

## ARTICLE OPEN



# Super-resolving microscopy images of Li-ion electrodes for fine-feature quantification using generative adversarial networks

Orkun Furat<sup>1</sup>, Donal P. Finegan<sup>2</sup>, Zhenzhen Yang<sup>3</sup>, Tom Kirstein<sup>1</sup>, Kandler Smith<sup>2</sup> and Volker Schmidt<sup>1</sup>

For a deeper understanding of the functional behavior of energy materials, it is necessary to investigate their microstructure, e.g., via imaging techniques like scanning electron microscopy (SEM). However, active materials are often heterogeneous, necessitating quantification of features over large volumes to achieve representativity which often requires reduced resolution for large fields of view. Cracks within Li-ion electrode particles are an example of fine features, representative quantification of which requires large volumes of tens of particles. To overcome the trade-off between the imaged volume of the material and the resolution achieved, we deploy generative adversarial networks (GAN), namely SRGANs, to super-resolve SEM images of cracked cathode materials. A quantitative analysis indicates that SRGANs outperform various other networks for crack detection within aged cathode particles. This makes GANs viable for performing super-resolution on microscopy images for mitigating the trade-off between resolution and field of view, thus enabling representative quantification of fine features.

*npj Computational Materials* (2022)8:68; <https://doi.org/10.1038/s41524-022-00749-z>

## INTRODUCTION

Imaging techniques like, for example, scanning electron microscopy (SEM), electron backscatter diffraction (EBSD), and X-ray microtomography (micro-CT) have emerged as powerful tools for characterizing the microstructure of various kinds of materials<sup>1–4</sup>, for investigating structure–property relationships<sup>5,6</sup>, and for analyzing the influence of manufacturing parameters on the resulting materials' microstructures<sup>7</sup>. In addition, methods from machine learning have developed rapidly in recent years, especially for computer vision tasks. These developments include methods for image classification<sup>8,9</sup>, segmentation<sup>10–12</sup>, and synthesis<sup>13–16</sup>. Increasingly intensive, these methods are adapted and applied for solving similar tasks within the field of materials science. For example, modified versions of the convolutional neural network (CNN) architecture, so-called U-nets, which have been developed for the segmentation of biomedical image data in Ronneberger et al. (2015)<sup>12</sup>, found numerous applications within materials science for image segmentation tasks<sup>1,17–22</sup>. Breakthroughs in this direction are of great importance, since the quality of segmentation results has a significant impact on subsequent analyses like the statistical characterization of materials<sup>1–3</sup>, the calibration of stochastic geometry models for the generation of digital twins, i.e., the generation of virtual but realistic microstructures<sup>23–25</sup>, and conducting numerical simulations of effective materials properties<sup>26,27</sup>.

Moreover, methods from machine learning are not limited to segmentation tasks within materials science. In fact, neural networks are able to perform the previously mentioned subsequent analyses steps as well. For example, Mianroodi et al. (2021)<sup>28</sup> reported that a trained U-net architecture can predict a microstructure's local stress fields faster than spectral solvers of the associated partial differential equations. Furthermore, methods from machine learning have been used for the generation of

digital twins of real microstructures<sup>29–31</sup>. These studies deployed so-called generative adversarial networks (GANs) which are typically trained in an unsupervised way and have first been deployed for image synthesis tasks<sup>13</sup>. Supervised versions of GANs have been used for performing super-resolution, i.e., for enhancing the resolution of digital images<sup>32–34</sup>. Furthermore, GANs can be trained also in unsupervised scenarios, i.e., when training data consists of non-matching pairs of low-resolution and high-resolution images<sup>35,36</sup>. Besides GANs, there are further methods from machine learning for performing super-resolution<sup>37–39</sup>—for an exhaustive survey on super-resolution methods, the reader is referred to Wang et al. (2021)<sup>40</sup>.

In the field of materials science, super-resolution of microscopy image data is of great interest, since imaging techniques are often time-consuming and there is a trade-off between the imaged area/volume of the material and the resolution achieved. More precisely, imaging performed with a small pixel/voxel size (i.e., high-resolution) can capture more details of a material's microstructure which, however, leads to a smaller area/volume of the material being imaged. Therefore, due to local material heterogeneities, single images obtained by high-resolution imaging may not be statistically representative<sup>41</sup>. On the other hand, low-resolution imaging can capture larger areas/volumes, yet, fine details of the microstructure may not be visible.

This dilemma of balancing field of view and resolution is particularly prevalent in the field of Li-ion batteries where electrodes have multi-scale architectural heterogeneities, each requiring analysis of representative volumes for accurate characterization<sup>42</sup>. For example, electrode particles have distributions of shapes and sizes, necessitating a field of view large enough to capture a volume of particles that provides representative characterization of their morphology<sup>43</sup>. For extremely small features that greatly vary across relatively large volumes, both a

<sup>1</sup>Institute of Stochastics, Ulm University, Helmholtzstraße 18, 89069 Ulm, Germany. <sup>2</sup>National Renewable Energy Laboratory, 15013 Denver W Parkway, Golden, CO 80401, USA.

<sup>3</sup>Chemical Sciences and Engineering Division, Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, IL 60439, USA. ✉email: [orkun.furat@uni-ulm.de](mailto:orkun.furat@uni-ulm.de); [donal.finegan@nrel.gov](mailto:donal.finegan@nrel.gov)

large field of view and high resolution is needed. Cracks within electrode particles are an example of such extremely small (<500 nm) features<sup>44</sup>. It is expected that cracks will vary for different particle architectures, therefore likely requiring similar representative volumes for particle characterization but requiring higher resolution. Therefore, crack characterization within electrode particles requires both large field of view and high resolution, which in the case of SEM is both time-consuming and expensive.

This issue can be remedied, by super-resolving experimentally measured low-resolution images, which would lead to detailed image data of larger areas/volumes. For example, in Hagita et al. (2018)<sup>45</sup> and de Haan et al. (2019)<sup>46</sup>, GANs were used for super-resolving SEM image data of silica and gold nanoparticles, respectively, whereas in Jung et al. (2021)<sup>47</sup>, a CNN was used for super-resolving EBSD image data. Specifically, the approaches in Hagita et al. (2018)<sup>45</sup> and Jung et al. (2021)<sup>47</sup> used downsampled high-resolution images in order to obtain low-resolution versions of the former as training data. In this context, however, downsampling of high-resolution images does not always model experimentally measured low-resolution images accurately<sup>48</sup>. Therefore, networks which have been trained on synthetic low-resolution images may not perform as well on experimentally measured low-resolution images<sup>49</sup>.

In the present paper, we deploy a slightly modified version of the GAN described in Ledig et al. (2017)<sup>33</sup>, the so-called super-resolution GAN (SRGAN), for performing super-resolution on SEM image data of differently aged  $\text{LiNi}_{1-x-y}\text{Mn}_y\text{Co}_x\text{O}_2$  (NMC) particles within cathodes for Li-ion batteries. The aging of such particles leads to the formation of cracks, which are fine features being heterogeneously present throughout particles within the cathode material. Thus, large areas/volumes have to be imaged in order to obtain representative data. This makes low-resolution image data of aged cathode particles an ideal case for studying the viability of super-resolution architecture SRGAN. For training the SRGAN, we use pairs of experimentally measured low-resolution SEM images and corresponding (experimentally measured) high-resolution images where the resolution of the latter is  $a = 2.5$  times higher than the former. Note that the network architecture described in the present paper can easily be adapted for performing super-resolution for different values of  $a$ . We compare the super-resolution results obtained by the SRGAN with those obtained by trained versions of the networks described in de Haan et al. (2019)<sup>46</sup> and Jung et al. (2021)<sup>47</sup> which have been used for super-resolving microscopy image data. This quantitative comparison indicates that, in the context of super-resolving SEM image data of NMC particles, the trained SRGAN outperforms the networks studied so far in literature.

Additionally, we train another GAN using the approach described in Yuan et al. (2018)<sup>35</sup>. Therefore, during training we consider a scenario in which experimentally measured low- and high-resolution images are available—however, in which we do not have matching (i.e., registered) pairs of such images. This network is trained with downsampled versions of the experimentally measured high-resolution images. Nevertheless, during training the network receives experimentally measured low-resolution images as well, such that it can learn features which are specific to experimental low-resolution images and not present in downsampled images. In direct comparison, the GAN trained with this approach outperforms networks which have solely been trained with downsampled high-resolution images. This indicates that GANs can be used to reliably enhance the resolution of experimentally measured image data in order to obtain more detailed, yet statistically representative microscopy image data—even when no registered image pairs with low- and high-resolutions are available. For example, this approach could be of interest for super-resolving low-resolution microscopy

images within existing datasets which have been measured without corresponding high-resolution images.

Since the SEM image data considered in the present paper depicts differently aged/cracked cathode particles, this dataset will serve as the basis for investigating the influence of aging parameters on the crack formation within cathode particles in future studies. Therefore, in the present paper, we additionally study to what extent super-resolution supports the analysis of crack formation. More precisely, we segment the cracks within super-resolved image data which we compare to cracks determined from high-resolution images. We observe a significant improvement for crack segmentation results when using super-resolved images in comparison to upsampled low-resolution images, see the discussion section for more details. This indicates that super-resolving SEM image data of cathode materials can significantly support the analysis of battery aging processes. Moreover, super-resolution using machine learning methods is not limited to SEM image data of cathode materials. The networks discussed in the present paper could easily be deployed onto image data obtained by different measurement techniques like, for example, atomic force microscopy<sup>50</sup>.

Thus, this technique is expected to have a plethora of applications in materials science and particularly Li-ion electrode characterization where understanding the distributions of small components and features such as conductive carbon, cracks, and unwanted deposits are critical to understanding the performance and degradation of cells.

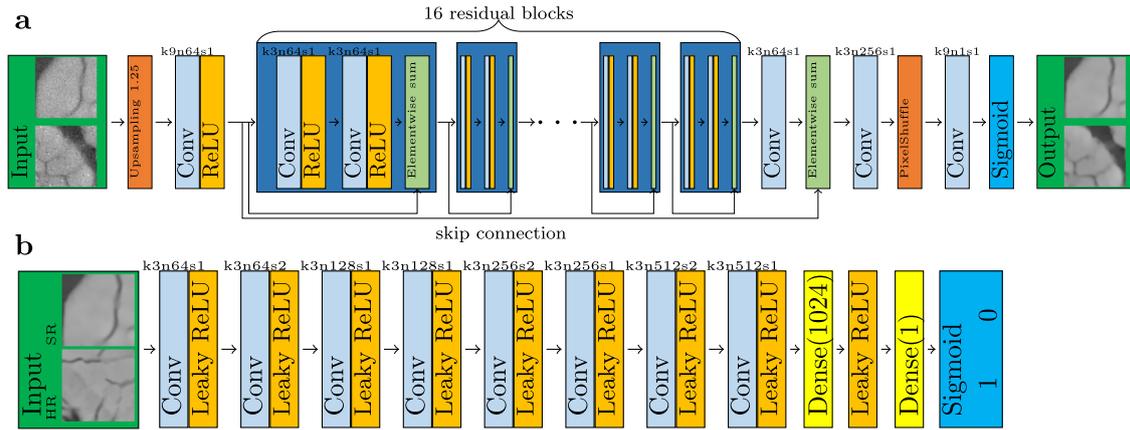
## RESULTS

### Architecture of the generative adversarial network

In this section, we describe the network architecture which we deploy for increasing the resolution of SEM image data of cathode materials for Li-ion batteries, see the sample details given below. We use the so-called SRGAN architecture described in Ledig et al. (2017)<sup>33</sup> which is based on a GAN. More precisely, the considered GAN consists of two neural networks, i.e., a generator  $G_{\theta_G} = G$  and a discriminator  $D_{\theta_D} = D$ , where  $\theta_G \in \mathbb{R}^{n_G}$  and  $\theta_D \in \mathbb{R}^{n_D}$  for some  $n_G, n_D > 0$  denote the weights of the generator and the discriminator, respectively. The former receives a (single-channel) low-resolution SEM image  $I_{LR}: \{1, \dots, h\} \times \{1, \dots, w\} \times \{1\} \rightarrow [0, 1]$  with height  $h > 1$  and width  $w > 1$  as input for which it computes a high-resolution version  $\hat{I}_{HR} = G(I_{LR}): \{1, \dots, ah\} \times \{1, \dots, aw\} \times \{1\} \rightarrow [0, 1]$  of  $I_{LR}$  as output, where  $a > 1$  is a scaling factor such that  $ah$  and  $aw$  are integers. The high-resolution version  $\hat{I}_{HR}$  of  $I_{LR}$  should resemble the corresponding (experimentally measured) high-resolution SEM image  $I_{HR}: \{1, \dots, ah\} \times \{1, \dots, aw\} \times \{1\} \rightarrow [0, 1]$ . Note that the high-resolution image data considered in the present paper has been denoised, meaning that the generator  $G$  performs both super-resolution and denoising, see the methods section for more details on the image data.

On the other hand, the discriminator  $D$  is supposed to distinguish between experimentally measured high-resolution images and those computed by the generator  $G$ , where the discriminator's output has to be as high as possible for the high-resolution image  $I_{HR}$ , and as low as possible for  $\hat{I}_{HR}$  computed by the generator, i.e.,  $D(I_{HR}) = 1$  and  $D(\hat{I}_{HR}) = 0$ . Moreover, both networks  $G$  and  $D$  are in contest with each other, i.e., during training the generator  $G$  tries to produce high-resolution versions of low-resolution SEM images which are evaluated by the discriminator  $D$  as experimentally measured ones.

Now, we describe the architectures of the considered neural networks  $G$  and  $D$  in detail. To accommodate our data and hardware situation we slightly modify the original architecture of SRGAN (cf. Fig. 4 in Ledig et al., 2017<sup>33</sup>). Since the pixel size of the low-resolution SEM data considered in the present paper is 2.5 times larger than the pixel size of the high-resolution data, we



**Fig. 1 Network architectures.** Modified versions of the architectures described in Ledig et al. (2017)<sup>33</sup> for the generator  $G$  (a) and discriminator  $D$  (b) considered in the present paper. The labels above convolutional layers (Conv) indicate the kernel size ( $k$ ), the number of feature maps ( $n$ ) and the stride ( $s$ ). For example, the label  $k9n64s1$  indicates a convolutional layer with kernel size  $k=9$ , number of feature maps  $n=64$ , and stride  $s=1$ .

chose architectures which accommodate this, i.e., we set  $\alpha=2.5$ . As generator  $G$  for our GAN architecture we use a SRResNet<sup>33</sup> with 16 residual blocks. In order to increase the spatial resolution of feature maps by a factor of  $\alpha=2.5$  into each spatial dimension the input is upsampled by a factor of 1.25 and a single PixelShuffle layer<sup>51</sup> prior to the output layer is used (see Fig. 1a). Note that we made further modifications to the original SRGAN architecture by replacing PReLU layers<sup>52</sup> with ReLU layers<sup>53</sup> and by omitting BatchNormalization layers<sup>54</sup>. In this manner the generator of SRGAN coincides with the architecture utilized in Jung et al. (2021)<sup>47</sup> to which we will compare the performance of SRGAN below. Furthermore, by using ReLU layers instead of PReLU we additionally decrease the number of network weights, thus, increasing computational feasibility. The BatchNormalization layers were removed since they can decrease the network's accuracy for performing super-resolution tasks<sup>55,56</sup>. Additionally, the removal of BatchNormalization accommodates the small batch size utilized during the training procedure described below<sup>57</sup>. Finally, we use a sigmoid activation function for the convolutional output layer and reduce the number of its feature maps from three to one such that the network's outputs are single channel images with values in the interval  $[0, 1]$ , i.e., the network's outputs can be interpreted as grayscale images. The discriminator  $D$  is a slightly modified version (i.e., BatchNormalization layers were omitted) of the discriminator used in Ledig et al. (2017)<sup>33</sup> (see Fig. 1b).

### Optimization of network parameters

In order to train a GAN to perform super-resolution we formulate an optimization problem which consists of two components. The first component measures how much an image  $\hat{I}_{HR}$  computed by the generator  $G$  deviates from the actual high-resolution image  $I_{HR}$ . For this purpose, in statistical learning, a common loss function is the mean squared error (MSE) given by

$$\text{MSE}(I_1, I_2) = \frac{1}{cwh} \sum_{x=1}^w \sum_{y=1}^h \sum_{c=1}^c (I_1(x, y, c) - I_2(x, y, c))^2, \quad (1)$$

where  $I_1, I_2 : \{1, \dots, h\} \times \{1, \dots, w\} \times \{1, \dots, c\} \rightarrow \mathbb{R}$  are images with  $c$  channels, height  $h$  and width  $w$ . However, Ledig et al. (2017)<sup>33</sup> showed that for super-resolution tasks better results can be achieved with the so-called perceptual loss  $\text{PL}_{i,j,v}$  which is given by

$$\text{PL}_{i,j,v}(I_1, I_2) = \text{MSE}(\phi_{i,j,v}(I_1), \phi_{i,j,v}(I_2)), \quad (2)$$

where  $\phi_{i,j,v}(I_k)$  denotes the output of the  $i$ th convolution layer before the  $j$ th maxpooling layer of the pre-trained Visual

Geometry Group (VGG) network<sup>58</sup> with depth  $v \in \{16, 19\}$  for the input image  $I_k$  with  $k=1, 2$ . Then, one objective during the training of the generator  $G$  is the minimization of  $\text{PL}_{i,j,v}(G(I_{LR}), I_{HR})$  for some specification of  $i, j$  and  $v$ . The other objective is to "trick" the discriminator  $D$  to believe that the generator's output  $G(I_{LR})$  is an experimentally measured high-resolution image, i.e., the minimization of  $\log(1 - D(G(I_{LR}))$ . On the other hand, the discriminator  $D$  is supposed to distinguish between  $G(I_{LR})$  and  $I_{HR}$ , i.e., during training the objective is also to maximize  $\log D(I_{HR}) + \log(1 - D(G(I_{LR})))$ . Then, putting  $i=j=2$  and  $v=19$ , the minimax problem for optimizing the GAN is given by

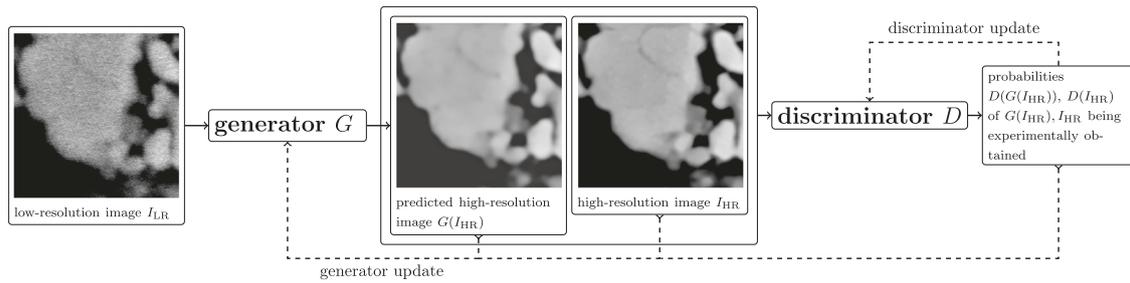
$$\min_{\theta_G \in \Theta_G} \max_{\theta_D \in \Theta_D} \mathbb{E}[\text{PL}_{2,2,19}(G_{\theta_G}(J_{LR}), J_{HR})] + \gamma (\mathbb{E}[\log D_{\theta_D}(J_{HR})] + \mathbb{E}[\log(1 - D_{\theta_D}(G_{\theta_G}(J_{LR})))]), \quad (3)$$

where  $\gamma > 0$  denotes the adversarial weight, and  $\Theta_G \subset \mathbb{R}^{n_G}$ ,  $\Theta_D \subset \mathbb{R}^{n_D}$  are the sets of admissible weights for the generator  $G$  and discriminator  $D$ , respectively<sup>33</sup>. Furthermore,  $J_{LR}$  denotes the random low-resolution image obtained by taking a  $96 \times 96$ -sized cutout from the training data at random, and  $J_{HR}$  is the corresponding random high-resolution image. Note that the optimization problem given in Formula (3) requires pairs of low-resolution and high-resolution images  $I_{LR}$  and  $I_{HR}$  (see Fig. 2). If no experimentally measured pairs of such low-resolution and high-resolution images are available, training can still be performed by synthetically downsampling the high-resolution image  $I_{HR}$ . For example, using bilinear or bicubic interpolation we can obtain downsampled versions  $I_{LR}$  of  $I_{HR}$  for training purposes<sup>59</sup>. Then, the corresponding optimization problem is given by

$$\min_{\theta_G \in \Theta_G} \max_{\theta_D \in \Theta_D} \mathbb{E}[\text{PL}_{2,2,19}(G_{\theta_G}(\tilde{J}_{LR}), J_{HR})] + \gamma (\mathbb{E}[\log D_{\theta_D}(J_{HR})] + \mathbb{E}[\log(1 - D_{\theta_D}(G_{\theta_G}(\tilde{J}_{LR})))]), \quad (4)$$

where  $J_{HR}$  denotes the random high-resolution image obtained by taking a  $240 \times 240$ -sized cutout from the training data at random, and  $\tilde{J}_{LR}$  denotes the downsampled version of  $J_{HR}$  with size  $96 \times 96$ .

However, note that a network which is trained according to the rule described in Formula (4) with artificially generated low-resolution images, may not perform well on experimentally measured low-resolution images, since artificially downsampled images may not exhibit the same features (e.g. the same type of noise) as experimentally measured images with the same resolution<sup>48,49</sup>. For such unsupervised data scenarios, so-called CycleGAN<sup>36</sup> architectures can be considered for performing super-resolution<sup>35,48,60</sup>.



**Fig. 2 Scheme for training the SRGAN.** Training procedure according to the optimization problem given in Formula (3) when matching pairs of experimentally measured low-resolution images  $I_{LR}$  with corresponding high-resolution versions  $I_{HR}$  are available.

### Simulation-based training procedures

We now describe the training of various neural networks for performing super-resolution tasks. In particular, we train two different versions of the network architecture described in the previous sections, which is based on the GAN considered in Ledig et al. (2017)<sup>33</sup>. Furthermore, we train the GAN architecture described in de Haan et al. (2018)<sup>46</sup> and two variants of the architecture presented in Jung et al. (2021)<sup>47</sup>. Then, in the next section, we quantitatively compare the super-resolution results obtained by the architectures considered in the present paper with those described in de Haan et al. (2018)<sup>46</sup> and Jung et al. (2021)<sup>47</sup>.

First, we describe the training of the SRGAN architecture described above (see Fig. 1) by solving the optimization problem given in Formula (3) where we set the adversarial weight  $\gamma$  equal to 2.0. Before training, we split the available 33 pairs of experimentally measured low-resolution and corresponding high-resolution images (see the “Methods” section below for details) into training, validation and test sets, each consisting of 24, 5 and 4 pairs of images, respectively. Then, the network weights are initialized using the Glorot scheme<sup>61</sup>, followed by solving the optimization problem given in Formula (3) using the stochastic gradient descent method Adam<sup>62</sup> with a learning rate of  $10^{-4}$ . More precisely, in each training step, 32 (rotated) cutouts of size  $96 \times 96$  are taken at random from the low-resolution images in the training data set accompanied with the corresponding high-resolution cutouts with size  $240 \times 240$ . These images are used to estimate the expected values within the objective function of the minimax problem given in Formula (3) and the corresponding gradient with respect to the network weights  $\theta_G$  and  $\theta_D$ . Due to memory limitations, in each training step the gradient is computed by determining 32 gradients for each individual cutout followed by averaging. Note that, using the averaged gradient, the weights of the generator  $G_{\theta_G}$  and the discriminator  $D_{\theta_D}$  are updated alternately in each step of the optimization procedure.

To avoid overfitting, every 20 steps the performance of the generator is evaluated with respect to the  $PL_{2,2,19}$  loss on 92 pairs of cutouts taken at random from the validation set. Note that each validation step is performed on the same set of 92 pairs of cutout images. If the performance on the validation data set does not improve within 10 consecutive performance checks, the training procedure is stopped and the network’s weights are reset to the best performing version, which we denote by SRGAN. The networks were implemented using the Python package TensorFlow<sup>63</sup> and trained in <10 h on a GPU (system specifications—RAM: 32 GB; CPU: AMD Ryzen 5 3600 with six 3.6 GHz cores; GPU: NVIDIA GeForce RTX 3060).

Analogously, we train the architectures described in de Haan et al. (2018)<sup>46</sup> and Jung et al. (2021)<sup>47</sup> with their respective loss functions (cf. Eqs. (2)–(4) in de Haan et al. (2018)<sup>46</sup> and Table 2 in Jung et al. (2021)<sup>47</sup>), where we denote the corresponding trained

networks by U-NetGAN and SRResNet1, respectively. Note that the latter one is not a GAN architecture, i.e., the update step for the discriminator is skipped during training. Furthermore, the architecture described in Jung et al. (2021)<sup>47</sup> had to be slightly modified to accommodate our super-resolution task of increasing the spatial resolution by a factor of 2.5 in each dimension. More precisely, we upsample the input by a factor of 1.25 and use just a single PixelShuffle layer for upsampling. Thus, the SRResNet1 architecture coincides with the architecture of the generator of SRGAN.

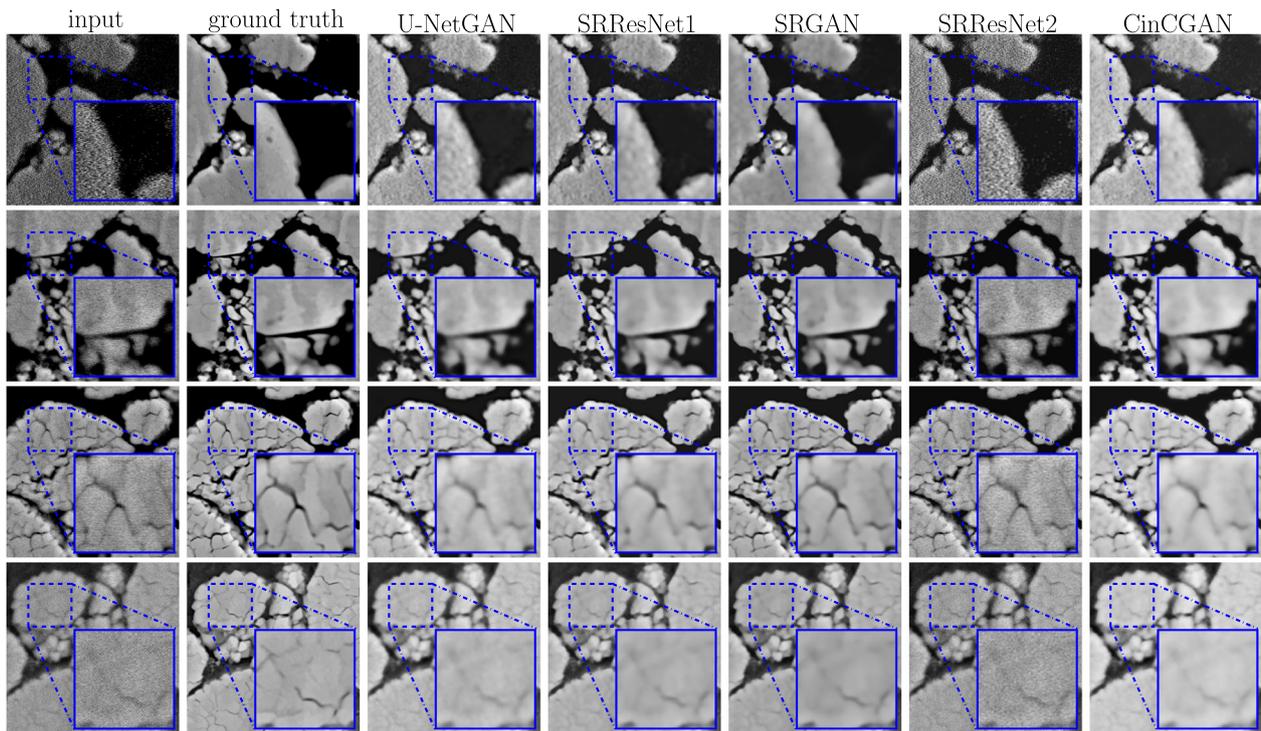
In addition to the training of the three architectures described above—for which we utilize training data comprised of matching pairs of experimentally measured low-resolution and high-resolution image data—we train two further networks for which we do not utilize such matching pairs. Here, we train another variant of the SRResNet architecture. Therefore, similarly to the training procedure described in Jung et al. (2021)<sup>47</sup> we create batches by taking  $240 \times 240$  sized cutouts at random from the high-resolution training data, from which we compute synthetic low-resolution images by downsampling. We denote the corresponding trained network by SRResNet2. Recall that training on downsampled high-resolution images can lead to a poor performance when applying the trained network to actual low-resolution data<sup>48,49</sup>. Thus, in addition, we utilize GANs, namely the CinCGAN architecture, to overcome this issue, cf. Fig. 2 in Yuan et al. (2018)<sup>35</sup>. This architecture consists of two GANs, where the task of the first GAN is to denoise low-resolution images such that they resemble downsampled versions of high-resolution images. The task of the second GAN is to super-resolve the output of the first GAN. To accommodate our data situation, we slightly modify the original CinCGAN architecture by replacing the network denoted by SR in Yuan et al. (2018)<sup>35</sup> with the architecture visualized in Fig. 1a, such that our CinCGAN architecture increases the spatial resolution of low-resolution images by the factor  $\alpha = 2.5$ . An overview of the network architectures, the optimization problems and the considered training data for the five networks is given in Table 1. Some super-resolution results obtained by the trained networks are depicted in Fig. 3.

### Quantitative analysis of super-resolution results

To begin with, a visual comparison of super-resolution results achieved by the five trained networks, described in the previous section, is given in Fig. 3. Then, we quantitatively analyze the super-resolution results (see Table 1). For that purpose, we leverage the test data which consists of four pairs of low-resolution and corresponding high-resolution images which have not been used for network training. Therefore, we denote these pairs of images by  $(I_{LR}^{(1)}, I_{HR}^{(1)}), \dots, (I_{LR}^{(4)}, I_{HR}^{(4)})$ . For each trained network, we predict high-resolution versions  $\hat{I}_{HR}^{(1)}, \dots, \hat{I}_{HR}^{(4)}$  of  $I_{LR}^{(1)}, \dots, I_{LR}^{(4)}$ . Then, the discrepancies between the predictions  $\hat{I}_{HR}^{(1)}, \dots, \hat{I}_{HR}^{(4)}$  and the high-resolution images  $I_{HR}^{(1)}, \dots, I_{HR}^{(4)}$  are

**Table 1.** Overview of the training specifications for the neural networks considered in the present paper.

Name	U-NetGAN	SRResNet1	SRGAN	SRResNet2	CinCGAN
Generator architecture	Modified U-Net	SRResNet	SRResNet	SRResNet	SRResNet
Discriminator	Yes	No	Yes	No	Yes
Optimization problem	cf. Eqs. (2)–(4) in de Haan et al. (2019) <sup>46</sup>	Minimization of the mean absolute error	Optimization problem given in Eq. (3)	Minimization of the mean absolute error	cf. Eqs. (5) and (10) in Yuan et al. (2018) <sup>35</sup>
Training data	Matching pairs of low-resolution and high-resolution images	Matching pairs of low-resolution and high-resolution images	Matching pairs of low-resolution and high-resolution images	High-resolution images with synthetically downsampled low-resolution images	Non-matching low-resolution and high-resolution images and synthetically downsampled low-resolution images



**Fig. 3 Visual comparison of super-resolution results.** For four different cutouts (rows 1–4), super-resolution results are shown which have been obtained by five trained networks. Experimentally measured (noisy) low-resolution images which serve as input are depicted in the first column. The corresponding (denoised) high-resolution images are shown in the second column. They serve as ground truth. The super-resolution results obtained by U-NetGAN, SRResNet1, SRGAN, SRResNet2, and CinCGAN are depicted in columns 3–7, respectively. Magnifications with a zoom factor 2 of the dashed blue squares are visualized in the blue solid-lined squares.

computed using various loss functions. In particular, we consider the average of the mean squared error (MSE) given by

$$\overline{\text{MSE}} = \frac{1}{4} \sum_{k=1}^4 \text{MSE} \left( \hat{I}_{\text{HR}}^{(k)}, I_{\text{HR}}^{(k)} \right). \quad (5)$$

Moreover, we evaluate the predictions by computing two different types of VGG losses, i.e., we compute

$$\overline{\text{PL}}_{2,2,v} = \frac{1}{4} \sum_{k=1}^4 \text{PL}_{2,2,v} \left( \hat{I}_{\text{HR}}^{(k)}, I_{\text{HR}}^{(k)} \right), \quad (6)$$

for  $v = 16, 19$ . The resulting values of  $\overline{\text{MSE}}$ ,  $\overline{\text{PL}}_{2,2,16}$  and  $\overline{\text{PL}}_{2,2,19}$ , which have been obtained for the trained networks, are listed in Table 2. In addition, to evaluate the discrepancy between predicted and experimentally measured high-resolution images, we consider the mean structural similarity index (MSSIM) as defined in Wang et al. (2004)<sup>64</sup>. The values, which are obtained for

the corresponding averages

$$\overline{\text{MSSIM}} = \frac{1}{4} \sum_{k=1}^4 \text{MSSIM} \left( \hat{I}_{\text{HR}}^{(k)}, I_{\text{HR}}^{(k)} \right), \quad (7)$$

are listed in Table 2. Note that in contrast to the values of  $\overline{\text{MSE}}$ ,  $\overline{\text{PL}}_{2,2,16}$  and  $\overline{\text{PL}}_{2,2,19}$ , larger values of  $\overline{\text{MSSIM}}$  indicate better results. Thus, altogether, SRGAN leads to better predictions than the remaining four networks, see also the “Discussion” section.

### Super-resolution for improved crack detection

In the previous section, we investigated the performance of super-resolution results obtained by the trained networks by direct comparison to the grayscale high-resolution images. Recall, that the SEM image data considered in the present paper depicts differently aged cathode particles where the aging leads to cracks within the particles. Thus, for investigating the influence of aging

**Table 2.** Quantitative comparison of super-resolution results, where the values of MSE,  $\overline{PL}_{2,2,16}$ ,  $\overline{PL}_{2,2,19}$ , and MSSIM obtained for SRGAN (marked in boldface) indicate that SRGAN leads to better predictions than the remaining four networks.

Name	U-NetGAN	SRResNet1	SRGAN	SRResNet2	CinCGAN
MSE	5.70e−03	5.91e−03	<b>5.67e−03</b>	6.03e−03	5.70e−03
$\overline{PL}_{2,2,19}$	2.33e+01	2.33e+01	<b>2.21e+01</b>	4.79e+01	2.32e+01
$\overline{PL}_{2,2,16}$	6.19e+01	6.25e+01	<b>5.92e+01</b>	1.24e+02	6.06e+01
MSSIM	8.85e−01	8.80e−01	<b>8.88e−01</b>	7.73e−01	8.86e−01

on the crack formation, the cracks have to be identified reliably from SEM image data. Therefore, in this section we investigate how super-resolution of low-resolution SEM image data improves subsequent procedures for the crack segmentation.

For that purpose, let  $l_{HR} : \{1, \dots, h\} \times \{1, \dots, w\} \rightarrow \mathbb{R}$  be a high-resolution image within the test data set which consists of four pairs of low- and high-resolution images (see Fig. 4a). Using a modified version of the method described in Westhoff et al. (2018)<sup>65</sup> we compute a segmentation map  $S_{HR} : \{1, \dots, h\} \times \{1, \dots, w\} \rightarrow \{0, 1, 2\}$  which is given by

$$S_{HR}(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \text{ is associated with the background,} \\ 1, & \text{if } \mathbf{x} \text{ is associated with a crack,} \\ 2, & \text{if } \mathbf{x} \text{ is associated with a particle,} \end{cases} \quad (8)$$

for each  $\mathbf{x} \in \{1, \dots, h\} \times \{1, \dots, w\}$ . Figure 4b visualizes the segmentation map  $S_{HR}$  of the high-resolution image  $l_{HR}$  depicted in Fig. 4a. For more details on the segmentation procedure, see the Supplementary Note 1 and Supplementary Fig. 1.

For technical reasons, we extend the domain of  $S_{HR}$  to the (continuous) rectangle  $[1, h] \times [1, w] \subset \mathbb{R}^2$  using nearest-neighbor interpolation by

$$S_{HR}(\mathbf{x}) = S_{HR}(\lceil x_1 \rceil, \lceil x_2 \rceil) \quad (9)$$

for each  $\mathbf{x} = (x_1, x_2) \in [1, h] \times [1, w]$ , where  $\lceil x_i \rceil$  denotes the closest integer to  $x_i$  with  $\lceil x_i \rceil = x_i - 0.5$  if  $2x_i$  is an odd integer<sup>25</sup>. Then, we can determine the set of points associated with cracks by

$$C_{HR} = \{\mathbf{x} \in [1, h] \times [1, w] : S_{HR}(\mathbf{x}) = 1\}. \quad (10)$$

Analogously, for a super-resolution version  $\hat{l}_{HR}$  of  $l_{HR}$  we compute the segmentation map  $\hat{S}_{HR}$  and the set  $\hat{C}_{HR}$  of points associated with cracks determined from  $\hat{l}_{HR}$  (see Fig. 4d). In order to investigate to what extent super-resolution improves crack segmentation results, we determine the set of points associated with cracks from the corresponding low-resolution image  $l_{LR}$  without performing super-resolution. More precisely, we upsample  $l_{LR}$  by a factor of 2.5 using bilinear interpolation<sup>59</sup> followed by denoising (see the “Methods” section for more details on the denoising procedure). Then, the upsampled and denoised image is segmented such that we obtain the corresponding segmentation map  $\hat{S}_{HR}$  and the set  $\hat{C}_{HR}$  of points which are associated with cracks (see Fig. 4c).

In order to quantify the similarity between cracks  $\hat{C}_{HR}$  determined from super-resolution/upsampled images and the ground truth  $C_{HR}$  we consider the Jaccard index which is given by

$$J(\hat{C}_{HR}, C_{HR}) = \frac{v_2(\hat{C}_{HR} \cap C_{HR})}{v_2(\hat{C}_{HR} \cup C_{HR})}, \quad (11)$$

where  $v_2(C)$  denotes the area of a set  $C \subset [1, h] \times [1, w]$ <sup>66</sup>. Note that the values of the Jaccard index are normalized, i.e., the value  $J(\hat{C}_{HR}, C_{HR})$  belongs to the interval  $[0, 1]$  and large values indicate a good match between the sets  $\hat{C}_{HR}$  and  $C_{HR}$ . The values of the Jaccard index for cracks segmented from upsampled low-resolution images (as reference) and from

super-resolution images computed by the trained networks U-NetGAN, SRResNet1, SRGAN, SRResNet2, and CinCGAN are listed in Table 3.

Additionally, we investigate how well quantities for characterizing crack formation in particles can be estimated using super-resolved image data. More precisely, we compute the specific crack density  $\rho$  from the segmented high-resolution image data which is given by

$$\rho = \frac{v_2(C_{HR})}{v_2(C_{HR} \cup P_{HR})}, \quad (12)$$

where  $P_{HR}$  denotes the set of points associated with the solid phase of particles, i.e.,  $P_{HR} = \{\mathbf{x} \in [1, h] \times [1, w] : S_{HR}(\mathbf{x}) = 2\}$ . From the high-resolution image data we determine the specific crack density to be  $\rho = 0.123$ . Analogously, we estimate the specific crack density  $\hat{\rho}$  from upsampled/super-resolved low-resolution data, see Table 3 for the relative errors with respect to  $\rho$ .

Additionally, we compute descriptors which characterize the cracks in order to quantify the improvement of segmentation results when utilizing super-resolved image data. First, we determine connected components in  $C_{HR}$ , i.e., we determine  $m \geq 1$  connected components  $C_1, \dots, C_m \subset C_{HR}$  with  $C_{HR} = \bigcup_{i=1}^m C_i$ . Then, for each component  $C_i$  we compute the area-equivalent diameter  $d_i$  by

$$d_i = 2 \sqrt{\frac{v_2(C_i)}{\pi}}, \quad i = 1, \dots, m. \quad (13)$$

Then, we determine the probability density  $f : [0, \infty) \rightarrow [0, \infty)$  of the area-equivalent diameter of cracks computed from the high-resolution image data, by fitting a log-normal distribution<sup>67</sup> with density  $f_{\sigma, \mu}$  to the area-equivalent diameters  $d_1, \dots, d_m$  using maximum-likelihood estimation<sup>68</sup>, see the Supplementary Note 2 for further details. Note that the probability density  $f_{\sigma, \mu}$  is given by

$$f_{\sigma, \mu}(x) = \frac{1}{\sigma \sqrt{2\pi}x} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right), \quad (14)$$

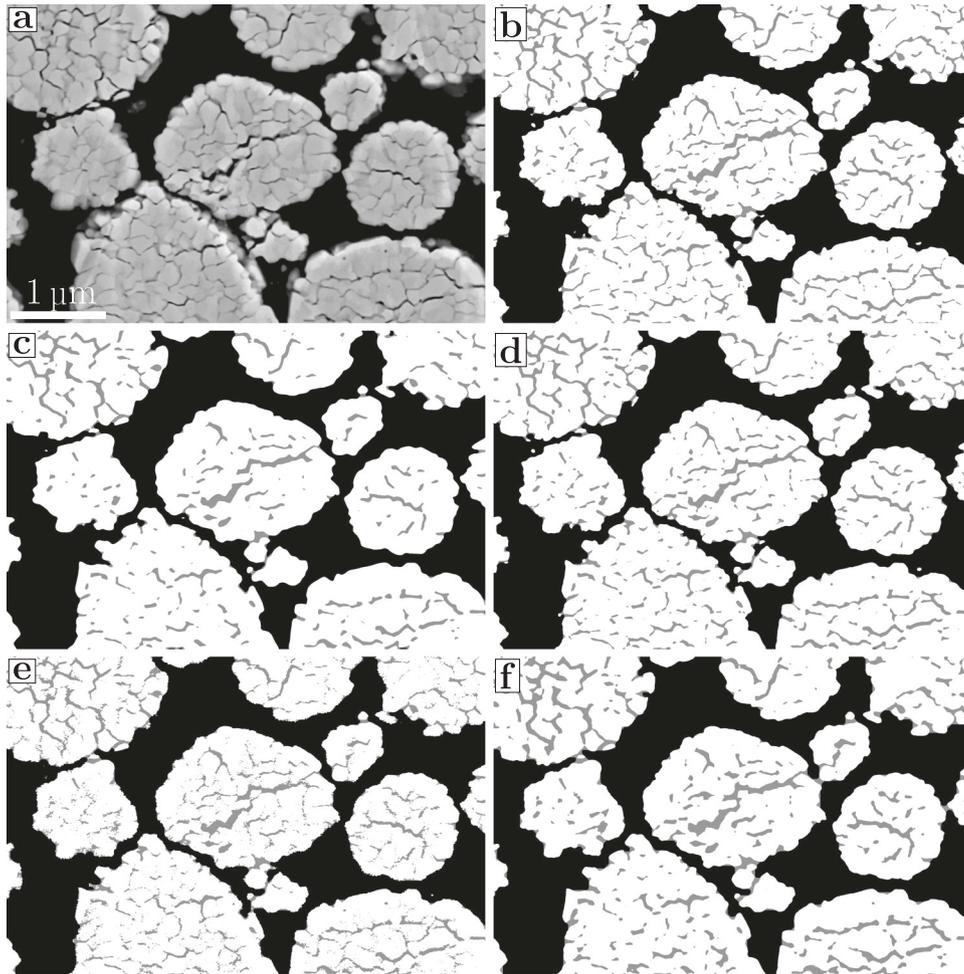
where  $\sigma, \mu > 0$  are model parameters. The corresponding log-normal fit for the probability density of crack diameters computed from high-resolution image data is visualized in Fig. 5a (blue line). Analogously, the corresponding probability densities  $\hat{f}$  are determined for cracks computed from upsampled/super-resolved low-resolution images, see Fig. 5a (For a visualization of histograms and corresponding log-normal fits, see Supplementary Fig. 2). Note that for the computation of the probability densities  $f$  and  $\hat{f}$  we disregarded area-equivalent diameters  $< 50$  nm as the corresponding connected components are indistinguishable from noise. Then, the discrepancy between the probability density  $\hat{f}$  and the corresponding ground truth  $f$  can be quantified by

$$\|f - \hat{f}\| = \int_0^\infty |f(x) - \hat{f}(x)| dx. \quad (15)$$

The values of  $\|f - \hat{f}\|$  for probability densities of crack sizes determined from upsampled low-resolution images (as reference) and from super-resolution images computed by the trained networks U-NetGAN, SRResNet1, SRGAN, SRResNet2, and CinCGAN are listed in Table 3. Altogether, SRGAN performs best with respect to crack segmentation, see also the discussion provided in the next section.

## DISCUSSION

The super-resolution results achieved by the five networks considered in the present paper are visualized in Fig. 3. They indicate that the networks perform quite well, especially when evaluated on low-resolution images with low amount of noise (see Fig. 3) (second, third and fourth rows). However, the network



**Fig. 4 Crack segmentation.** High-resolution image  $I_{HR}$  (a) and the corresponding segmentation map  $S_{HR}$  computed from  $I_{HR}$  (b) where black color indicates the background, gray color the cracks and white color the particles. The corresponding segmentation map  $\hat{S}_{HR}$  computed from the upsampled low-resolution image (c) and from the images super-resolved by SRGAN (d), SRResNet2 (e), and CinCGAN (f). All figures use the same length scale.

SRResNet2 seems to perform worse than the other networks on noisy data, see Fig. 3 (first row). The reason for this might be that, as in Jung et al. (2021)<sup>47</sup>, it has been trained with pairs of high-resolution images and corresponding downsampled versions, for which the latter ones do not necessarily exhibit the same kind of noise as experimentally measured low-resolution images. This is also reflected quantitatively in Table 2 which indicates that the other networks mostly outperform SRResNet2. This issue is resolved by training networks with both experimentally measured low- and high-resolution images.

For example, in the unsupervised scenario, i.e., when no matching pairs of low-resolution and high-resolution images are available, CinCGAN performs significantly better than SRResNet2 (see Table 2), where we can see that it even has a similar performance as U-NetGAN which has been trained with matching pairs of experimentally measured low- and high-resolution images. This indicates that GANs are a viable option for performing super-resolution on microscopy image data when no matching pairs of low-resolution and high-resolution image are available/obtainable for training purposes.

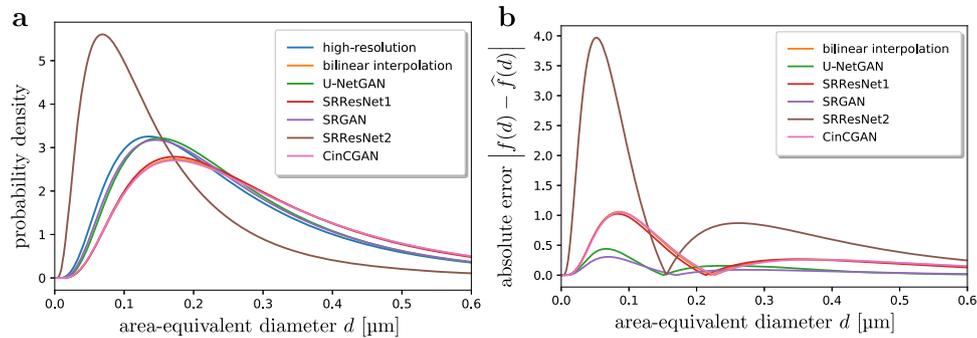
Among the networks which have been trained with matching pairs of experimentally measured low-resolution and high-resolution images (i.e., U-NetGAN, SRResNet1, and SRGAN) the network SRGAN exhibits the best performance (see Table 2). It even outperforms the other GAN architecture U-NetGAN. Apart

from differences in the network architecture, this can also be attributed to differences in the optimization problem which has been solved during the training procedure of U-NetGAN. More precisely, the generator of U-NetGAN was trained to minimize the  $L_1$  loss (i.e., mean absolute error)<sup>69</sup> as well as the anisotropic total variation loss<sup>70</sup>, cf. Eqs. (2)–(4) in de Haan et al. (2019)<sup>46</sup>; whereas the generator of SRGAN was trained to minimize the perceptual loss  $PL_{2,2,19}$ . Nevertheless, SRGAN also performs best with respect to the other considered performance measures (i.e., MSE,  $PL_{2,2,16}$  and MSSIM) which have not been optimized during training. In summary, GANs trained to minimize the perceptual loss seem to be a viable option for performing super-resolution of microscopy image data.

Note that the quantitative results of Table 2 discussed so far, do not reflect how well cracks within the super-resolved image data can be quantitatively analyzed—which is, however, an important aspect for investigating structural degradation in cathode materials. More precisely, the discrepancy measures listed in Table 2 are computed by averaging pixel-wise discrepancies, see, for example, Eqs. (1) and (2). However, pixels associated with cracks within the image data make up only a small fraction of all pixels, such that inaccuracies of pixels associated with cracks in super-resolved images only marginally affect these discrepancy measures. For example, the quantitative comparison regarding crack segmentation results listed in Table 3 indicates that the error

**Table 3.** Quantitative comparison of crack segmentation results., where the values of  $J(\hat{C}_{HR}, C_{HR})$ ,  $|\rho - \hat{\rho}|/\rho$  and  $\|f - \hat{f}\|$  obtained for SRGAN (marked in boldface) indicate that SRGAN allows for a better segmentation of cracks than upsampling using bilinear interpolation or super-resolving using the remaining four networks.

Name	Bilinear interpolation	U-NetGAN	SRResNet1	SRGAN	SRResNet2	CinCGAN
$J(\hat{C}_{HR}, C_{HR})$	0.556	0.575	0.615	<b>0.679</b>	0.58	0.577
$ \rho - \hat{\rho} /\rho$	0.136	0.197	0.072	<b>0.036</b>	0.144	0.068
$\ f - \hat{f}\ $	0.23	0.074	0.216	<b>0.053</b>	0.614	0.23



**Fig. 5** Distribution of crack sizes. Probability densities of area-equivalent diameters of cracks computed from high-resolution, upsampled low-resolution and super-resolved image data (a), and point-wise absolute errors with respect to the probability density  $f$  computed from high-resolution image data (b).

of the crack size distribution determined from super-resolved images by CinCGAN (which performs reasonably well according to Table 2) is relatively large. In particular, Fig. 5b indicates that such errors especially occur for small crack sizes, which, as mentioned above, only marginally influence the results listed in Table 2.

Overall Table 3 indicates that super-resolving image data can lead to a better segmentation of the crack phase within NMC particles from SEM data than simply upsampling low-resolution images. More precisely, the values of the Jaccard index listed in Table 3 indicate that the application of SRGAN leads to a significant improvement over upsampling of the low-resolution image using bilinear interpolation, i.e., the Jaccard index is 0.556 for the upsampling method, whereas the application of SRGAN leads to a Jaccard index of 0.679. Moreover, we observe that, with a relative error of 0.036, the specific crack density  $\rho$  can be reliably estimated using image data which has been super-resolved by SRGAN. In comparison to this, the relative error using upsampled low-resolution data is 0.136.

Furthermore, Fig. 5a shows that the crack size distribution determined from the upsampled low-resolution data is, in comparison to the distribution determined from high-resolution data, shifted to the right, where the point-wise absolute errors are visualized in Fig. 5b. This discrepancy between the size distributions of cracks determined from low-resolution and high-resolution data can be reduced by super-resolving the low-resolution data with SRGAN. More precisely, Fig. 5b shows that the point-wise absolute errors of the probability density  $\hat{f}$  computed from super-resolved data obtained with SRGAN are close to 0. This is also reflected by the  $\|f - \hat{f}\|$  values in Table 3. Overall, SRGAN outperforms the remaining networks considered in the present paper with respect to the segmentation of cracks. Further improvements of the results achieved with SRGAN could be obtained by considering further discriminators which distinguish between alternative representations (e.g., a representation in some feature space) of super-resolved and high-resolution images<sup>34</sup>. Note that the relatively poor result for the crack size distribution achieved by SRResNet2 can be attributed to noisy predictions of the network which affects the resulting segmentation, see Fig. 4e, f. More precisely, we observe that many cracks are

**Table 4.** Sample details on the composition of NMC532 used in this work.

Toda Li(Ni <sub>0.5</sub> Mn <sub>0.3</sub> Co <sub>0.2</sub> )O <sub>2</sub>	90 wt%
Timcal C45 carbon	5 wt%
Solvay 5130 pVdF binder	5 wt%
Coating thickness	42 μm
Porosity	34.1%
Loading density	11.24 mg cm <sup>-2</sup>

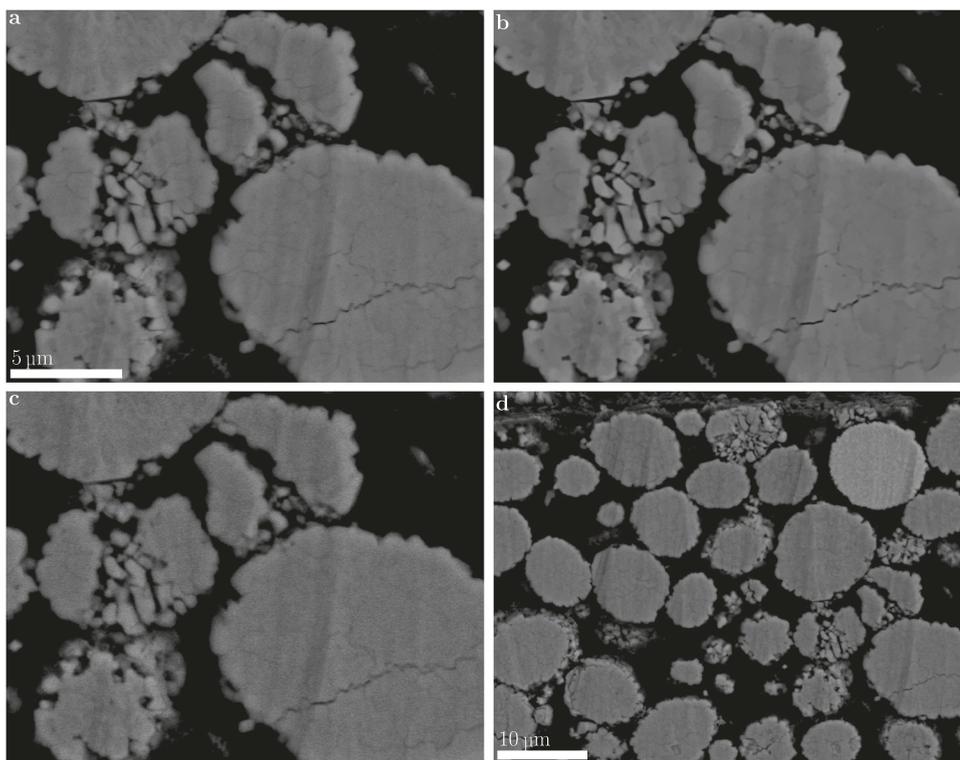
wrongly fragmented into multiple regions, which significantly changes the crack size distribution. This indicates that, in order to perform an in-depth analysis of crack formation in NMC particles, SRResNet2 would require further calibration and/or additional post-processing steps would have to be performed on the images super-resolved by this network. Nevertheless, the super-resolution results achieved by SRGAN suggest that it might be well suited for further analyzes of crack formation in NMC particles.

## METHODS

### Sample details and preparation

Single-sided electrodes were made in a dry room by Cell Analysis, Modeling and Prototyping (CAMP) Facility at Argonne National Laboratory. The NMC cathode composition was given in Table 4 and were used as received. The graphite anode composition and separator can be found in Yang et al. (2021)<sup>71</sup>. The cathode and anode sheets were dried under active vacuum at 120 °C overnight. The cathode electrodes were cut into 14.1 cm<sup>2</sup> area for assembling single layer pouch full cells, paired with graphite electrodes cut into 14.9 cm<sup>2</sup> area sheets. The electrolyte consisted of 1.2 M LiPF<sub>6</sub> in ethylmethyl carbonate:ethylene carbonate (EMC:EC, 7:3 by wt).

The cells were formed, characterized, and cycled using a MACCOR 4000 battery tester at 30 °C in a temperature-controlled chamber. The cells were pre-formed at C/10 for three cycles followed by C/2 rate for three cycles between 3.0 and 4.1 V. The cells were cycled at C/10 for two cycles and then charged at 1C/6C/9C (CC-CV with 10 min total time limit) and



**Fig. 6 Training data.** Experimentally measured high-resolution image of cracked cathode particles before (a) and after denoising (b). The corresponding experimentally measured low-resolution image (c) was determined via registration from the low-resolution reference image (d), where figures a–c use the same length scale.

discharged at C/2 (CC) for 25/225/600 cycles between 3.0 and 4.1 V. Detailed cell testing information can be found in Tanim et al. (2021)<sup>72</sup>.

### Imaging using scanning electron microscopy

Small pieces of the samples ( $1 \times 1 \text{ cm}^2$ ) were cut out from the pristine and fast-charge aged NMC532 cathodes. The samples were mounted in the cross-sectional polisher and polished using 4 kV Ar+ ion beam for 4 h. The resulting cross-section samples were imaged using JEOL JSM-6610LV SEM instrument in the backscattering mode.

### Data preprocessing

Before we train neural networks for performing super-resolution tasks, we preprocess the 46 high-resolution and 102 low-resolution SEM images (see Fig. 6a, d). Since the high-resolution images are noisy we smooth them by deploying the non-local means denoising algorithm<sup>73</sup> (see Fig. 6b).

Then, in a second step we normalize each (single-channel low-resolution and denoised high-resolution) image  $I : \{1, \dots, h\} \times \{1, \dots, w\} \times \{1\} \rightarrow \mathbb{R}$  with height  $h$  and width  $w$  with respect to their mean value  $\mu$  and standard deviation  $\sigma$ , i.e., we compute the normalized version  $I_{\text{normalized}}$  of  $I$  by

$$I_{\text{normalized}} = \frac{1}{\sigma}(I - \mu), \quad (16)$$

where  $\mu = \frac{1}{hw} \sum_{x=1}^w \sum_{y=1}^h I(x, y, 1)$  and  $\sigma^2 = \frac{1}{hw-1} \sum_{x=1}^w \sum_{y=1}^h (I(x, y, 1) - \mu)^2$ . Afterwards, we rescale the pixel values of  $I_{\text{normalized}}$  such that they belong to the interval  $[0, 1]$ , i.e., we compute  $I_{\text{scaled}}$  by

$$I_{\text{scaled}}(x, y, 1) = \begin{cases} 1, & \text{if } I_{\text{normalized}}(x, y, 1) \geq 3, \\ 0, & \text{if } I_{\text{normalized}}(x, y, 1) \leq -3, \\ \frac{1}{6}(I_{\text{normalized}}(x, y, 1) + 3), & \text{else.} \end{cases} \quad (17)$$

The rescaling performed in Eq. (17) accommodates the neural network described in the “Results” section above as its outputs also take values in the interval  $[0, 1]$ . Note that some of the high-resolution images are magnifications of low-resolution images. Thus, using image registration techniques we can determine matching pairs of low-resolution and high-resolution images. More precisely, we use the `matchTemplate()`

function of the Python package OpenCV<sup>74</sup> to determine 33 pairs of low-resolution images with corresponding high-resolution images (see Fig. 6b, c).

### DATA AVAILABILITY

The datasets generated during and/or analyzed during the current study are available from the corresponding authors on reasonable request.

### CODE AVAILABILITY

All formulations and algorithms necessary to reproduce the results of this study are described in the Results and Methods sections and in Ledig et al. (2017)<sup>33</sup>, de Haan et al. (2019)<sup>46</sup>, Jung et al. (2021)<sup>47</sup> and Yuan et al. (2018)<sup>35</sup>.

Received: 4 November 2021; Accepted: 10 March 2022;

Published online: 13 April 2022

### REFERENCES

- Furat, O. et al. Mapping the architecture of single electrode particles in 3D, using electron backscatter diffraction and machine learning segmentation. *J. Power Sources* **483**, 229148 (2021).
- Furat, O. et al. Stochastic modeling of multidimensional particle properties using parametric copulas. *Microsc. Microanal.* **25**, 720–734 (2019).
- Ditscherlein, R. et al. Multiscale tomographic analysis for micron-sized particulate samples. *Microsc. Microanal.* **26**, 676–688 (2020).
- Michael, H. et al. A dilatometric study of graphite electrodes during cycling with X-ray computed tomography. *J. Electrochem. Soc.* **168**, 010507 (2021).
- Neumann, M., Stenzel, O., Willot, F., Holzer, L. & Schmidt, V. Quantifying the influence of microstructure on effective conductivity and permeability: virtual materials testing. *Int. J. Solids Struct.* **184**, 211–220 (2020).
- Lu, X. et al. 3D microstructure design of lithium-ion battery electrodes assisted by X-ray nano-computed tomography and modelling. *Nat. Commun.* **11**, 2079 (2020).
- Kuchler, K. et al. Analysis of the 3D microstructure of experimental cathode films for lithium-ion batteries under increasing compaction. *J. Microsc.* **272**, 96–110 (2018).

8. Girshick, R. Fast R-CNN. In *Proc. IEEE International Conference on Computer Vision*, 1440–1448 (Santiago, Chile, IEEE Computer Society, 2015).
9. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **28**, 91–99 (2015).
10. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. In *Proc. IEEE International Conference on Computer Vision*, 2961–2969 (Venice, Italy, IEEE Computer Society, 2017).
11. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. European Conference on Computer Vision* (eds Ferrari, V., Hebert, M., Sminchisescu, C. & Weiss, Y.) 801–818 (Munich, Germany, Springer, 2018).
12. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. Medical Image Computing and Computer-Assisted Intervention* (eds Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F.) 234–241 (Cham, Switzerland, Springer, 2015).
13. Goodfellow, I. et al. Generative adversarial nets. In *Proceedings of Advances in Neural Information Processing Systems* (eds Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. Q.) 2672–2680 (Montréal, Canada, MIT Press, 2014).
14. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein generative adversarial networks. In *Proc. International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) 214–223 (Sydney, Australia, JMLR, 2017).
15. Kingma, D. P. & Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Proceedings of Advances in Neural Information Processing Systems* (eds Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N. & Garnett, R.) 10236–10245 (Montreal, Canada, 2018).
16. Ardizzone, L., Lüth, C., Kruse, J., Rother, C. & Köthe, U. Guided image generation with conditional invertible neural networks. Preprint at <https://arxiv.org/abs/1907.02392> (2019).
17. Furat, O. et al. Machine learning techniques for the segmentation of tomographic image data of functional materials. *Front. Mater.* **6**, 145 (2019).
18. Evsevlev, S., Paciornik, S. & Bruno, G. Advanced deep learning-based 3D microstructural characterization of multiphase metal matrix composites. *Adv. Eng. Mater.* **22**, 1901197 (2020).
19. Kodama, M. et al. Three-dimensional structural measurement and material identification of an all-solid-state lithium-ion battery by X-ray nanotomography and deep learning. *J. Power Sour. Adv.* **8**, 100048 (2021).
20. Fend, C., Moghiseh, A., Redenbach, C. & Schladitz, K. Reconstruction of highly porous structures from FIB-SEM using a deep neural network trained on synthetic images. *J. Microsc.* **281**, 16–27 (2021).
21. Müller, S. et al. Deep learning-based segmentation of lithium-ion battery microstructures enhanced by artificially generated electrodes. *Nat. Commun.* **12**, 6205 (2021).
22. Ge, M., Su, F., Zhao, Z. & Su, D. Deep learning analysis on microscopic imaging in materials science. *Mater. Today Nano* **11**, 100087 (2020).
23. Prifling, B. et al. Parametric microstructure modeling of compressed cathode materials for Li-ion batteries. *Comput. Mater. Sci.* **169**, 109083 (2019).
24. Neumann, M., Abdallah, B., Holzer, L., Willot, F. & Schmidt, V. Stochastic 3D modeling of three-phase microstructures for predicting transport properties: a case study. *Transp. Porous Media* **128**, 179–200 (2019).
25. Furat, O. et al. Artificial generation of representative single Li-ion electrode particle architectures from microscopy data. *npj Comput. Mater.* **7**, 105 (2021).
26. Allen, J. et al. Quantifying the influence of charge rate and cathode-particle architectures on degradation of Li-ion cells through 3D continuum-level damage models. *J. Power Sour.* **512**, 230415 (2021).
27. Gebäck, T. & Heintz, A. A lattice Boltzmann method for the advection-diffusion equation with Neumann boundary conditions. *Commun. Comput. Phys.* **15**, 487–505 (2014).
28. Mianroodi, J. R., Siboni, N. H. & Raabe, D. Teaching solid mechanics to artificial intelligence—a fast solver for heterogeneous materials. *npj Comput. Mater.* **7**, 99 (2021).
29. Mosser, L., Dubrule, O. & Blunt, M. J. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Phys. Rev. E* **96**, 043309 (2017).
30. Mosser, L., Dubrule, O. & Blunt, M. J. Stochastic reconstruction of an oolitic limestone by generative adversarial networks. *Transp. Porous Media* **125**, 81–103 (2018).
31. Gayon-Lombardo, A., Mosser, L., Brandon, N. P. & Cooper, S. J. Pores for thought: generative adversarial networks for stochastic reconstruction of 3D multi-phase electrode microstructures with periodic boundaries. *npj Comput. Mater.* **6**, 82 (2020).
32. Yu, X. & Porikli, F. Ultra-resolving face images by discriminative generative networks. In *Proc. European Conference on Computer Vision* (eds Leibe, B., Matas, J., Sebe, N. & Welling, M.), 318–333 (Cham, Switzerland, Springer, 2016).
33. Ledig, C. et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 105–114 (Honolulu, HI, USA, IEEE Computer Society, 2017).
34. Fan, L., Wang, Z., Lu, Y. & Zhou, J. An adversarial learning approach for super-resolution enhancement based on AgCl@Ag nanoparticles in scanning electron microscopy images. *Nanomaterials* **11**, 3305 (2021).
35. Yuan, Y. et al. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 814–81409 (Salt Lake City, UT, USA, IEEE Computer Society, 2018).
36. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE International Conference on Computer Vision*, 2242–2251 (Venice, Italy, IEEE Computer Society, 2017).
37. Wang, Z., Liu, D., Yang, J., Han, W. & Huang, T. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE International Conference on Computer Vision*, 370–378 (Santiago, Chile, IEEE Computer Society, 2015).
38. Dong, C., Loy, C. C., He, K. & Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**, 295–307 (2016).
39. Wang, Y., Wang, L., Wang, H. & Li, P. End-to-end image super-resolution via deep and shallow convolutional networks. *IEEE Access* **7**, 31959–31970 (2019).
40. Wang, Z., Chen, J. & Hoi, S. C. H. Deep learning for image super-resolution: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 3365–3387 (2021).
41. Gitman, I., Askes, H. & Sluys, L. Representative volume: existence and size determination. *Eng. Fract. Mech.* **74**, 2518–2534 (2007).
42. Usseglio-Viretta, F. L. E. et al. Resolving the discrepancy in tortuosity factor estimation for Li-ion battery electrodes through micro-macro modeling and experiment. *J. Electrochem. Soc.* **165**, A3403–A3426 (2018).
43. Heenan, T. M. M. et al. Resolving Li-ion battery electrode particles using rapid lab-based X-ray nano-computed tomography for high-throughput quantification. *Adv. Sci.* **7**, 2000362 (2020).
44. Petrich, L. et al. Crack detection in lithium-ion cells using machine learning. *Comput. Mater. Sci.* **136**, 297–305 (2017).
45. Hagita, K., Higuchi, T. & Jinnai, H. Super-resolution for asymmetric resolution of FIB-SEM 3D imaging using AI with deep learning. *Sci. Rep.* **8**, 5877 (2018).
46. de Haan, K., Ballard, Z. S., Rivenon, Y., Wu, Y. & Ozcan, A. Resolution enhancement in scanning electron microscopy using deep learning. *Sci. Rep.* **9**, 12050 (2019).
47. Jung, J. et al. Super-resolving material microstructure image via deep learning for microstructure characterization and mechanical behavior analysis. *npj Comput. Mater.* **7**, 96 (2021).
48. Lugmayr, A., Danelljan, M. & Timofte, R. Unsupervised learning for real-world super-resolution. In *Proc. IEEE/CVF International Conference on Computer Vision Workshop*, 3408–3416 (Seoul, Korea, IEEE Computer Society, 2019).
49. Prajapati, K. et al. Unsupervised single image super-resolution network (usisr-net) for real-world data using generative adversarial network. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1904–1913 (Seattle, WA, USA, IEEE Computer Society, 2020).
50. Luchkin, S. Y. et al. Solid-electrolyte interphase nucleation and growth on carbonaceous negative electrodes for Li-ion batteries visualized with in situ atomic force microscopy. *Sci. Rep.* **10**, 8550 (2020).
51. Shi, W. et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1874–1883 (Las Vegas, NV, USA, IEEE Computer Society, 2016).
52. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proc. IEEE International Conference on Computer Vision*, 1026–1034 (Santiago, Chile, IEEE Computer Society, 2015).
53. Nair, V. & Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *Proc. of the 27th International Conference on Machine Learning* (eds Fürnkranz, J. & Joachims, T.) 807–814 (Madison, WI, USA, Omnipress, 2010).
54. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. 32nd International Conference on Machine Learning* (eds Bach, F. & Blei, D.) 448–456 (Lille, France, JMLR, 2015).
55. Lim, B., Son, S., Kim, H., Nah, S. & Lee, K. M. Enhanced deep residual networks for single image super-resolution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1132–1140 (Honolulu, HI, USA, IEEE Computer Society, 2017).
56. Fan, Y. et al. Balanced two-stage residual networks for image super-resolution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1157–1164 (Honolulu, HI, USA, IEEE Computer Society, 2017).
57. Yang, G., Pennington, J., Rao, V., Sohl-Dickstein, J. & Schoenholz, S. S. A mean field theory of batch normalization. In *Proc. International Conference on Learning Representations* (New Orleans, LA, USA, OpenReview, 2019).

58. Liu, S. & Deng, W. Very deep convolutional neural network based image classification using small training sample size. In *Proc. 3rd IAPR Asian Conference on Pattern Recognition*, 730–734 (Kuala Lumpur, Malaysia, IEEE Computer Society, 2015).
59. Burger, W. & Burge, M. J. *Digital Image Processing* (Springer, 2016).
60. Zheng, T. et al. Super-resolution of clinical CT volumes with modified CycleGAN using micro CT volumes. Preprint at <https://arxiv.org/abs/2004.03272> (2020).
61. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proc. 13th International Conference on Artificial Intelligence and Statistics* (eds Teh, Y. W. & Titterton, M.) 249–256 (Sardinia, Italy, JMLR, 2010).
62. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *Proc. 3rd International Conference on Learning Representations, ICLR 2015* (eds Bengio, Y. & LeCun, Y.) (San Diego, CA, USA, 2015).
63. Abadi, M. et al. TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org (2015).
64. Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004).
65. Westhoff, D., Finegan, D. P., Shearing, P. R. & Schmidt, V. Algorithmic structural segmentation of defective particle systems: a lithium-ion battery study. *J. Microsc.* **270**, 71–82 (2018).
66. Leskovec, J., Rajaraman, A. & Ullman, J. D. *Mining of Massive Datasets* (Cambridge University Press, 2020).
67. Johnson, N. L., Kotz, S. & Balakrishnan, N. *Continuous Univariate Distributions*, vol. 1 (J. Wiley & Sons, 1994).
68. Held, L. & Bové, D. S. *Applied Statistical Inference* (Springer, 2014).
69. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
70. Rudin, L. I., Osher, S. & Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992).
71. Yang, Z. et al. Extreme fast-charging of lithium-ion cells: effect on anode and electrolyte. *Energy Technol.* **9**, 2000696 (2021).
72. Tanim, T. R. et al. Extended cycle life implications of fast charging for lithium-ion battery cathode. *Energy Storage Mater.* **41**, 656–666 (2021).
73. Buades, A., Coll, B. & Morel, J.-M. A non-local algorithm for image denoising. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 60–65 (San Diego, CA, USA, IEEE Computer Society, 2005).
74. Bradski, G. The OpenCV library. *DR DOBBS J.* **25**, 120–125 (2000).

## ACKNOWLEDGEMENTS

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding was provided by the U.S. DOE Office of Vehicle Technology Extreme Fast Charge Program, program manager Samuel Gillard. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes. The authors acknowledge the support from Argonne

National Laboratory, which is a U.S. DOE Office of Science Laboratory operated by UChicago Argonne, LLC, under Contract no. DE-AC02-06CH11357.

## AUTHOR CONTRIBUTIONS

SEM measurements were performed by Z.Y. Neural networks were implemented and calibrated by O.F. and T.K. Main parts of the paper were written by D.P.F., O.F. and Z.Y. All authors discussed the results and contributed to writing of the manuscript. K.S. and V.S. designed and supervised the research.

## FUNDING

Open Access funding enabled and organized by Projekt DEAL.

## COMPETING INTERESTS

The authors declare no competing interests.

## ADDITIONAL INFORMATION

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41524-022-00749-z>.

**Correspondence** and requests for materials should be addressed to Orkun Furat or Donal P. Finegan.

**Reprints and permission information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022