Figure 24. (a) Illustration on generating synthetic data for coaxial core-shell nanowires (b) Illustration of optical intensity-based mapping step



Supplementary Figure 25. Example microscopy images of typical coaxial nanocomposites from [6]

Supplementary table 1:

AP BBOX

#	Epoch	LR	ROI head	Dataset	NMS	IOU	Synthteic Dataset						Ptychography						SXTM						SEM					
							AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
1	250	0.02	256	250	0.7	0.6	89.442	96.027	93.853	82.205	92.259	77.576	36.566	60.935	37.33	21.04	47.024	82.525	27.267	54.404	23.423	25.228	37.774	nan	27.61	51.8	22.884	20.586	43.495	38.356
2	250	0.02	256	500	0.7	0.6	91.843	97.017	95.005	84.934	94.405	83.741	39.022	63.243	41.817	21.376	54.799	68.399	29.292	52.363	28.663	22.625	49.623	nan	25.349	54.951	18.113	20.59	41.197	15.561
3	250	0.02	256	750	0.7	0.6	92.219	97.005	95.918	86.195	94.243	79.064	36.783	60.881	63.881	27.772	48.273	60.149	30.51	53.577	29.092	24.447	48.116	nan	19.367	45.737	15.484	12.204	39.725	11.275
4	250	0.02	256	1000	0.7	0.6	93.541	97.022	95.948	87.723	95.552	84.409	36.511	63.515	38.765	23.564	53.149	64.667	30.507	55.129	27.601	26.641	48.949	nan	15.253	34.259	11.702	10.47	31.856	7.378
5	500	0.02	256	250	0.7	0.6	90.949	96.024	93.988	82.957	93.415	80.512	31.932	53.916	34.5	26.089	42.091	63.432	24.392	47.339	22.543	18.969	39.851	nan	27.916	54.302	21.149	22.776	43.195	16.311
6	500	0.02	256	500	0.7	0.6	92.831	97.019	95.912	86.283	95.009	84.963	36.68	59.307	37.97	29.876	48.056	65.05	28.091	51.49	24.893	24.28	43.815	nan	18.379	40.882	14.003	12.277	36.136	9.714
7	500	0.02	256	750	0.7	0.6	93.336	97.023	95.997	86.688	95.672	83.057	36.738	60.179	39.845	31.98	46.715	36.279	29.266	55.557	22.886	22.631	50.123	nan	15.901	34.044	15.966	10.129	35.856	1.889
8	500	0.02	256	1000	0.7	0.6	93.878	97.125	96.005	86.821	95.896	82.152	39.854	60.198	93.785	31.887	46.914	33.843	29.656	53.525	21.658	23.652	50.856	nan	13.522	30.253	12.983	9.32	29.444	0.938
9	750	0.02	256	250	0.7	0.6	92.219	96.984	95.918	86.196	94.243	79.064	35.783	60.953	36.881	27.772	48.273	60.149	28.293	51.763	29.092	24.447	48.116	nan	19.376	45.737	15.484	12.204	39.725	11.275
10	750	0.02	256	500	0.7	0.6	92.82	97.007	95.961	86.542	94.863	84.029	36.718	56.638	31.411	24.911	51.581	54	28.394	49.442	27.977	17.437	49.245	nan	19.926	41.958	14.748	13.234	38.578	9.752
11	750	0.02	256	750	0.7	0.6	93.454	97.02	95.962	86.821	95.351	85.507	28.577	52.753	31.009	23.564	44.562	42.272	28.339	51.366	26.156	22.658	45.223	nan	11.326	24.101	8.3969	8.529	29.034	0.717
12	750	0.02	256	1000	0.7	0.6	94.002	97.026	95.956	86.326	95.667	84.021	36.665	68.513	34.661	23.564	54.45	35.033	28.624	50.411	29.483	23.792	45.881	nan	26.324	53.757	22.953	21.896	39.353	12.192
13	500	0.01	256	750	0.7	0.6	93.564	97.012	95.97	86.973	95.441	86.751	39.145	64.638	43.965	25.248	52.753	72.525	27.01	48.58	22.643	19.418	46.104	nan	21.561	47.927	15.718	14.011	44.587	15.317
14	500	0.03	256	750	0.7	0.6	93.279	96.039	95.947	87.506	95.481	81.511	38.584	63.618	42.159	25.248	44.998	75.05	28.907	53.598	26.712	20.984	49.943	nan	23.177	52.523	17.977	16.119	43.255	15.736
15	500	0.02	512	750	0.7	0.6	92.582	97.023	95.978	86.95	95.538	87.127	36.741	67.437	35.131	29.175	51.153	43.012	27.892	49.67	25.994	19.424	50.432	nan	16.338	34.205	14.156	10.659	36.278	7.238
16	500	0.02	128	750	0.7	0.6	93.513	97.017	95.988	85.892	94.657	86.631	37.48	60.69	40.679	22.554	56.39	35.929	28.599	52.531	26.215	23.129	45.605	nan	18.921	38.46	17.457	12.908	38.247	5.173
17	500	0.02	256	750	0.6	0.6	93.278	97.02	94.93	86.104	95.206	81.588	38.115	62.294	41.914	20.759	52.149	70.875	26.236	49.788	22.625	21.202	45.238	nan	15.005	31.237	16.028	8.212	36.859	2.085
18	500	0.02	256	750	0.8	0.6	93.523	97.023	96.006	87.039	95.79	87.896	39.941	64.682	42.285	21.46	55.349	72.525	29.77	55.562	25.961	22.095	50.669	nan	17.189	38.743	15.537	12.119	35.377	4.376
19	500	0.02	256	750	0.7	0.7	92.902	97.015	95.965	86.965	95.274	87.262	41.089	74.603	43.25	28.34	52.772	62.624	27.818	48.405	25.991	20.793	48.555	nan	18.431	41.861	11.268	14.121	35.727	0.9
20	500	0.02	256	750	0.7	0.8	93.58	97.026	95.991	86.996	95.787	88.464	31.981	58.498	26.728	17.337	48.706	57.673	28.36	49.874	26.712	20.907	47.699	nan	21.12	40.879	17.512	15.214	38.266	3.433
21*	500	0.001	256	750	0.7	0.6	89.743	93.198	92.154	83.139	91.624	82.917	35.215	60.759	40.045	21.29	48.851	68.662	23.056	44.669	18.68	15.449	42.188	nan	17.596	44.015	11.741	10.031	40.668	11.34
22*	500	0.005	256	750	0.7	0.6	90.951	93.719	93.627	85.161	93.16	79.148	36.092	61.201	39.678	22.716	42.525	72.667	26.386	51.151	24.184	18.439	47.485	nan	20.639	50.073	15.423	13.56	40.777	13.175

AP SEGM MASK

#	Epoch	LR	ROI head	Dataset	NMS	IOU	Synthteic Dataset						Ptychography						SXTM						SEM					
							AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
1	250	0.02	256	250	0.7	0.6	87.392	96.013	93.628	86.179	88.027	89.052	38.091	57.648	37.33	18.274	47.283	85.05	21.791	51.877	17.587	20.862	28.877	nan	12.927	23.126	5.206	7.842	22.443	36.634
2	250	0.02	256	500	0.7	0.6	88.878	69.954	94.82	88.32	86.293	92.174	36.682	57.232	36.188	14.107	48.029	75.05	20.473	50.162	14.137	15.407	33.1	nan	11.961	35.639	5.829	8.677	18.911	22.739
3	250	0.02	256	750	0.7	0.6	89.568	96.981	95.715	89.559	86.648	91.193	34.089	57.013	27.97	17.842	48.651	60.099	21.36	52.172	15.618	17.329	32.782	nan	10.328	28.021	4.877	7.135	19.056	11.966
4	250	0.02	256	1000	0.7	0.6	90.195	96.992	95.771	91.006	89.808	93.938	36.524	59.765	35.295	19.637	51.905	65.05	21.2	52.661	16.034	19.276	30.743	nan	9.301	25.165	2.936	6.892	17.568	3.425
5	500	0.02	256	250	0.7	0.6	87.908	95.992	93.719	86.667	88.43	91.579	32.219	52.743	34.5	15.509	44.628	65.001	19.879	47.427	19.375	15.385	31.82	nan	12.688	37.916	5.643	8.337	21.884	25.743
6	500	0.02	256	500	0.7	0.6	89.872	96.998	95.8	89.905	89.823	94.133	35.259	61.106	37.29	23.564	46.789	60	19.95	49.047	15.129	17.568	29.637	nan	10.359	27.54	4.811	7.361	18.576	12.15
7	500	0.02	256	750	0.7	0.6	90.034	97.012	95.741	91.053	89.933	91.249	39.089	57.955	35.984	31.98	46.773	50.098	19.775	49.932	10.217	16.431	29.575	nan	8.924	23.383	6.415	7.473	14.864	0.612
8	500	0.02	256	1000	0.7	0.6	90.254	97.465	94.566	90.658	88.483	90.663	38.65	58.2	36.722	29.547	45.223	51.426	19.061	50.78	10.147	16.923	28.182	nan	7.058	22.537	3.222	6.452	10.538	0.242
9	750	0.02	256	250	0.7	0.6	89.568	96.715	95.715	89.559	89.648	91.193	34.089	57.013	27.97	17.842	48.651	60.099	21.049	48.671	17.437	16.013	34.714	nan	10.328	28.021	4.877	7.135	19.056	11.966
10	750	0.02	256	500	0.7	0.6	90.095	96.982	94.882	90.368	90.161	89.888	33.278	56.317	33.768	14.107	48.449	70	19.873	45.644	15.834	17.399	28.432	nan	10.853	27.68	5.584	7.516	19.168	10.099
11	750	0.02	256	750	0.7	0.6	89.958	96.991	94.769	90.774	90.074	91.574	29.491	49.655	25.091	12.504	43.97	62.525	19.233	44.349	9.518	14.963	30.885	nan	7.317	18.645	3.764	5.41	17.358	0.002
12	750	0.02	256	1000	0.7	0.6	90.487	96.967	95.795	90.871	89.986	88.59	35.906	63.17	29.384	18.178	50.455	40.099	20.354	47.83	14.392	15.522	32.605	nan	11.688	38.58	6.838	11.008	22.567	31.733
13	500	0.01	256	750	0.7	0.6	89.72	95.991	94.802	90.439	89.882	91.022	42.327	62.519	39.964	22.442	58.03	85.05	21.324	48.669	16.46	15.498	35.096	nan	12.659	35.553	5.088	8.089	23.695	12.871
14	500	0.03	256	750	0.7	0.6	89.87	96.038	95.843	90.545	90.025	88.21	41.662	62.803	42.159	26.931	50.644	70.099	21.029	48.945	19.856	14.98	36.532	nan	11.36	31.898	2.579	8.034	20.676	10.099
15	500	0.02	512	750	0.7	0.6	89.614	97.014	94.856	90.267	89.942	92.049	36.782	60.295	31.811	24.238	50.463	47.574	18.863	46.696	14.206	13.922	31.692	nan	9.888	26.11	5.03	7.517	18.414	7.855
16	500	0.02	128	750	0.7	0.6	89.932	96.963	95.815	89.778	89.717	91.271	37.754	65.675	33.763	20.559	53.89	52.525	20.509	45.412	16.023	15.744	33.161	nan	9.806	27.939	5.053	8.343	16.855	1.327
17	500	0.02	256	750	0.6	0.6	89.945	96.927	95.669	90.514	89.986	87.904	38.103	61.643	35.552	17.168	54.463	65.05	17.886	48.337	11.576	14.96	27.923	nan	9.876	25.43	5.222	6.854	19.187	1.01
18	500	0.02	256	750	0.8	0.6	90.139	96.966	95.819	89.955	89.685	93.658	40.463	69.542	45.256	16.495	56.477	62.525	19.856	47.661	12.99	16.038	30.639	nan	10.157	28.263	3.212	7.478	19.99	4.208
19	500	0.02	256	750	0.7	0.7	89.952	96.995	95.842	90.513	89.73	93.434	41.719	67.042	40.438	19.785	52.774	70.099	20.102	45.413	16.213	15.34	33.946	nan	10.121	29.123	4.612	7.096	20.093	0.24
20	500	0.02	256	750	0.7	0.8	89.74	97.002	95.871	90.513	90.134	91.604	31.322	54.62	26.728	12.504	48.132	55.149	20.544	47.769	13.7	16.793	31.269	nan	11.078	31.909	5.388	8.323	19.667	3.084
21*	500	0.001	256	750	0.7	0.6	85.891	92.175	90.984	86.612	86.054	87.196	34.904	58.636	36.036	18.479	54.138	66.231	17.359	44.758	12.485	11.521	31.158	nan	8.676	31.616	1.09	4.097	19.734	8.889
22*	500	0.005	256	750	0.7	0.6	87.532	93.718	93.523	88.209	87.687	85.867	36.179	60.384	39.678	24.404	48.188	67.701	18.484	46.484	17.308	12.417	34.034	nan	8.786	29.386	0.054	5.45	18.13	7.521

Supplementary reference:

1. OpenCV documentation, https://docs.opencv.org/4.x/d3/db4/tutorial_py_watershed.html.
2. Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J. & Ogden, J. M. Pyramid methods in image processing. *RCA engineer* 29, 33–41 (1984).
3. Lin, T.-Y. et al. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125 (2017).
4. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
5. Jonathan Hui. Understanding Feature Pyramid Networks for object detection (FPN) https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c
6. Zhu, Hongmei, and Xuchuan Jiang. Development of a General Fabrication Strategy for Carbonaceous Noble Metal Nanocomposites with Photothermal Property. *Nanoscale research letters* 15.1, 1-11 (2020).