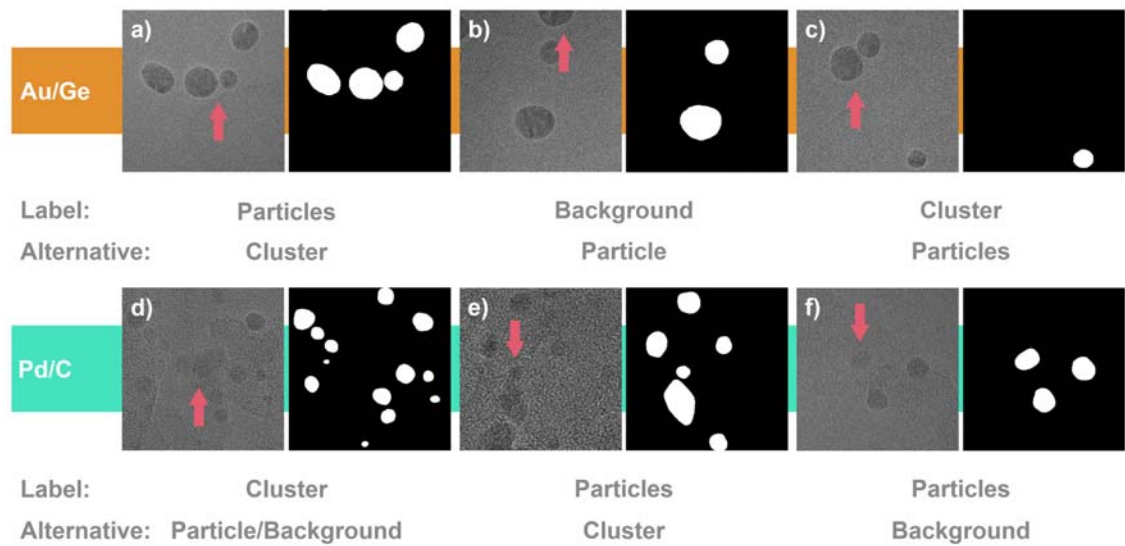
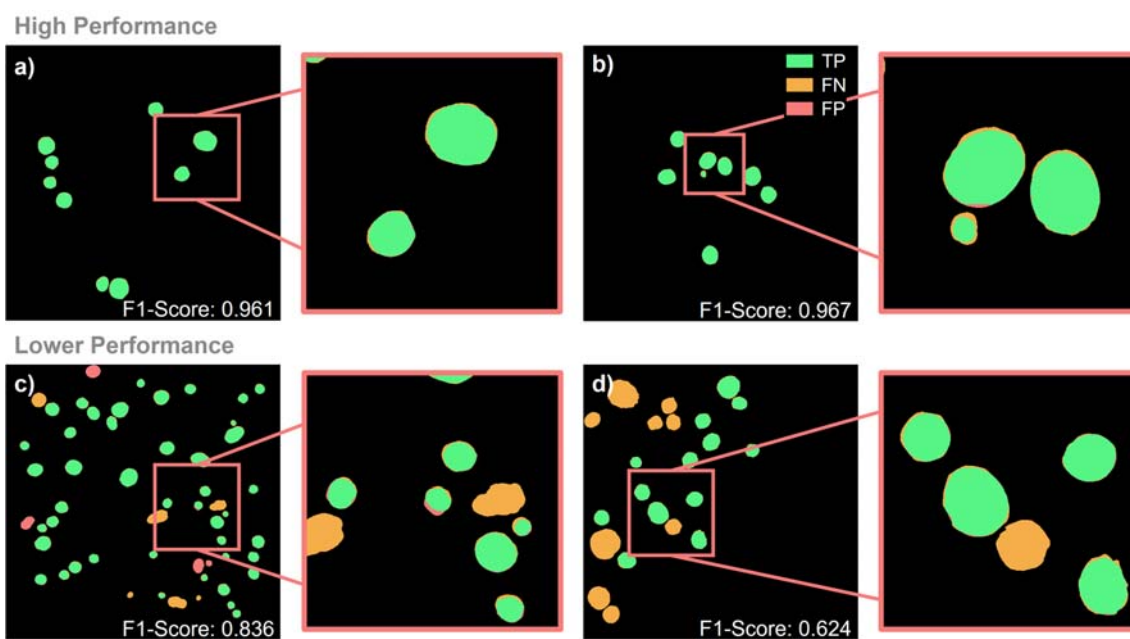


Supplementary Table 1: Post-processing steps applied to the 256 x 256 SegNet output for full inference reconstruction in Mathematica 12.1.

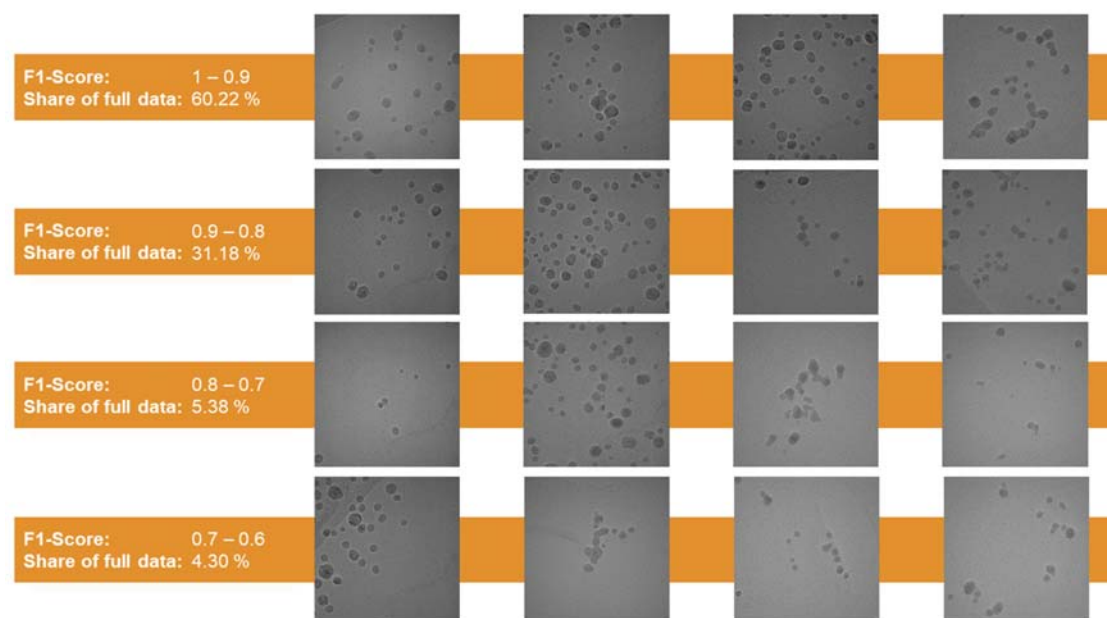
Au/Ge	Pd/C
i) Binarisation; thres. = 0.999	i) Binarisation, thres. = 0.999
ii) Delete small components	ii) Water-shedding
iii) Delete border components	iii) Dilation; $r_{Diskmatrix} = 7$
iv) Filling transform	iv) Delete border components
	v) Filling transform
	vi) Erosion; $r_{Diskmatrix} = 7$
	vii) Circularity Filtering; thres. = 0.8



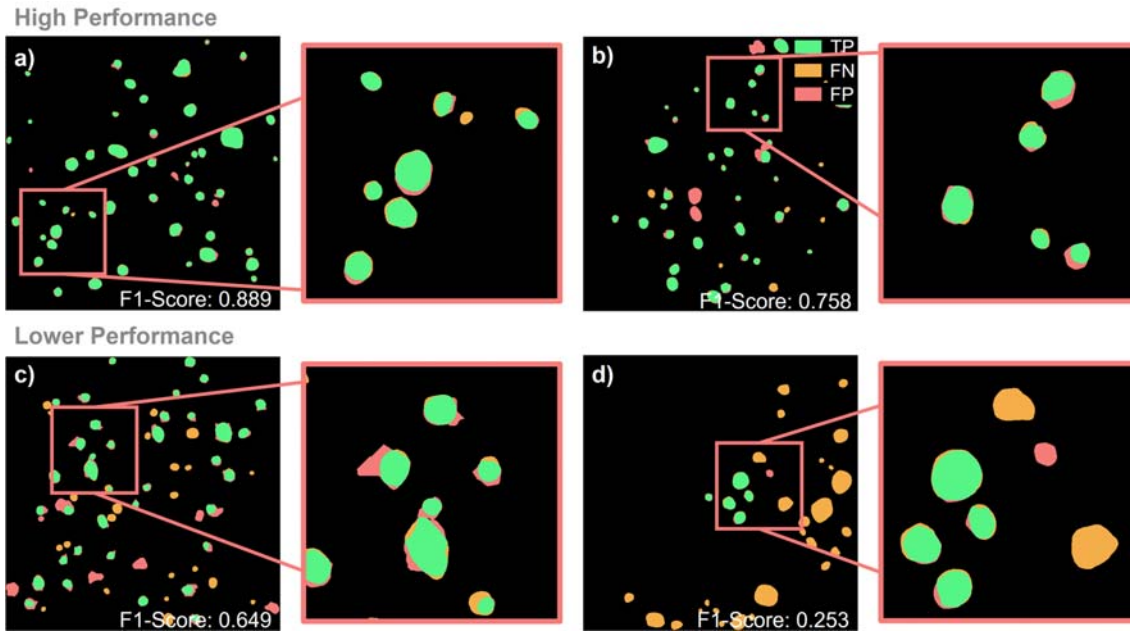
Supplementary Figure 1: Crops of experimental images with respective ground truth labels. a) – c) Examples of the Au/Ge system and **d) – f)** Examples of the Pd/C system. All cases show labelling scenarios for which subjective judgement was required. The given and potential alternative labels are added below each example, while red arrows highlight the instance in question. **a)** An instance labelled as two separate nanoparticles, whereas a ‘cluster’ label is an alternative as particles are in very close proximity. **b)** Particle cut by the edge of the image field of view. In alternative approaches, a positive label could equally be assigned. **c)** The label ‘cluster’ as a clear contact area was identified. **d)** Instance with ambiguous contrast. In this case, the label ‘cluster’ was applied, whereas splitting as ‘particle’ and ‘background’ were potential alternatives. **e)** An instance labelled as separate nanoparticles. **f)** The instance in question was given the label ‘particle’ although the respective contrast is lower than in other examples. Decisions of this nature were most common for the Pd/C dataset.



Supplementary Figure 2: Colour-coded segmentation masks for the Au/Ge system. A corresponding figure with respect to Fig. 2. Parts of the segmentation output is enlarged to highlight results discussed in the main text.



Supplementary Figure 3: Selected examples of experimental HRTEM images in the Au/Ge system that were assessed at indicated F1-Scores.



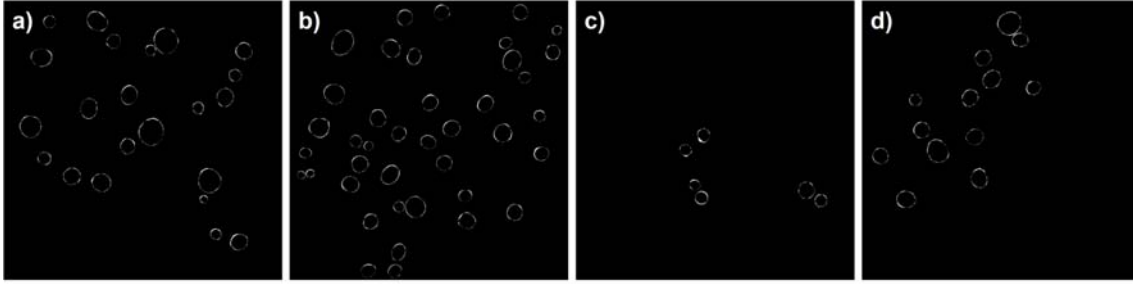
Supplementary Figure 4: Colour-coded segmentation masks for the Pd/C system. A corresponding figure with respect to Fig. 3. Parts of the segmentation output is enlarged to highlight results discussed in the main text.

Supplementary Table 2: Comparison of KDE-plots between ground truth (GT) and inferred datasets (Inf.) for the Au/Ge system. Individual F1-Scores refer to plots in Figure 5 of the main document.

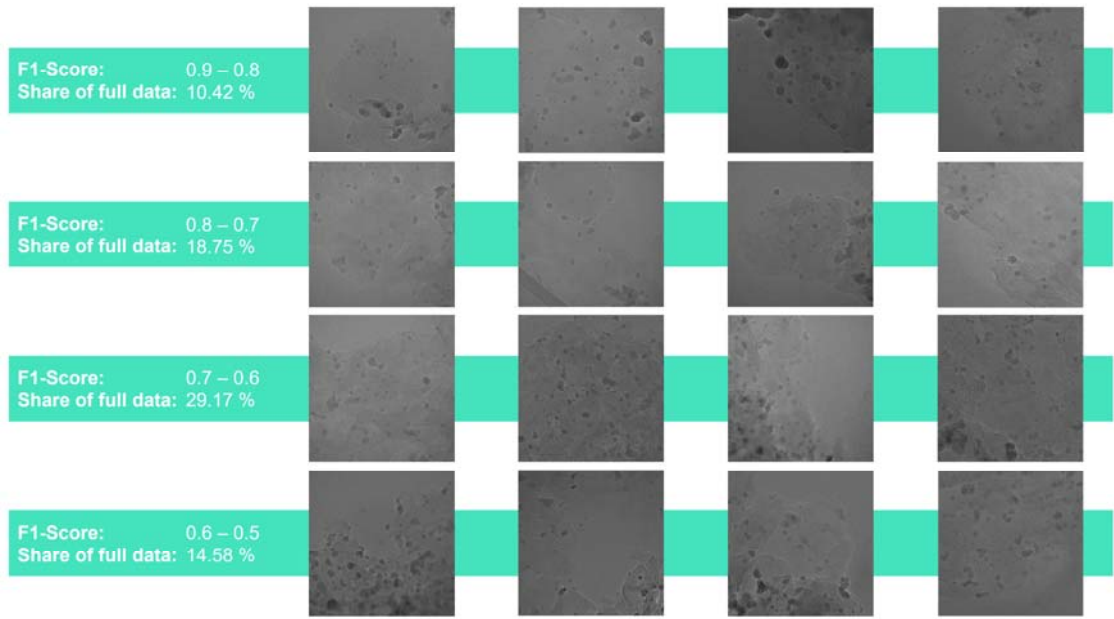
F1-Score	1-0.9		1-0.8		1-0.7		1-0.6	
KDE prop.	x_{ymax}	FWHM	x_{ymax}	FWHM	x_{ymax}	FWHM	x_{ymax}	FWHM
Inf.	51.16	10.15	49.79	21.15	49.79	21.45	49.79	21.89
GT	56.62	9.50	56.91	19.73	56.91	19.94	56.91	20.48

Supplementary Table 3: Error estimation of pure border mismatch on individual nanoparticles in the Au/Ge system between inferred images and ground truth images based on determined nanoparticle areas in nm².

F1-Score	1-0.9	0.9-0.8	0.8-0.7	0.7-0.6
Inference	76.53	68.65	62.77	63.99
Ground Truth	84.35	76.34	68.35	70.18
Error [%]	9.27	10.07	8.16	8.82



Supplementary Figure 5: Exemplar images drawn from different F1-Score intervals for border mismatch comparison in the Au/Ge system. F1-Score intervals **a)** [0.9, 1], **b)** [0.8, 0.9], **c)** [0.7, 0.8], **d)** [0.6, 0.7]. Respective images show differences between ground truth and inference images whereby only instances mostly consisting of TP contributions were considered to isolate the boundary mismatch error.



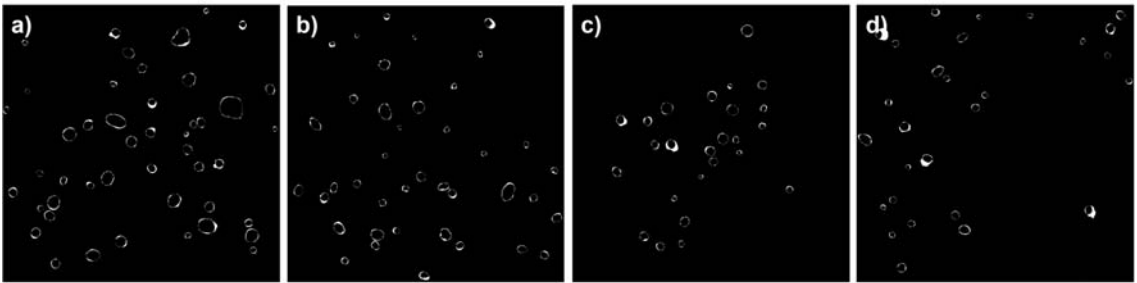
Supplementary Figure 6: Selected examples of experimental HRTEM images in the Au/Ge system that were assessed at the indicated F1-Scores.

Supplementary Table 4: Comparison of KDE-plots between ground truth (GT) and inferred datasets (Inf.) for the Au/Ge system. Individual F1-Scores refer to plots in Figure 5 of the main document.

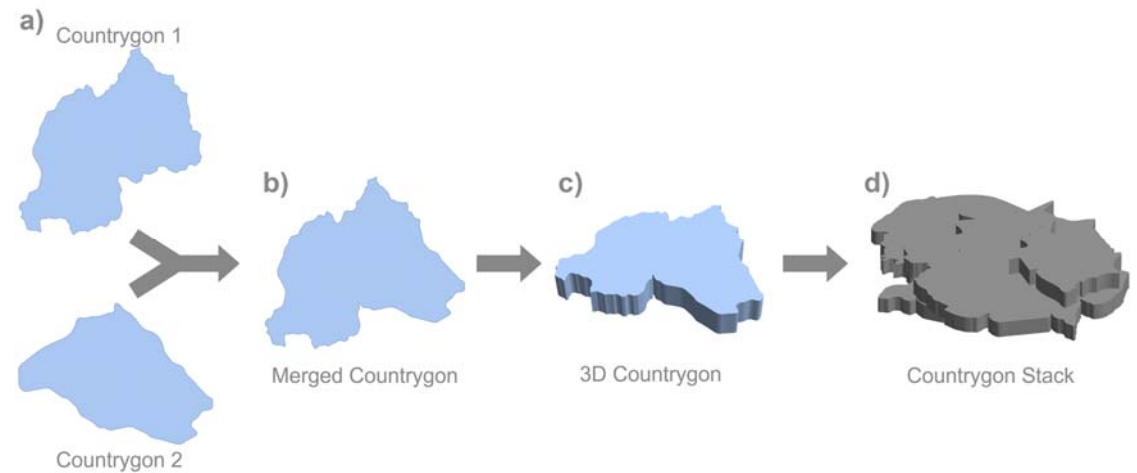
F1-Score	0.9-0.8		0.9-0.7		0.9-0.6		0.9-0.5	
KDE prop.	x_{ymax}	FWHM	x_{ymax}	FWHM	x_{ymax}	FWHM	x_{ymax}	FWHM
Inf.	17.06	38.71	13.23	34.80	13.23	32.69	13.23	32.69
GT	14.36	32.76	10.50	27.67	10.45	24.08	10.45	24.08

Supplementary Table 5: Error estimation of pure border mismatch on individual nanoparticles in the Pd/C system between inferred images and ground truth images based on determined nanoparticle areas in nm².

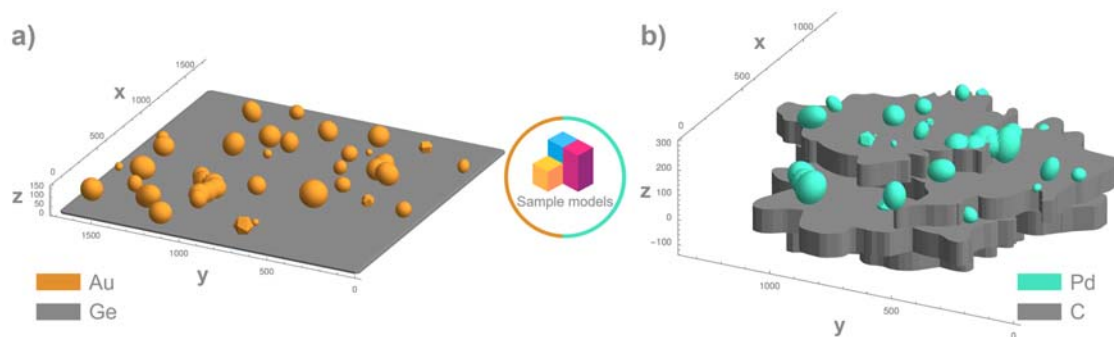
F1-Score	0.9-0.8	0.8-0.7	0.7-0.6	0.6-0.5
Inference	30.37	22.87	28.31	27.30
Ground Truth	29.16	23.70	25.18	23.58
Error [%]	4.14	3.50	12.43	15.77



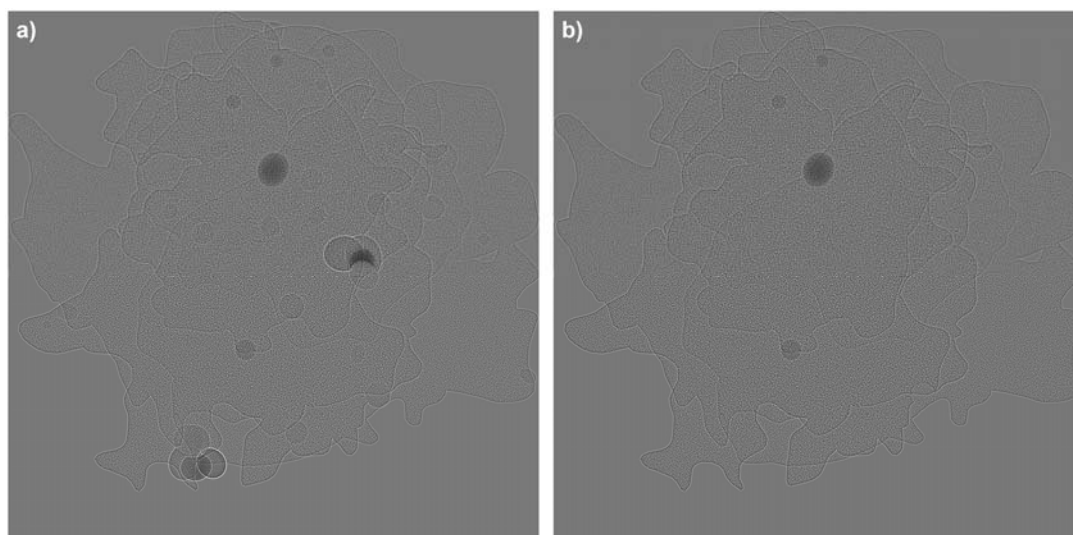
Supplementary Fig. 7: Exemplar images drawn from different F1-Score intervals for border mismatch comparison of the Pd/C system. F1-Score intervals **a)** [0.8, 0.9], **b)** [0.7, 0.8], **c)** [0.6, 0.7], **d)** [0.5, 0.6]. Respective images show differences between ground truth and inference images whereby only instances mostly consisting of TP contributions were considered to isolate the boundary mismatch error.



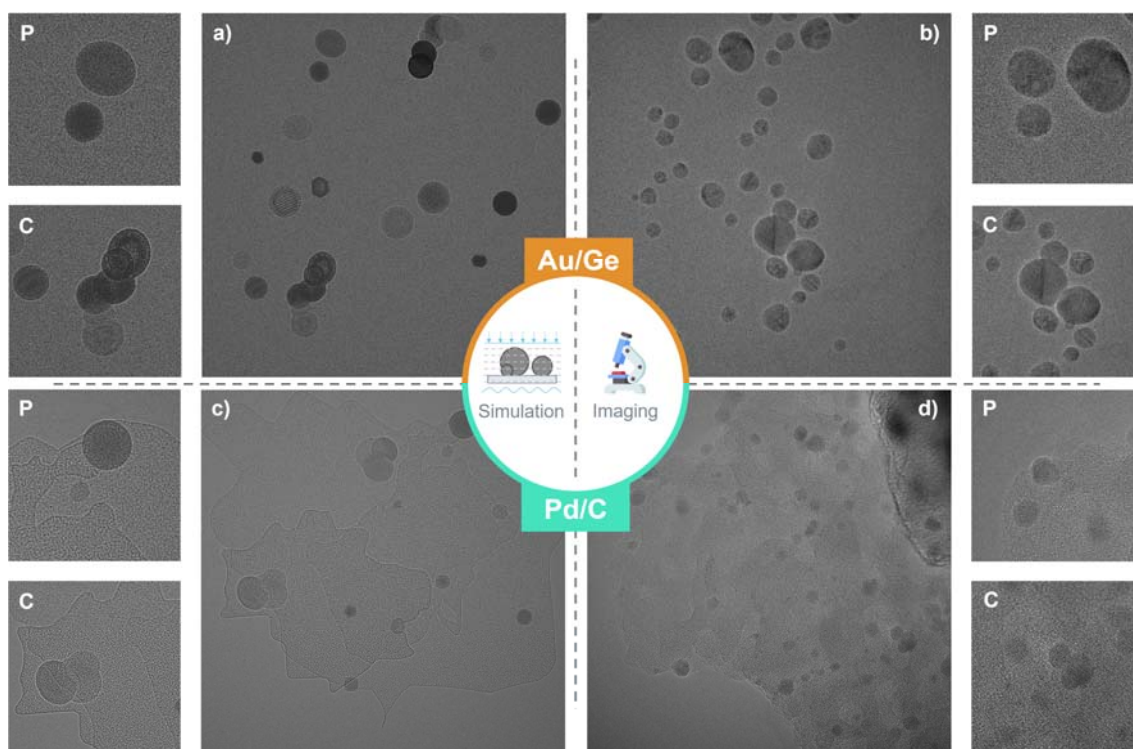
Supplementary Figure 8: Schematic workflow depicting the generation of the supporting material used in Pd/C models. **a)** Two individually generated Countrygons. **b)** New Countrygon generated by overlaying Countrygons from the previous step. **c)** Linear extension in of Countrygon into third dimension. **d)** Countrygon stack generated from up to five 3D Countrygon layers as seen in previous step.



Supplementary Figure 9: 3D convex hull representations of the sample models generated in Mathematica 12.1. a) The Au/Ge system where multiple nanoparticles and clusters were placed on a flat amorphous carbon film. **b)** The Pd/C system is shown where multiple 'Countrygon' layers were stacked to create a realistic model matching experimental images in projection. Nanoparticles were also put on the side of the supporting material in the Pd/C system. ***



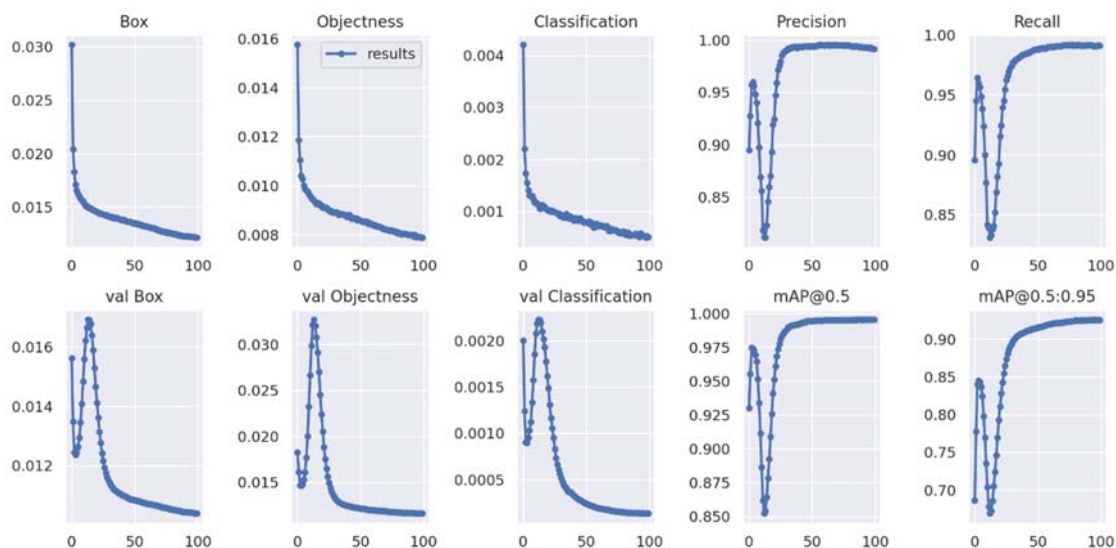
Supplementary Figure 10: Multislice image simulations before and after application of the GLD criterion. Image of the Pd/C sample model a) before the application of GLD criterion. **b)** after application of the GLD criterion.



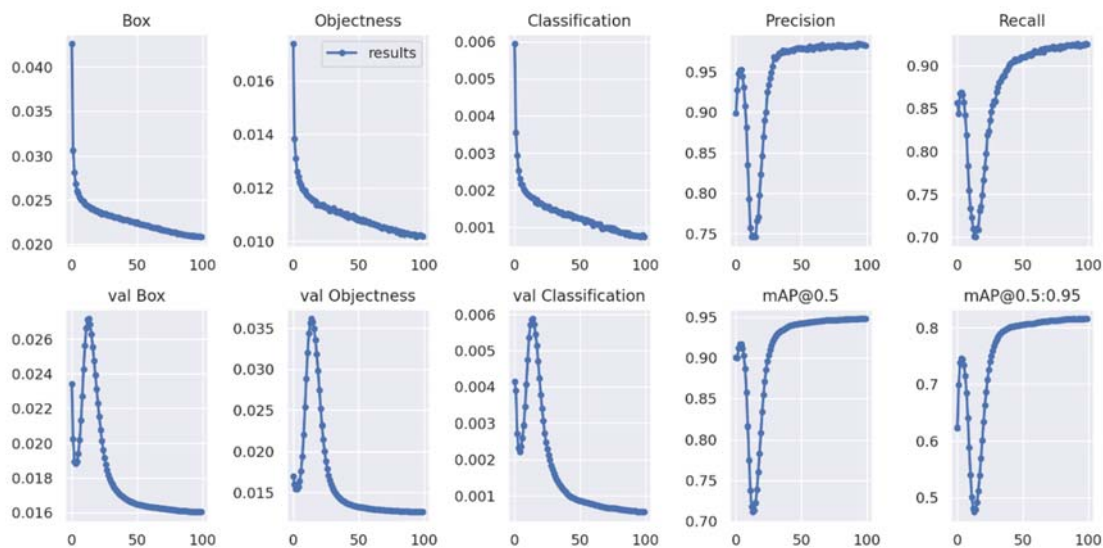
Supplementary Figure 11: Side-by-side comparison of typical simulations used as training data and experimental images. a) Image simulation for the Au/Ge system. **b)** Selected experimental image of the Au/Ge specimen. **c)** Image simulation for the Pd/C system. **d)** Experimental image of the Pd/C catalyst. Individual labels 'P' and 'C' indicate examples of particles and clusters.

Supplementary Table 6: Hyperparameters used in the training stage of the respective CNN modules.

	Object Detection module		Segmentation module	
Hyperparameters	lr0:	0.001		
	lrf:	0.2		
	momentum:	0.937		
	weight_decay:	0.0005		
	warmup_epochs:	3.0		
	warmup_momentum:	0.8		
	warmup_bias_lr:	0.1		
	box:	0.05		
	cls:	0.5		
	cls_pw:	1.0		
	obj:	1.0		
	obj_pw:	1.0		
	iou_t:	0.2		
	anchor_t:	4.0	Learning rate:	0.001
	fl_gamma:	0.0	Optimiser:	ADAM
	hsv_h:	0.0	Batch-size:	16
	hsv_s:	0.0	Train/Val split:	90/10 %
	hsv_v:	0.0		
	degrees:	0.0		
	translate:	0.1		
	scale:	0.5		
	shear:	0.0		
	perspective:	0.0		
	flipud:	0.0		
	fliplr:	0.0		
	mosaic:	0.0		
	mixup:	0.0		
	Epochs:	100		
	Train/Val split:	90/10 %		
	Optimiser:	ADAM		
	Batch-size:	8		



Supplementary Figure 12: Training results for YOLOv5x for the Au/Ge dataset.



Supplementary Figure 13: Training results for YOLOv5x for the Pd/C dataset.

Supplementary Table 7: Details for the SegNet training stage for both material systems.

	Au/Ge		Pd/C	
SegNet training details	Epochs:	82	Epochs:	71
	Training time:	211 h	Training time:	229 h
	Final train error:	$2.77 \cdot 10^{-3}$	Final train error:	$1.73 \cdot 10^{-3}$
	Final val error:	$3.27 \cdot 10^{-3}$	Final val error:	$2.32 \cdot 10^{-3}$

Supplementary Methods 1

Fast deployment guide for *nNPipe*

A brief hands-on manual for running the *nNPipe* method on the provided Au/Ge and Pd/C dataset to access statistically significant metrics. Written by Kevin P Treder.

In the following, the Au/Ge dataset is taken as example. Note that similar steps need to be executed for the Pd/C dataset, whereby corresponding weights, notebooks and files provided in <https://zenodo.org/record/7024893#.Y2tv43bMJPY> are ought to be used.

1) Preparing the Object Detection module

- i) Clone or download the yolov5 repository on <https://github.com/ultralytics/yolov5>.
- ii) Use files from 'nNPipe_resources/neural_nets/yolov5/train_files' to replace the files in the 'ultralytics/yolov5-master folder'. All files have similar names, whereby they can be found in different sub-folders. For audetect.yaml and pddetect.yaml, simply copy them into the 'yolov5-master/data' folder.
- iii) Within the 'yolov5-master/data' folder, create a folder named 'weights' and paste the trained weights provided in 'nNPipe_resources/neural_nets/yolov5/weights'.
- iv) Copy the 'nNPipe_resources/experimental_images/au_ge/img' folder into the 'yolov5-master/data' folder. Note that the destination is only a suggestion.
- v) Make sure to create a virtual environment in which requirements for running yolov5 are met. More info can be found on <https://github.com/ultralytics/yolov5>.

2) Running the Object Detection Inference

i) Once set up, you can run the inference. Open a terminal, direct to the yolov5-master folder and activate your virtual environment.

ii) To replicate similar results as discussed in the original paper, run the following command:

```
python detect.py --source data/experimental_images/au_ge/img' --weights  
data/weights/au_weights.pt --conf-thres 0.5 --save-txt --device 0 --name custom_name1
```

Note that the tag custom_name1 is a placeholder for a folder name of your preference. This command will not run if the yolov5x.yaml file was not replaced as explained in 1.ii). An in-depth explanation of the individual parser arguments is given in the detect.py file. An output folder is created automatically with its path given in the terminal. It should contain images with bounding boxes, as well as a separate folder with corresponding '.txt.' files with the bounding box data given in yolo-format ([x,y,w,h], whilst x,y, denote the centre of the bounding box).

3) Running the SegNet module

This step currently requires the Wolfram Mathematica 12.1 software.

i) Open the 'nNPipe_resources/neural_nets/nNPipe/2k_Reconstruction_nNPipe_AuGe.nb' notebook in Mathematica. Replace the image and boxes directory in:

```
Image directory  
  
ImgDir = "/mylabeldir/expX";  
  
Boxes directory  
  
ObDectDir = "/mylabeldir/expX";  
  
In[ ]:= DataImporter[ImgDir_, BoxDir_, Pos_] :=  
Block[{imgls, imgfile, boxfile, imgname, boxfilename, img},  
  imgls = FileNames[ImgDir <> "*.tif"];  
  imgname = imgls[[Pos]];  
  boxfilename =  
    BoxDir <> "/" <> StringTrim[StringSplit[imgname, "/"]][[-1]], ".tif" <> ".txt";  
  img = Import[imgname];  
  boxfile = Import[boxfilename, "Data"];  
  {img, imgname, boxfile, boxfilename}  
]
```

with paths where the **raw experimental Au/Ge HRTEM images** and the **inferred bounding boxes** are deposited respectively. In this example, the former can be found in 'nNPipe_resources/experimental_images/au_ge/img', while the latter was generated in 2.ii).

Subsequently, run the DataImporter function. Note that this function is sensitive to '/' characters at the end of the path, often being the reason behind error messages.

ii) Load the network by passing the correct path and file in:

Load the network:

```
NetDir =  
    "/myweightsdir/";  
  
AuNet = Import[NetDir <> "/" <> "au_segnet_weights.wlnet"]
```

Out[]= NetChain[

In this example, the corresponding notebook does also contain a quick demo to visualise how each detected nanoparticle is processed as 256 x 256 pixel image crop. The main function required to generate the full 2048 x 2048 pixel output is established by the AssembleAU function. Executing it, setting up the desired directory (i.e. SetDirectory function to direct) and subsequently running the NNSegmentation function will process all experimental Au/Ge images.