


## ARTICLE OPEN



# Microstructure segmentation with deep learning encoders pre-trained on a large microscopy dataset

Joshua Stuckner<sup>1</sup> , Bryan Harder<sup>1</sup> and Timothy M. Smith<sup>1</sup>

This study examined the improvement of microscopy segmentation intersection over union accuracy by transfer learning from a large dataset of microscopy images called MicroNet. Many neural network encoder architectures were trained on over 100,000 labeled microscopy images from 54 material classes. These pre-trained encoders were then embedded into multiple segmentation architectures including UNet and DeepLabV3+ to evaluate segmentation performance on created benchmark microscopy datasets. Compared to ImageNet pre-training, models pre-trained on MicroNet generalized better to out-of-distribution micrographs taken under different imaging and sample conditions and were more accurate with less training data. When training with only a single Ni-superalloy image, pre-training on MicroNet produced a 72.2% reduction in relative intersection over union error. These results suggest that transfer learning from large in-domain datasets generate models with learned feature representations that are more useful for downstream tasks and will likely improve any microscopy image analysis technique that can leverage pre-trained encoders.

npj Computational Materials (2022)8:200; <https://doi.org/10.1038/s41524-022-00878-5>

## INTRODUCTION

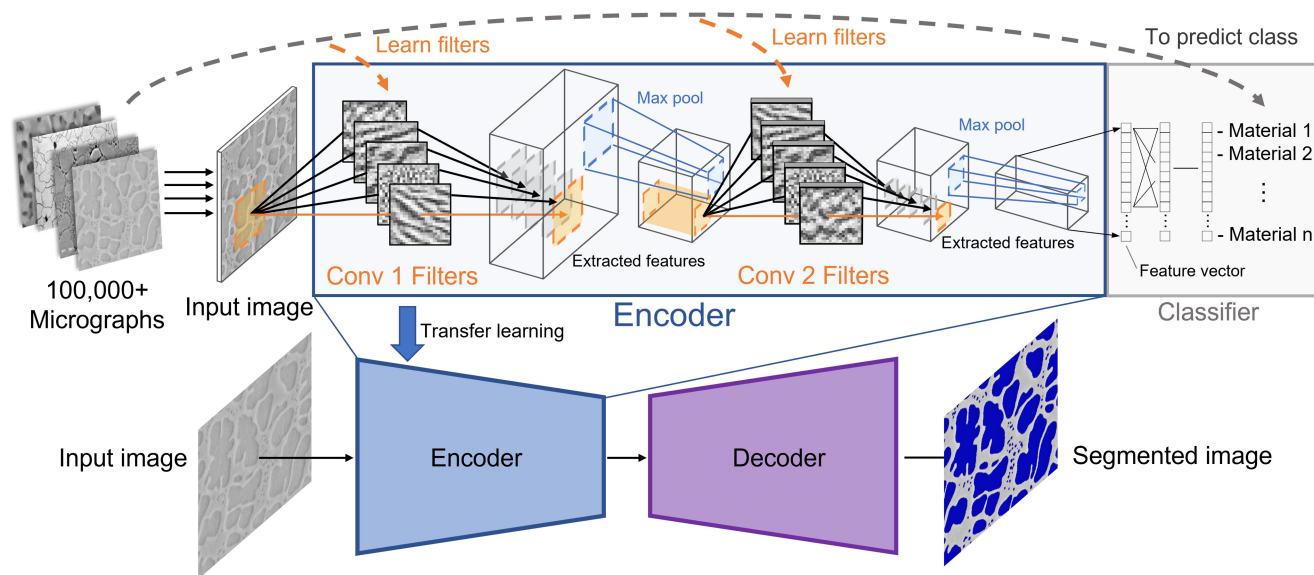
Establishing processing-structure-property relationships is critical to the design and improvement of materials. Microscopy image segmentation is often the first and hardest step in quantifying material structure, which is the central link in processing-structure-property relationships. Traditional microstructure quantification requires numerous manual measurements on a micrograph (e.g., refs. 1,2), is tedious, time-consuming, and prone to bias. Automatic segmentation using classic computer vision techniques such as image thresholding and morphology operations<sup>3,4</sup> is much faster and repeatable, but difficult to implement and often not robust to slight changes in imaging or sample conditions. Recently, convolutional neural networks (CNN) pre-trained on ImageNet<sup>5</sup> have produced superior microscopy classification and segmentation results and are much easier to implement<sup>6–16</sup>. However, segmentation CNNs require expensively labeled training data to operate well and ImageNet pre-training does not adequately alleviate this problem when transferred to microscopy segmentation tasks because many of the learned filters are not applicable (e.g., those adapted to detect dogs). Therefore, we created MicroNet, a large dataset containing over 100,000 labeled microscopy images. Here, we report that leveraging transfer learning from classification models pre-trained on MicroNet rather than ImageNet produces segmentation models with higher intersection over union (IoU) accuracy during one-shot and few-shot learning and with higher accuracy on out-of-distribution test images from different chemical composition, etching, and imaging conditions than the training images.

Semantic segmentation with CNNs is performed with encoder-decoder type architectures, which offer state-of-the-art performance on benchmark datasets such as the cityscapes dataset<sup>17</sup>. The encoder uses learned convolutional filters to extract semantic information from the input image, transforming the image data into a latent representation vector. The decoder then maps the extracted information to each pixel location in the image to

generate a pixelwise classification prediction of the objects in original image (i.e., semantic segmentation).

Training data for semantic segmentation is expensive and time consuming to create. Transfer learning can be used to supplement small training datasets by transferring learned filters from a model trained on another similar task, such as image classification, which is significantly easier to create training data for. Transfer learning is successful when the filters that the model learns from training on the first task are directly applicable to the target task. To leverage transfer learning with encoder-decoder architectures, a CNN is first trained to perform image classification on a large image dataset as shown in the schematic in Fig. 1. Multi-class classification is a common source of pre-training data because the data is relatively cheap to obtain compared to other tasks and the filters learned from this task are useful for other downstream tasks such as semantic segmentation<sup>5</sup>. The classification model uses an encoder with many convolutional layers to extract a feature representation vector from the image and a classification head, which contains several fully connected neural network layers, to make a classification prediction based on the extracted feature representation. The convolutional layers learn to extract useful features for classification during training by learning useful image filters. These learned filters are likely to be useful for other image analysis tasks such as segmentation. Transfer learning is applied when the trained convolution layers from an image classification model are copied directly to the encoder in an encoder-decoder segmentation model. ImageNet contains images of everyday life and is a common source of pre-training image classification data. The convolutional filters that are used to classify ImageNet images have also been applied very successfully to microscopy segmentation. However, recent work has shown that the first few layers of VGG models (a powerful early CNN classification model that is still widely used) are highly useful for transfer learning to microscopy segmentation while the deeper layers are not<sup>18</sup>. This is because the initial layers detect simple features like edges, corners, and simple textures, which are likely to appear in microscopy images,

<sup>1</sup>Materials and Structures Division, NASA Glenn Research Center, 21000 Brookpark Rd, Cleveland, OH 44135, USA. ✉email: [joshua.stuckner@nasa.gov](mailto:joshua.stuckner@nasa.gov)



**Fig. 1 Schematic of pre-training CNN encoders on MicroNet and embedding into a segmentation model through transfer learning.** First, a classification model (top) with a convolutional encoder (blue box) and dense classifier head (gray box) is trained to predict the class of each material by learning filters (Conv filters, orange) which extract relevant features into a feature vector. CNNs contain many layers of convolutional filters, though only two are shown in the illustration. Through transfer learning (blue arrow) the learned filters are copied into an encoder-decoder segmentation model (bottom) which then learns to segment microscopy images with less training data than without transfer learning.

while the deeper layers detect higher level features such as dog ears and car tires, which do not appear in microscopy images.

The central hypothesis of the work presented here is that because higher level feature detectors from models trained on ImageNet do not transfer well to microscopy segmentation, the full advantages of transfer learning are not realized. Therefore, we trained classification models on a large dataset of microscopy images called MicroNet so that the models could learn to detect higher level microstructure features that do not appear in pictures of everyday life such as grain boundaries, precipitates, and oxide layers. We show that transfer learning from models trained on MicroNet rather than ImageNet produces more accurate segmentation results with less training data (in one experiment, improving the IoU from 74.8% to 93% when training on a single image) and is more robust to changing imaging and sample conditions (improving the IoU accuracy from 72.5% to 78.5% on out-of-distribution images in another experiment).

## RESULTS

### Pre-training classification models

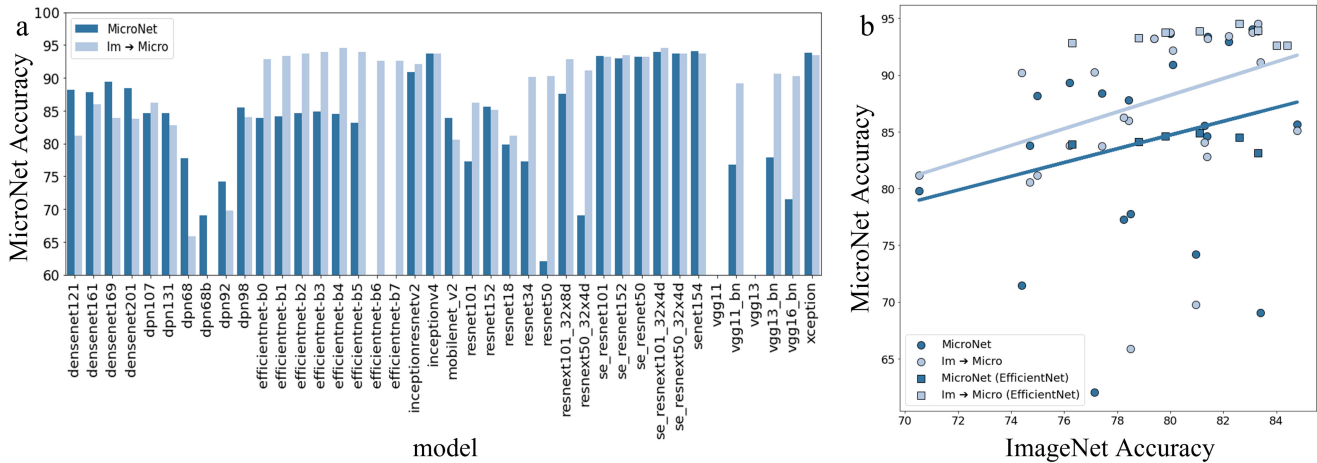
Seventy-six models were trained to classify MicroNet images into one of 54 classes. Forty models were initially pre-trained on ImageNet before training on MicroNet and 36 were randomly initialized. Pre-training on ImageNet allows useful filters learned from classifying natural images to be reused and finetuned for microscopy classification through transfer learning. Initial ImageNet filters that are not useful may be replaced during training on MicroNet but may still be beneficial by providing a better initialization. On average, models initialized with ImageNet weights converged about 20% faster than those initialized with the random initialization. The classification accuracy of these models on the MicroNet validation set are shown in Fig. 2. The best classification model was EfficientNet-b4 pre-trained on ImageNet, which achieved a classification accuracy of 94.5%. SENet-154 achieved the highest accuracy of the models trained from scratch with an accuracy of 94.0%. The EfficientNet, ResNet, and VGG models showed a strong benefit from pre-training on ImageNet. However, it is interesting to note that some

architectures, including the squeeze-and-excitation (SE) and inception families of models, showed no benefit from initial pre-training on ImageNet. For the DenseNet and MobileNet architectures, pre-training was detrimental. Besides architecture quality, variability in performance could be partially explained by the random batches given to the models during training. There is no obvious trend between model size and the benefit of ImageNet pre-training for classification accuracy.

An open question is whether progress in developing vision models, which are optimized for natural images (images of everyday life, often captured with consumer cameras), will transfer well to microscopy images or if architecture design is overfit to natural images. From the trendlines in Fig. 2b it can be seen that in general, model architectures reported to perform better on ImageNet classification tended to perform better when trained to classify MicroNet data. However, a notable exception to the trend is the EfficientNet model architectures, which are indicated by the square markers in Fig. 2b. EfficientNet architectures that performed better when trained to classify ImageNet did not perform better when trained to classify MicroNet, whether pre-trained on ImageNet or not. A significant amount of testing was done during the development of EfficientNet to optimize the depth, width, resolution scaling, and other hyperparameters to develop an architecture that performed well on ImageNet<sup>19</sup>. A microscopy dataset of comparable size to ImageNet would be helpful to study the full extent to which progress on natural image processing transfers to microscopy image processing. Such a study could help determine whether it would be fruitful to design architectures, scaling rules, and techniques specifically for microscopy analysis instead of largely borrowing best practices from large research efforts on natural images. From our results it seems that there would be value in optimizing the scaling compound coefficient used in EfficientNet for microscopy specific data.

### Segmentation datasets

The real measure of the value of the trained classification models (i.e., pre-trained encoders) is how well the image representations learned by the encoders transfer to downstream tasks such as



**Fig. 2 Accuracy of classification models.** **a** The prediction accuracy of classification models on the MicroNet validation set. The models indicated with dark blue were randomly initialized and trained from scratch while the “Im → Micro” models shown in light blue were pre-trained on ImageNet and then finetuned on the MicroNet data. **b** Comparison of each architecture’s reported accuracy when trained to classify ImageNet data versus each architecture’s the classification accuracy when trained to classify MicroNet data in this study. EfficientNet models are indicated with square markers and all others with circular markers.

**Table 1.** Number of images in the train, validation, and test splits of each experimental dataset.

Experiment	# Train	# Val	# Test
Super-1	10	4	4
Super-2	4	4	4
Super-3	1	4	4
Super-4	4	4	5*
EBC-1	18	3	3
EBC-2	4	3	3
EBC-3	1	3	3

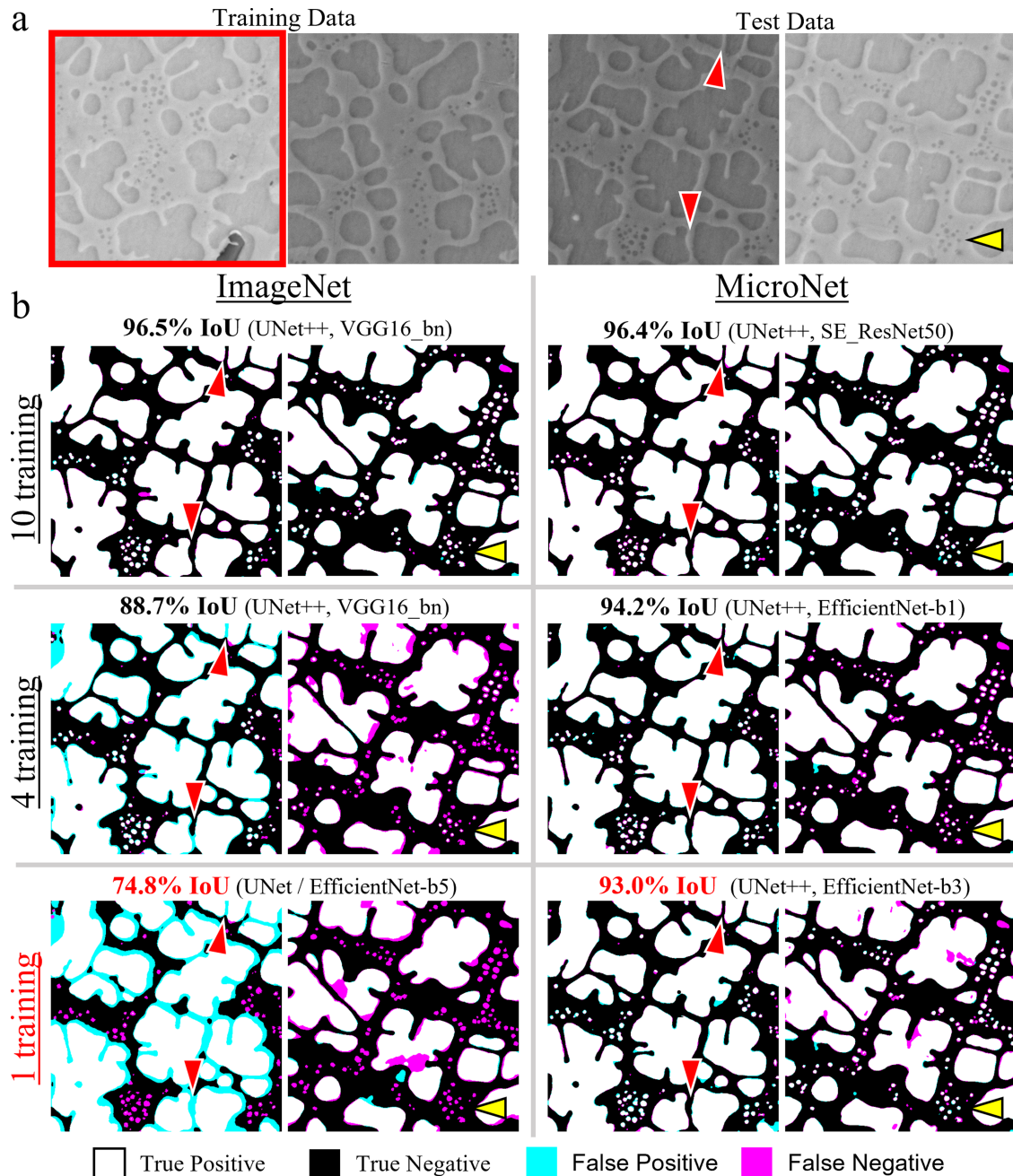
\*The test images in Super-4 were taken under different imaging conditions than the train and validation sets.

segmentation. The pre-trained encoders were applied through transfer learning to seven segmentation tasks which came from two materials: nickel-based superalloys (hereinafter referred to as Super) and environmental barrier coatings (EBC). The number of images in each dataset split is shown in Table 1. Super-1 and EBC-1 contain the full set of labeled data from their respective materials. Super-2 and EBC-2 have only four images in the training set to evaluate the models’ performance in few-shot learning, where the model is trained on only a few training samples. Super-3 and EBC-3 had only one image in the training set to evaluate performance during one-shot learning. Super-4 had test images that were taken under different imaging and sample conditions to test how well the models would generalize to unseen out-of-distribution data (e.g., images from different microscopes, microscopists, microscope settings, sample preparation conditions, or different research groups).

### Ni-superalloy segmentation

Pre-training on MicroNet led to a significant increase in accuracy for few-shot and one-shot learning on the Super datasets. The training data splits and segmentation accuracy masks for these datasets are shown in Fig. 3. The training, validation, and test splits had similar looking images and an equal number of dark and light contrast images in each split. The train split for Super-3 had only one image which is outlined in red in Fig. 3a and did not contain

dark contrast images. The performance of the best models pre-trained on MicroNet and ImageNet for each of the Super-1 to Super-3 datasets are displayed above the segmentation accuracy masks in Fig. 3. When training on Super-1 with ten training images (Fig. 3b, first row), both the ImageNet and MicroNet models performed well and accurately segmented the secondary and tertiary precipitates. On Super-2 with only four training images (Fig. 3b, second row), the MicroNet model had only a slight reduction in accuracy (96.4% IoU to 94.2% IoU) while the ImageNet model had a large reduction in accuracy from 96.5% to 88.7%. With four training images the ImageNet model failed to identify many of the tertiary precipitates in the dark contrast images as indicated by the yellow triangles. The ImageNet model also over-segmented and combined some of the secondary precipitates (indicated by the red triangles). With four training images, the segmentation output of the MicroNet model allowed for much more accurate size and morphology measurements of the precipitates. When training on Super-3 with only a single training image (Fig. 3b, third row), the improvement of MicroNet over ImageNet was even more striking. The ImageNet model was reduced to 74.8% IoU and the segmentation was unusable for measuring size statistics, morphology, or even the area fraction of the precipitates. In the darker contrast image, the secondary precipitates were over-segmented and incorrectly combined into one as indicated by the red triangles. The tertiary precipitates were not detected at all by the ImageNet model. Meanwhile, the MicroNet model had a very high accuracy of 93.0% IoU during one-shot learning, which was nearly equal to the accuracy obtained from training on the full dataset. The tertiary precipitates were identified, and the secondary precipitates were properly separated in both the light and dark contrast test images. Even when training on a single image, the MicroNet model produced segmentation masks that could be used to calculate highly accurate size, morphology, area fraction, and other quantitative microstructure metrics. Trying to extract those metrics from the ImageNet one-shot model would be highly misleading and significantly overestimate the size of the secondary precipitates while ignoring the tertiary precipitates. Figure 6 shows the error of precipitate area measurements on the segmented test images compared to measurements on the manually labeled test images. The Super-3 MicroNet model trained on one image had only a 10% error on the secondary precipitates while the ImageNet model had a 90% error (Fig. 6a). Similarly, when measuring the tertiary precipitates, the MicroNet models had only a slight



**Fig. 3 Segmentation results on Ni-superalloys.** **a** shows images from the training and test data splits. Super-1 had ten training images. Super-2 had four images. Super-3 had only one training image which is outlined in red. **b** show the segmentation accuracy masks for the highest accuracy ImageNet and MicroNet models for the first three Super datasets. White pixels indicate true positive predictions, black is true negative, cyan is false positive, and magenta is false negative. The left column shows the models pre-trained on ImageNet. As the number of training images reduce, there is a dramatic reduction in segmentation IoU accuracy. The right column shows the models pre-trained on MicroNet. Even with only 1 training image, the model accuracy is only slightly reduced when pre-training on MicroNet. Red and yellow triangles are placed at the same location in each ground truth and segmentation accuracy mask for the left and right test image respectively. The red triangles indicate an example where ImageNet models over-segment and combine secondary precipitates where the MicroNet models accurately segment the precipitate edges and maintain precipitate separation. The yellow triangles indicate an example where MicroNet models accurately identify tertiary precipitates that were not identified by ImageNet models in the few-shot and one-shot case.

increase in error (about 26% to 38%) when reducing the training images from ten to one (Super-1 to Super-3) while the ImageNet model went from 25% to over 175% error when reducing the training images from ten to one (Fig. 6b). Percent error is higher for tertiary than secondary precipitates for all models because small segmentation errors produced larger percent size

differences in the smaller precipitates. It is interesting to note that in the one-shot case, the MicroNet models produced only slight systematic differences in segmentation predictions due to image contrast compared to models pre-trained on ImageNet despite the lack of darker contrast images in the training data. This suggests that pre-training on MicroNet leads to models that are

**Table 2.** Average performance of models initialized with different pre-training weights for each experiment.

Pre-training	Super-1	Super-2	Super-3	Super-4	EBC-1	EBC-2	EBC-3
None	76.9% ± 22.8%	46.2% ± 7.1%	48.3% ± 6.2%	34.0% ± 7.5%	68.0% ± 31.4%	48.3% ± 27.7%	35.1% ± 10.3%
Imagenet	<b>93.8% ± 7.9%</b>	62.1% ± 12.1%	59.7% ± 7.9%	47.7% ± 14.1%	87.9% ± 19.9%	82.9% ± 17.4%	<b>43.9% ± 7.0%</b>
MicroNet	93.6% ± 8.7%	74.6% ± 14.3%	66.9% ± 13.2%	52.3% ± 10.5%	87.9% ± 18.2%	81.6% ± 17.1%	40.3% ± 6.2%
Im → Micro	85.8% ± 19.2%	<b>74.6% ± 16.2%</b>	<b>70.0% ± 13.9%</b>	<b>52.5% ± 16.2%</b>	<b>88.8% ± 16.6%</b>	<b>81.7% ± 14.7%</b>	41.4% ± 8.2%

On average pre-training on ImageNet followed by pre-training on MicroNet often produced the best results. The highest accuracy score for each dataset is shown in bold. In some cases, the overall best model was pre-trained on MicroNet or ImageNet. Models trained from scratch without any pre-training always performed worse. Experimentation is often required to select the best model. The results from this table can help guide experimentation.

**Table 3.** Average performance of decoder architectures for each experiment.

Decoder	Super-1	Super-2	Super-3	Super-4	EBC-1	EBC-2	EBC-3
DeepLabV3+ <sup>20</sup>	–	–	–	–	86.9% ± 15.3%	76.0% ± 22.4%	–
FPN <sup>55</sup>	–	–	–	–	75.6% ± 34.9%	–	–
LinkNet <sup>54</sup>	84.9% ± 20.1%	59.3% ± 15.5%	53.1% ± 11.9%	42.1% ± 12.8%	81.6% ± 25.8%	65.8% ± 26.2%	–
PAN <sup>57</sup>	–	–	–	–	85.9% ± 17.9%	–	–
PSPNet <sup>56</sup>	–	–	–	–	72.4% ± 28.5%	–	–
Unet <sup>52</sup>	88.0% ± 15.1%	<b>67.0% ± 17.7%</b>	<b>62.3% ± 12.3%</b>	48.4% ± 14.9%	89.9% ± 13.1%	<b>76.7% ± 22.1%</b>	<b>40.3% ± 8.4%</b>
Unet++ <sup>53</sup>	<b>89.9% ± 16.3%</b>	66.6% ± 17.8%	62.1% ± 11.9%	<b>49.3% ± 14.9%</b>	<b>90.3% ± 15.2%</b>	76.5% ± 25.5%	40.0% ± 8.9%

On average and in general, Unet and Unet++ performed the best on the experiments performed here. DeepLabV3+ is a newer segmentation architecture that performs well on natural images and performed quite well on EBC-2. The highest accuracy score for each dataset is shown in bold. The results from this table can help guide experimentation to select the best model for a particular segmentation task.

more robust to changes in imaging or sample conditions. Overall, pre-training on MicroNet produced a 72.2% reduction in relative IoU error in the one-shot case compared to ImageNet.

The average model performance across all encoder and decoder combinations when initialized with different pre-training weights are shown for each experiment in Table 2. Although the error bars are large because a few models failed to converge during training, it appears that on average when using less training data in Super-2 and Super-3, pre-training with ImageNet-then-MicroNet was slightly better than pre-training with MicroNet or ImageNet alone. Pre-training with MicroNet showed better performance than ImageNet. With no pre-training (randomly initialized encoder weights), model performance was significantly reduced. Table 3 shows that the UNet and UNet++ decoders were consistently more accurate than LinkNet decoders for Super-1 to Super-3. From Table 4, none of the encoder architectures demonstrated clearly superior performance on Super-1 to Super-3, although some performed poorly on average.

### Assessing the generalization to new image conditions

Segmentation accuracy on micrographs with different sample and imaging conditions was greatly improved when pre-training on MicroNet. Figure 4 shows the segmentation accuracy of the Super-4 experiment where the test data was from a different distribution than the training and validation data (shown in Fig. 3a). The test images for Super-4 contained micrographs from a different alloy (Fig. 4, top row), several different etching conditions (rows 2–4), and poor imaging or sample preparation conditions (bottom row). For this experiment, the top MicroNet model had an IoU of 78.5% compared to 72.5% for the top ImageNet model. Although the accuracy on this extremely out-of-distribution test set was less than the in-distribution test sets, consider how useful the MicroNet segmentation masks would be for extracting useful morphology statistics such as size and shape compared to those produced by ImageNet pre-training. The red triangles in Fig. 4

indicate several examples where the ImageNet model commonly over-segmented and combined the secondary precipitates making accurate size and shape analysis impossible. The MicroNet model was significantly more accurate in identifying the separation between secondary precipitates allowing for accurate precipitate size and shape analysis. A careful observer may notice a couple rare instances of MicroNet over-segmentation in the bottom row, but the separation accuracy is extraordinarily improved over the ImageNet model. MicroNet's performance on segmenting the small tertiary precipitates is also vastly superior to ImageNet with several examples indicated by the yellow triangles in Fig. 4. In the first three rows, the ImageNet model did not identify the vast majority of the tertiary precipitates while the MicroNet model was able to successfully identify and segment them allowing for downstream size and morphology analysis. Automatic measurements on the secondary and tertiary precipitate sizes (Super-4, Fig. 6a, b, respectively) were significantly more accurate with the MicroNet model. The higher accuracy of pre-trained MicroNet encoders on out-of-distribution data indicates that pre-trained MicroNet encoders are more general and useful for comparing results between research groups, microscopes, sample preparation conditions, and imaging conditions. The MicroNet models have higher usability on a much wider range of sample and imaging conditions without having to label additional training data.

On average, the EfficientNet family of encoders had the highest performance on the out-of-distribution data in the Super-4 experiment as shown in Table 4. Table 3 shows that UNet++ was the highest performing decoder on average and UNet was nearly as good. Table 2 gives the average results of different pre-training weights on Super-4. Pre-training on ImageNet-then-MicroNet or MicroNet gave the best results on average (52.5% and 52.3%, respectively) and was better than pre-training on ImageNet (47.7%) and significantly better than without pre-training (34.0%).

**Table 4.** Average performance of encoder architectures for each experiment.

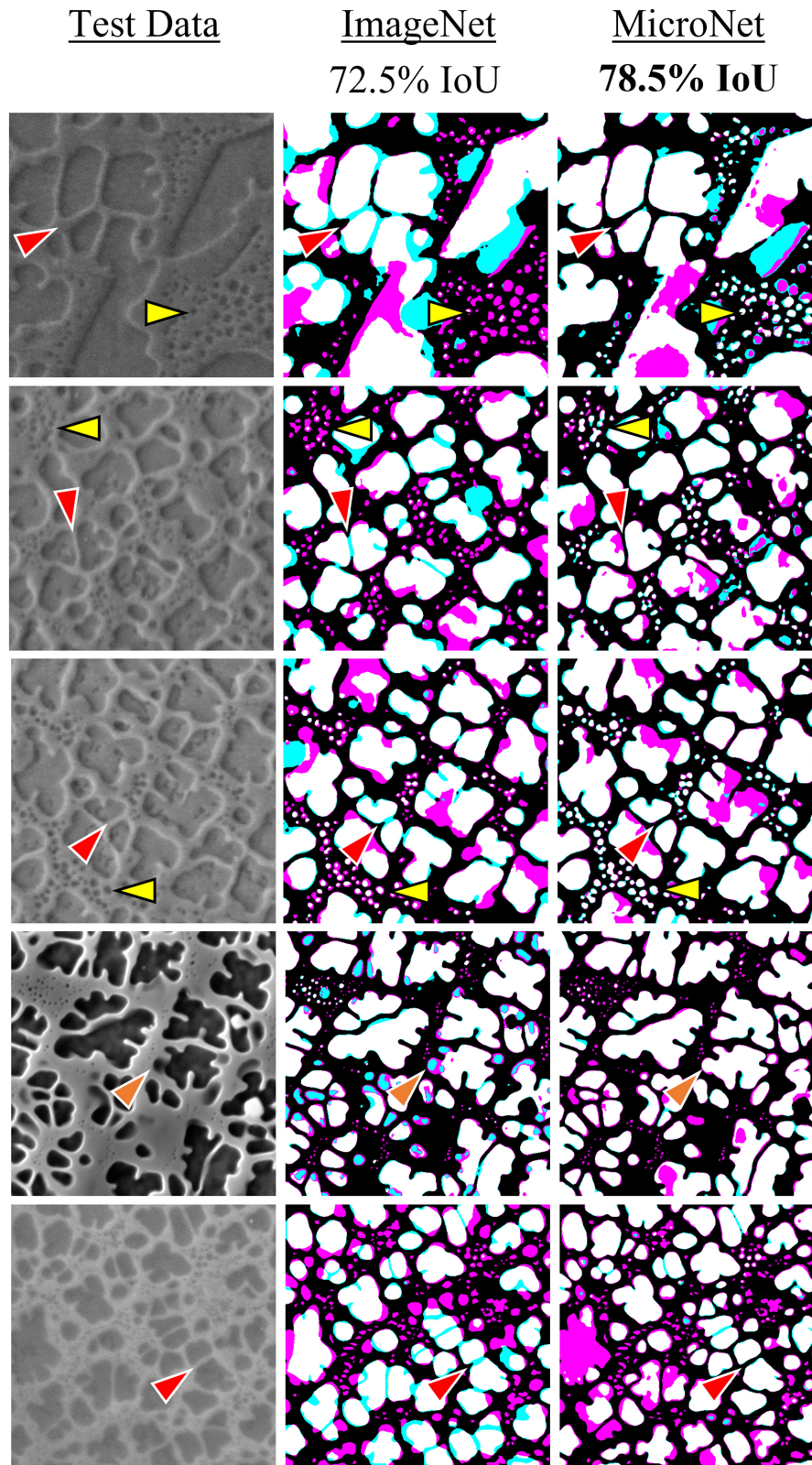
Encoder	Super-1	Super-2	Super-3	Super-4	EBC-1	EBC-2	EBC-3
DenseNet121 <sup>41</sup>	–	–	–	–	89.1% ± 6.2%	80.9% ± 10.8%	38.3% ± 5.3%
DenseNet161 <sup>41</sup>	–	–	–	–	90.9% ± 4.8%	86.1% ± 5.5%	39.6% ± 2.9%
DenseNet169 <sup>41</sup>	–	–	–	–	90.8% ± 4.7%	81.5% ± 9.2%	–
DenseNet201 <sup>41</sup>	–	–	–	–	90.3% ± 5.1%	<b>86.5% ± 4.8%</b>	–
dpn107 <sup>42</sup>	–	–	–	–	90.7% ± 5.6%	80.7% ± 22.1%	–
dpn131 <sup>42</sup>	–	–	–	–	86.7% ± 18.0%	79.8% ± 16.8%	–
dpn68 <sup>42</sup>	–	–	–	–	74.6% ± 29.2%	61.9% ± 27.1%	40.1% ± 5.2%
dpn68b <sup>42</sup>	–	–	–	–	69.6% ± 29.6%	56.9% ± 29.4%	36.5% ± 8.5%
dpn92 <sup>42</sup>	–	–	–	–	84.1% ± 17.8%	78.3% ± 12.3%	–
dpn98 <sup>42</sup>	–	–	–	–	87.1% ± 17.9%	72.9% ± 26.9%	–
EfficientNet-b0 <sup>19</sup>	77.7% ± 23.9%	56.9% ± 14.3%	51.1% ± 12.6%	53.7% ± 15.8%	59.0% ± 41.5%	55.9% ± 32.7%	34.7% ± 16.9%
EfficientNet-b1 <sup>19</sup>	66.9% ± 26.4%	62.9% ± 19.5%	52.8% ± 19.7%	58.7% ± 13.9%	60.0% ± 40.7%	57.1% ± 33.7%	40.7% ± 7.1%
EfficientNet-b2 <sup>19</sup>	65.1% ± 26.0%	66.9% ± 21.6%	51.1% ± 18.4%	<b>59.4% ± 14.6%</b>	68.6% ± 36.0%	57.6% ± 33.6%	–
EfficientNet-b3 <sup>19</sup>	69.4% ± 26.9%	68.1% ± 22.0%	62.4% ± 19.1%	59.2% ± 11.4%	71.6% ± 33.9%	60.0% ± 33.0%	–
EfficientNet-b4 <sup>19</sup>	74.8% ± 26.0%	67.9% ± 18.7%	58.9% ± 15.2%	57.2% ± 13.9%	70.3% ± 35.3%	61.4% ± 33.0%	–
EfficientNet-b5 <sup>19</sup>	76.3% ± 23.0%	70.5% ± 22.5%	61.3% ± 17.2%	57.7% ± 17.3%	73.4% ± 35.6%	64.6% ± 31.3%	36.4% ± 17.0%
Inception-ResNet-V2 <sup>44</sup>	94.5% ± 3.1%	64.4% ± 12.2%	63.0% ± 11.0%	43.1% ± 11.1%	87.9% ± 10.4%	75.6% ± 25.7%	38.2% ± 5.2%
Inception-V4 <sup>44</sup>	91.6% ± 8.8%	76.2% ± 19.9%	<b>69.1% ± 13.3%</b>	48.6% ± 12.1%	84.0% ± 26.4%	73.2% ± 26.9%	39.3% ± 6.3%
MobileNet-V2 <sup>46</sup>	–	–	–	–	71.3% ± 34.5%	60.5% ± 34.2%	45.7% ± 6.5%
ResNet-101 <sup>43</sup>	–	–	–	–	86.4% ± 18.1%	76.1% ± 15.7%	38.5% ± 8.6%
ResNet-152 <sup>43</sup>	–	–	–	–	86.2% ± 18.7%	75.5% ± 23.7%	–
ResNet-18 <sup>43</sup>	–	–	–	–	89.8% ± 5.9%	82.1% ± 7.3%	–
ResNet-34 <sup>43</sup>	94.3% ± 2.1%	57.9% ± 11.7%	59.5% ± 7.2%	35.8% ± 11.3%	91.1% ± 5.3%	80.2% ± 9.7%	–
ResNet-50 <sup>43</sup>	91.2% ± 7.2%	48.9% ± 11.9%	55.3% ± 5.9%	32.2% ± 9.4%	82.5% ± 24.2%	81.2% ± 9.6%	38.2% ± 3.4%
ResNeXt-101_32x8d <sup>47</sup>	95.3% ± 1.5%	57.5% ± 12.1%	57.3% ± 4.4%	39.8% ± 8.8%	–	–	–
Resnext-50_32x4d <sup>47</sup>	90.9% ± 8.9%	54.7% ± 11.6%	54.6% ± 7.8%	39.0% ± 9.6%	81.9% ± 23.9%	69.9% ± 27.0%	33.4% ± 5.1%
SE_ResNet-101 <sup>48</sup>	93.3% ± 8.4%	66.4% ± 13.1%	57.8% ± 9.4%	47.7% ± 11.2%	93.3% ± 4.0%	84.6% ± 7.8%	42.4% ± 6.6%
SE_ResNet-152 <sup>48</sup>	95.2% ± 1.9%	60.6% ± 12.5%	57.4% ± 6.0%	38.6% ± 9.3%	93.0% ± 4.4%	82.7% ± 19.2%	40.2% ± 4.1%
SE_ResNet-50 <sup>48</sup>	94.8% ± 4.3%	63.1% ± 15.3%	57.3% ± 5.4%	39.8% ± 9.2%	88.9% ± 18.2%	81.6% ± 14.6%	<b>47.3% ± 8.2%</b>
SE_ResNeXt-101_32x4d <sup>47</sup>	95.4% ± 1.3%	67.3% ± 19.3%	60.8% ± 13.5%	44.2% ± 8.6%	91.8% ± 6.0%	79.5% ± 18.7%	45.6% ± 5.3%
SE_ResNeXt-50_32x4d <sup>47</sup>	<b>96.0% ± 0.3%</b>	67.3% ± 17.4%	59.6% ± 5.1%	44.4% ± 9.4%	<b>92.6% ± 4.0%</b>	83.3% ± 14.0%	41.6% ± 2.5%
SENet-154 <sup>48</sup>	94.5% ± 4.2%	<b>76.6% ± 17.5%</b>	63.2% ± 12.7%	51.0% ± 15.0%	91.9% ± 6.7%	75.9% ± 30.0%	42.2% ± 4.5%
VGG-13_bn <sup>39</sup>	94.2% ± 6.6%	65.2% ± 17.8%	64.3% ± 14.8%	46.6% ± 15.0%	–	–	–
VGG-16_bn <sup>39</sup>	95.2% ± 2.1%	66.6% ± 19.0%	66.2% ± 14.7%	40.2% ± 17.0%	87.5% ± 18.7%	84.0% ± 5.1%	33.8% ± 8.4%
Xception <sup>45</sup>	93.8% ± 4.5%	61.3% ± 10.8%	54.1% ± 7.8%	39.3% ± 9.9%	92.0% ± 5.1%	74.6% ± 33.9%	42.5% ± 9.7%

There was not a single best encoder architecture and the performance of the encoder architectures varied significantly between experiments. The highest accuracy score for each dataset is shown in bold. Often the newer encoder architectures such as the SE, inception, ResNeXt, and EfficientNet families tended to perform better than the older architectures such as VGG and ResNet, however this was not always the case. The large variation of encoder performance shows that experimenting with many model architectures and evaluating performance on a validation set is often required to achieve higher accuracy.

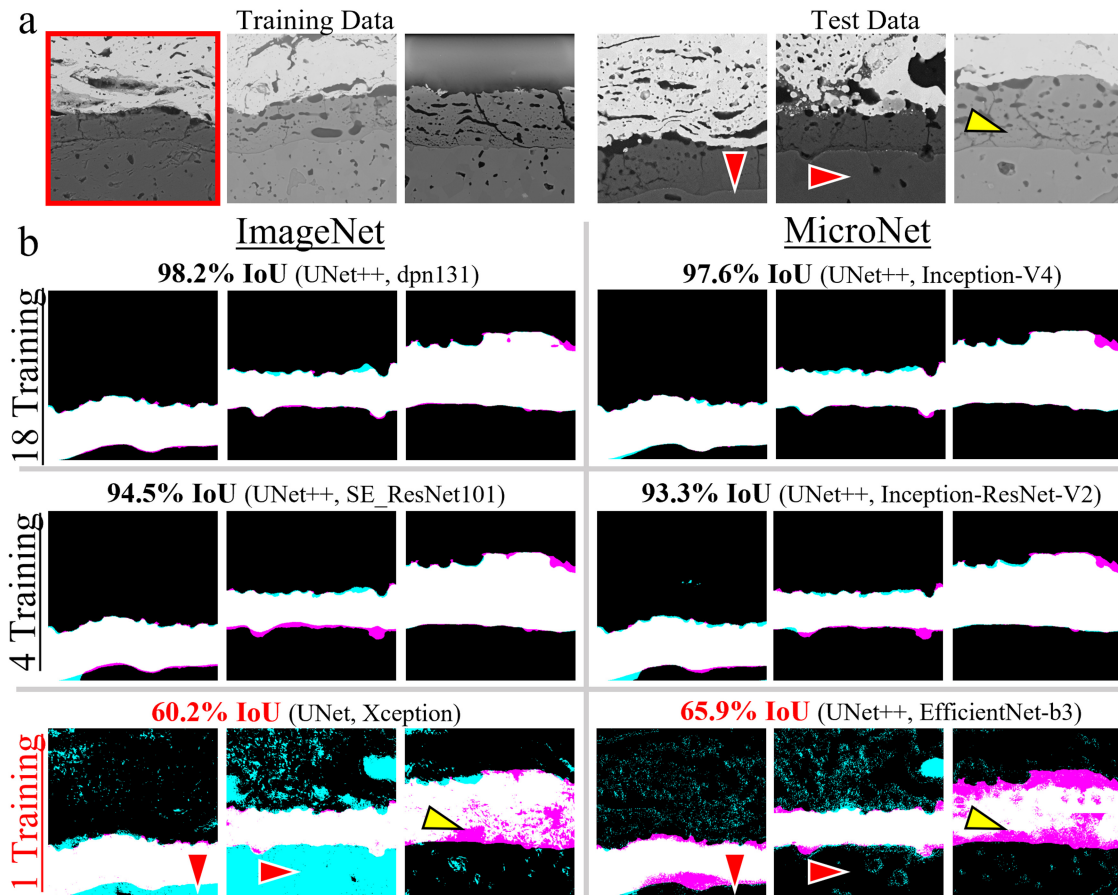
### Environmental barrier coating segmentation

When pre-training on MicroNet the top model showed significant improvement for the one-shot learning case on the environmental barrier coating datasets (EBC-3) compared to the top ImageNet model. From Table 3, the best decoder architecture on average appeared to be UNet or UNet++, although DeepLabV3+ was not evaluated for all datasets and appeared to be promising. The top models for each EBC dataset used the UNet++ decoder except one which used UNet (Fig. 5). There was not a clearly best encoder architecture for the EBC datasets as shown in Table 4, although some architectures were clearly inferior. Table 2 shows that on average across all encoders and decoders, ImageNet models performed slightly better for EBC-2 and EBC-3 while pre-training on ImageNet-then-MicroNet gave the best average performance on EBC-1. Models that were not pre-trained had significantly degraded performance. However, it is difficult to determine which

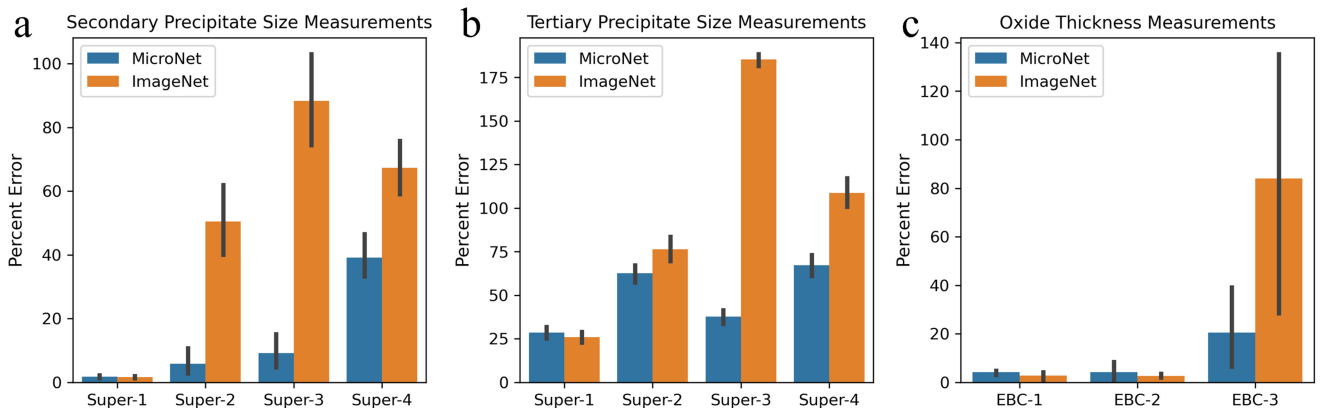
pre-training method was clearly superior from the average results because of the wide error bars and the occasional poor performance of a few models that randomly failed to converge. A clearer picture of the best pre-training method is given by the segmentation results of the best ImageNet and MicroNet model for each EBC dataset as shown in Fig. 5. On EBC-1 and EBC-2 when training with 18 and 4 images respectively, there was not a significant difference between pre-training on MicroNet and ImageNet, although ImageNet pre-training was slightly better for the top models. For EBC-3, when training on the single image outlined in red in Fig. 5a, the top MicroNet model saw a 14.3% reduction in relative error compared to the top ImageNet model (65.9% IoU vs. 60.2% IoU). The ImageNet model failed to distinguish between the substrate and the thermally grown oxide layer (indicated by the red triangles in Fig. 5) making it impossible to accurately measure oxide thickness. Meanwhile the one-shot



**Fig. 4 Accuracy of Super-4 segmentation models evaluated on test data with unseen imaging conditions.** The left column shows the images from the test set. The middle column shows the IoU accuracy masks for the best ImageNet model (UNet++, EfficientNet-b0). The right column shows the same for best top MicroNet model (UNet, EfficientNet-b1). Each row shows the test image and accuracy masks of the same image. The accuracy mask colors represent the same as in Fig. 3. The red triangles indicate example locations where the ImageNet model over-segmented and connected the tertiary precipitates which the MicroNet model accurately segmented. The yellow triangles indicate example locations where the ImageNet model failed to identify tertiary precipitates that the MicroNet model successfully identified. The orange triangles in the fourth row indicate one of many example locations where the ImageNet model improperly identified the corner of a secondary precipitate as a tertiary precipitate.



**Fig. 5 Results of EBC segmentation.** **a** shows examples from the train and test splits of the EBC datasets. The single training image for EBC-3 outlined in red. **b** shows the segmentation results for the top ImageNet and MicroNet models for each EBC experiment. The accuracy mask colors represent the same as in Fig. 3. The red triangles indicate example locations on the test image where the ImageNet models were not able to distinguish between the substrate and the thermally grown oxide when training on a single image. The yellow arrows indicate locations where the thermally grown oxide was under segmented when training on a single image with both the MicroNet and ImageNet models.



**Fig. 6 Plots showing percent error in size measurements performed on segmented images.** **a** compares the average percent measurement error when measuring the size of individual secondary precipitates from test set images segmented by the best models in the Super experiments. Error bars are the standard deviation of the error across all the precipitates in the test set images. **b** compares the error of tertiary precipitate size measurements. **c** compares the average percent error of thermally grown oxide thickness measurements after performing simple morphology operations on the segmented test images. The error bars are the standard deviation across the three test images.

MicroNet model was highly useable for oxide thickness measurements after simple morphological operations (such as binary opening which is useful for removing small objects to remove the noise in the segmentation mask). The error of oxide thickness measurements made on the segmented images is shown in

Fig. 6c. Measurement errors using models trained EBC-1 and EBC-2 were less than 5% for both MicroNet and ImageNet with ImageNet performing slightly better. For the one-shot case (EBC-3) MicroNet segmented the oxide with enough accuracy to obtain an average measurement error of 20% while the ImageNet model produced



unable segmentations leading to an average measurement error above 80%. Both models under-segmented the thermally grown oxide in the lighter contrast test image as indicated by the yellow arrow in Fig. 5. But considering that the models were trained on only one training image, that the lighter contrast image looked quite different from the training image, and that the image contained only a single instance of the oxide layer, the accuracy is surprisingly good.

## DISCUSSION

From a practical standpoint, choosing the best pre-training source and encoder architecture for a particular microscopy analysis task and dataset may require some experimentation. Here, we provide some guidelines based on the results presented in this manuscript and our unreported experience using the models. The provided code makes it easy to vary these parameters and experiment with different combinations. In general, encoders pre-trained on ImageNet and then further pre-trained on MicroNet often provided the best results. We suggest starting with ImageNet-then-MicroNet pre-trained encoders. On the Super-4 task, when applying the model to out of distribution data, Table 4 shows that pre-training on MicroNet and ImageNet-then-MicroNet was about equal on average while pre-training on ImageNet was worse. The top three models trained on Super-4 were pre-trained on MicroNet. Unet and Unet++ decoders were consistently the best in our experiments. However, DeeplabV3+ should also be considered due to its reportedly improved performance on natural images and ability to capture multi-scale context<sup>20</sup>. The performance of the encoder architectures varied significantly between experiments. We found that the newer encoder architectures such as the SE, inception, ResNeXT, and EfficientNet families tended to perform better than the older architectures such as VGG and ResNet, however this was not always the case. There was a moderate correlation between the encoders' MicroNet classification accuracy and their downstream segmentation accuracy with a Pearson's correlation coefficient of 0.55 and 0.58 on EBC-2 and Super-2 respectively. In short, experimentation is often required to achieve the best results and the provided code makes that easy; however, users are encouraged to start with the UNet++ decoder and an encoder with high MicroNet classification accuracy that was pre-trained on ImageNet-then-MicroNet.

Transfer learning works to the extent that a data-rich initial task is similar to the target task such that the learned representations from the initial task are applicable to the target. However, transfer learning has limitations and may not always provide the best results. One potential drawback is negative transfer where the transferred knowledge has a negative impact on the target task<sup>21</sup>. This could be caused in part by the loss of the nice starting condition properties provided by random Kaiming initialization without the benefit of useful starting filters ideally provided by transfer learning. The root cause of negative transfer is the divergence of the source data distribution to the target data distribution<sup>22</sup>. Thus, on many microstructure tasks, MicroNet may be less prone to negative transfer than ImageNet. However, microstructures are extremely diverse and in some instances the target task may be significantly different from both ImageNet and MicroNet and require the sourcing of additional task specific training data. Transfer learning also restricts the target task to the pre-trained model architectures that are available. Some tasks may require specialized model architectures such as those that can handle 3D microstructure data or extremely large images or for applications that require fast execution with small models that aren't suitable for distinguishing between large numbers of classes. Ultimately, the accuracy from pre-training may not be sufficient for the target application. In those cases, large amounts of labeled data from the target domain along with the flexibility of

hyperparameter and architecture optimization precluded by transfer learning may be required.

Transfer learning from CNN encoders pre-trained on MicroNet produced more accurate segmentation models with a higher IoU with significantly less training data than pre-training on ImageNet. MicroNet encoders also generalized to better to unseen data with different imaging or sample conditions. This is significant because creating labeled training data for segmentation tasks is expensive and time consuming and the labeled data cannot account for all possible imaging and sample conditions that the model should be expected to perform accurately on. By producing higher accuracy with less training data and generalizing better to out-of-distribution microscopy images, this technique shows promise to produce segmentation results that are more accurate and comparable between microscopes, microscopists, and research groups, thus increasing the utility and shareability of the trained models.

The improved segmentation accuracy suggests that the MicroNet pre-trained encoders generate superior microstructure feature representations and will likely improve the accuracy of other deep learning microscopy analysis tasks that commonly utilize pre-trained ImageNet encoders, making this technique broadly and generally applicable. The following microstructure analysis tasks that use pre-trained ImageNet encoders would likely benefit from MicroNet pre-trained encoders with only a small change to the code. Using deep regression to directly predict material properties or grain size<sup>23,24</sup>. Using the final feature vector from the encoder (with or without dimensionality reduction) as input into other ML algorithms such as support vector machines, gaussian processes, or random forests to predict material properties<sup>25,26</sup>. Extracting feature vectors from the entire encoder to predict properties<sup>6</sup>. Automatically classifying important features or defects in microstructure images<sup>27–29</sup>. Classifying EBSD patterns<sup>15</sup>. Classifying small patches for semantic segmentation<sup>14</sup>. Performing object detection and instance segmentation<sup>10</sup>. The pre-trained MicroNet encoders have been made readily available and the provided code contains examples to demonstrate how MicroNet encoders can be downloaded and used in existing projects that leverage pretrained ImageNet encoders by adding only a couple lines of code.

Ultimately better microstructure representations can be used to build more accurate data-driven models that establish processing-structure-property relationships to improve inverse design through techniques such as active learning. Inverse design allows practitioners to first determine target material properties based on design criteria and iteratively discover how to produce that material with far fewer experiments, saving significant time, money, and labor. Structure is the central link in processing-structure-property relationships and accurate microstructure segmentation and feature extraction is critical to quantitatively establishing these relationships.

## METHODS

### Description of datasets

A large dataset called MicroNet, containing 110,861 microscopy images, was created to pre-train classification models to be used as encoders in segmentation models. The majority of MicroNet images were sourced inhouse with additional images from the UltraHigh Carbon Steel Micrograph DataBase<sup>30</sup>, the Aversa Scanning Electron Microscopy (SEM) dataset<sup>31</sup>, synthetic SEM powder data<sup>32</sup>, SEM images from the Materials Data Repository hosted by the National Institute of Standards and Technology, and a photovoltaic dataset<sup>33</sup>. On average MicroNet images were much larger than ImageNet (1048 × 741 versus 469 × 387 pixels) giving the MicroNet dataset a pixel equivalence of 474,323 ImageNet images for the encoders to learn from. For comparison, ImageNet contains 14 million images across 20,000 classes (often combined into 1000 subclasses). MicroNet contained 54 classes and was split into train/validation sets with 50 images for each class in the validation set representing a 97.5/

2.5 training/validation split. The large training/validation ratio was required so that each class had the same number of images in the validation set and the smallest classes had at least two-thirds of their images in the training set. Fifty images per class was deemed large enough to acquire reliable validation accuracy to prevent overfitting during training. While the validation set was balanced, the training set had some class imbalance with several classes each containing less than 0.2% of the total images and one class containing 12.5% of the images. Most classes had over 1100 images or one percent of the training set. MicroNet contained images from optical, scanning electron, and transmission electron microscopes and included numerous material classes including metals, polymers, ceramics, and composites. Over half the images were from scanning electron microscopes and almost a third were from optical microscopes. Only one class contained synthetic data and accounted for less than two percent of the dataset. About 70% of the images had a single grayscale channel while others, especially the optical microscopy images, were three-channel RGB images. Micrographs from a variety of imaging techniques and material types were included to enhance the universality of transfer-learning from MicroNet for material microstructure quantification tasks.

Separating the data into appropriately labeled classes was not a straightforward task. Material classifications are almost continuously hierarchical in nature with broad categories such as metals and polymers at the top which can be subdivided an arbitrary number of times based on composition or processing. Due to the stochastic nature of materials processing, each material specimen could even be considered a unique class, like each individual human. ImageNet contains hierarchical labels, but most pre-trained encoders are standardly trained on 1000 pre-determined classes which, perhaps arbitrarily, group types of passenger cars into a single class and keeps many dog breeds in separate classes. Here, inhouse data was labeled in the following manner. 1. Images were obtained pre-grouped in folders based on the researcher and experiment that produced them. 2. Image folders were compared to ensure class uniqueness (separable by at least differences in composition or processing) and combined where appropriate. 3. Classes with less than 200 images (training and validation) were combined with another class if they shared a common root class and excluded from the dataset otherwise. (For example, several folders containing Ti-6Al-4V with differences in processing conditions were combined into a single class.) 4. All images were examined to ensure basic quality and label accuracy. Publicly available micrograph images from external sources were labeled in a similar manner starting from the original labels. Some classes were significantly different in appearance than other classes (e.g., the synthetic powder class and images of SiC-SiC composites) and were much easier to classify than classes that were similar in appearance (e.g., several classes of Ni-superalloys with slight differences in formulation). The classes were subdivided as much as possible without imparting too much class imbalance to encourage the models to learn better representations required to distinguish between similar classes.

The segmentation algorithms were tested on two sets of material micrographs: SEM images of a Ni-superalloy and cross-sectional SEM images of a SiC/SiC EBC with a thermally grown oxide layer. The Ni-superalloy had three classes to segment: a matrix phase, secondary precipitates (large blobs), and tertiary precipitates (small blobs). The EBC had two classes: an oxide layer and the background (not oxide layer). The segmentation training data was annotated using the GNU Image manipulation Program (GIMP).

### Training classification models

Many CNN classification models were trained on MicroNet to use as segmentation encoders through transfer learning. Models for each architecture were initialized with weights downloaded from the PyTorch model zoo from models that had been pre-trained on ImageNet. Additional models for most of the classification architectures were also initialized with random weights following Kaiming initialization to evaluate the effect of encoder training on MicroNet from scratch (VGG-11, VGG-13, EfficientNet-b6, and EfficientNet-b7 architectures were not trained from scratch). The Kaiming initialization was designed to reduce the exploding or vanishing gradient problem by encouraging the variance of activations to be similar across network layers when using rectified linear unit (ReLU) activation functions<sup>34,35</sup>. During training, any grayscale images were converted to color by copying the gray channel to the three RGB channels and all images were preprocessed by mean centering and normalizing each channel according to the ImageNet statistics in order to best utilize pre-trained weights<sup>36</sup>. Image transformations were used to augment the

training data set including random resizing, horizontal and vertical flipping, rotation, photometric distortions, and added noise. After random resizing, training images were cropped in a random location to the size required by the encoder architecture (usually  $224 \times 224$  pixels). Validation images were resized while preserving the aspect ratio such that the smaller side was the appropriate size, then the larger side was center cropped to produce a square input image. Each training image was augmented randomly each epoch. An epoch is one training iteration where the entire training data set is input to the model and the model weights are updated to better fit the desired output of the full training set. Training was performed on four Nvidia Quadro GV100 32 GB GPUs using the PyTorch Python library<sup>37</sup> in a similar fashion to ref.<sup>38</sup>. Optimization was performed with stochastic gradient descent with a momentum of 0.9 and an initial learning rate of 0.1 that decayed by 10% every 30 epochs in a manner consistent with ImageNet pre-training. Weight decay, which is the fraction each model parameter is reduced each epoch, was  $1e-4$ . A batch size (the number of samples shown to the model for each weight update) of 1024 was used where possible and reduced for larger models due to hardware memory constraints. Models were trained until there was no improvement to the validation score using early stopping with a patience of 30 epochs. The following encoder architectures were tested in this work: VGG<sup>39</sup> (with and without batch normalization<sup>40</sup>), DenseNet<sup>41</sup>, dual path networks (dprn)<sup>42</sup>, EfficientNet<sup>19</sup>, ResNet<sup>43</sup>, Inception-V4<sup>44</sup>, Inception-Resnet-V2<sup>44</sup>, Xception<sup>45</sup>, MobileNet-V2<sup>46</sup>, ResNeXt<sup>47</sup>, and SE-Net<sup>48</sup>.

### Training segmentation models

Segmentation models were trained on four Nvidia Quadro GV100 32 GB GPUs using PyTorch<sup>37</sup> and the segmentation models library<sup>49</sup>. Training data images were converted to color and each channel was normalized and mean centered in the same manner as the classification data. Training data augmentation included random cropping to  $512 \times 512$  pixels; random changes to contrast, brightness, and gamma; and added blur or image sharpening. The superalloy data was also randomly flipped vertically and horizontally and rotated while the EBC data was only horizontally flipped to preserve orientation significance. While not applied here, random resizing could be included when desired to make the models robust to changes in magnification or image resolution. The Adam<sup>50</sup> optimizer was used during training with a learning rate of  $2e-4$  until there was no improvement on the validation dataset for 30 epochs followed by training with a learning rate of  $1e-5$  until early stopping after another 30 epochs with no validation improvement. While the different segmentation architectures used in this study have been trained by others with various optimizers, Adam was used here on all segmentation models for consistency and because initial testing showed good results when using Adam. Minibatching was not used to train the segmentation models (i.e., the model weights were updated once each epoch after seeing the entire training set). The model validation metric to determine early stopping and compare different models was IoU. The loss function was a weighted sum of balanced cross entropy (BCE) and dice loss<sup>51</sup> with a 70% weighting towards BCE. BCE measures the cross-entropy error of segmentation predictions and works well when there is class imbalance by weighting the error of smaller area classes more heavily. Dice loss, also known as the F1-score, balances the error contribution of false negatives and false positives by taking the harmonic mean of precision and recall. Numerically, dice loss is very similar to IoU. Initial testing showed higher IoU validation accuracy with the combined loss function than either independently or when using IoU as the loss function directly. This is likely because BCE has more stable gradients while dice loss is more robust to imbalanced classes and similar to the real objective of maximizing IoU. The following decoder architectures were tested: Unet<sup>52</sup>, Unet++<sup>53</sup>, Linknet<sup>54</sup>, FPN<sup>55</sup>, PSPNet<sup>56</sup>, PAN<sup>57</sup>, and DeepLabV3+<sup>20,58</sup>.

### Size measurements

The size of the segmented secondary and tertiary Ni-superalloy precipitates in the ground truth and segmented images were measured by calculating the number of pixels covered by each precipitate. Individual precipitates were identified using a connected components algorithm implemented in the scikit-image python library<sup>59,60</sup>. The percent error of precipitate sizes was compared to measurements on the corresponding precipitate in the hand labeled ground truth images. The average percent error of the size measurements along with error bars indicating the standard deviations are shown in Fig. 6.

EBC oxide thickness measurements were performed on the ground truth and segmented images after performing simple binary morphology operations using scikit-image to reduce the segmentation noise (especially required for EBC-3). First, morphological closing followed by morphological opening was applied to remove small false negatives and small false positives respectively and to smooth the segmentation boundary. Then small, enclosed gaps up to 1000 pixels<sup>2</sup> in the segmented oxide layer were removed while falsely identified and separated regions of oxide layer up to 1000 pixels<sup>2</sup> were removed using the `remove_small_objects` function. Oxide thickness was measured by multiplying a medial axis transform of the image by a distance transform to produce a radius measurement at each pixel along the backbone of the oxide using the `medial_axis` function in scikit-image. Noise reduction using morphological operations was required to perform the medial axis transform. The average percent error and standard deviation of the average oxide thickness measured on the segmented images compared to the ground truth images are shown in Fig. 6.

## DATA AVAILABILITY

Data that supports the findings of this study, including labeled segmentation data and pre-trained encoders trained on MicroNet, are available at <https://github.com/nasa/pretrained-microscopy-models>.

## CODE AVAILABILITY

All code necessary to apply this technique and supports the findings in this study is available at <https://github.com/nasa/pretrained-microscopy-models>.

Received: 6 January 2022; Accepted: 20 August 2022;

Published online: 19 September 2022

## REFERENCES

- ASTM E112. Standard test methods for determining average grain size E112-10. *ASTM E112-10* **96**, 1–27 (2010).
- ASTM E45-18a, A. Standard test methods for determining the inclusion content of steel. *ASTM Book of Standards Volume: 03.01* (2018).
- Stuckner, J., Frei, K., McCue, I., Demkowicz, M. J. & Murayama, M. AQUAM: An open source Python package and GUI for the automatic quantitative analysis of morphologically complex multiphase materials. *Comput. Mater. Sci.* **139**, 320–329 (2017).
- Smith, T. M. et al. Characterization of nanoscale precipitates in superalloy 718 using high resolution SEM imaging. *Mater. Charact.* **148**, 178–187 (2019).
- Deng, J. et al. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* 248–255 (IEEE, 2009).
- DeCost, B. L., Lei, B., Francis, T. & Holm, E. A. High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel. *Microsc. Microanal.* **25**, 21–29 (2019).
- Roberts, G. et al. DefectNet—a deep convolutional neural network for semantic segmentation of crystallographic defects in advanced microscopy images. *Microsc. Microanal.* **25**, 164–165 (2019).
- Goetz, A. et al. Addressing materials' microstructure diversity using transfer learning. *npj Comput. Mater.* **8**, 1–13 (2022).
- Senanayake, N. M. & Carter, J. L. W. Computer vision approaches for segmentation of nanoscale precipitates in nickel-based superalloy IN718. *Integr. Mater. Manuf. Innov.* **9**, 446–458 (2020).
- Cohn, R. et al. Instance segmentation for direct measurements of satellites in metal powders and automated microstructural characterization from image data. *JOM* **73**, 1–14 (2021).
- Stan, T., Thompson, Z. T. & Voorhees, P. W. Building towards a universal neural network to segment large materials science imaging datasets. in *Developments in X-Ray Tomography XII* vol. 11113 297–302 (SPIE, 2019).
- Groschner, C., Choi, C., Nguyen, D., Ophus, C. & Scott, M. Machine learning for high throughput HRTEM analysis. *Microsc. Microanal.* **25**, 150–151 (2019).
- Potocek, P. et al. Sparse scanning electron microscopy data acquisition and deep neural networks for automated segmentation in connectomics. *Microsc. Microanal.* **26**, 403–412 (2020).
- Akers, S. et al. Rapid and flexible segmentation of electron microscopy data using few-shot machine learning. *npj Comput. Mater.* **7**, 1–9 (2021).
- Kaufmann, K., Lane, H., Liu, X. & Vecchio, K. S. Efficient few-shot machine learning for classification of EBSD patterns. *Sci. Rep.* **11**, 1–12 (2021).
- Durmaz, A. R. et al. A deep learning approach for complex microstructure inference. *Nat. Commun.* **12**, 1–15 (2021).
- Cordts, M. et al. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 3213–3223 (2016).
- Luo, Q., Holm, E. A. & Wang, C. A transfer learning approach for improved classification of carbon nanomaterials from TEM images. *Nanoscale Adv.* **3**, 206–213 (2021).
- Tan, M. & Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* 6105–6114 (2019).
- Chen, L.-C., Papandreou, G., Schroff, F. & Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv Prepr. arXiv1706.05587* (2017).
- Pan, S. J. & Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**, 1345–1359 (2009).
- Wang, Z., Dai, Z., Póczos, B. & Carbonell, J. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 11293–11302 (2019).
- Noraas, R., Somanath, N., Giering, M. & Olusegun, O. O. Structural material property tailoring using deep neural networks. In *AIAA Scitech 2019 Forum* 1703 (2019).
- Holm, E. A. et al. Overview: Computer vision and machine learning for microstructural characterization and analysis Department of Materials Science and Engineering, Carnegie Mellon University, Pittsburgh, PA USA Department of Materials Science and Engineering, MIT, Ca. 1–5.
- Larmuseau, M. et al. Compact representations of microstructure images using triplet networks. *npj Comput. Mater.* **6**, 1–11 (2020).
- Larmuseau, M. et al. Race against the Machine: can deep learning recognize microstructures as well as the trained human eye? *Scr. Mater.* **193**, 33–37 (2021).
- Kusche, C. et al. Large-area, high-resolution characterisation and classification of damage mechanisms in dual-phase steel using deep learning. *PLoS One* **14**, e0216493 (2019).
- Azimi, S. M., Britz, D., Engstler, M., Fritz, M. & Mücklich, F. Advanced steel microstructural classification by deep learning methods. *Sci. Rep.* **8**, 1–14 (2018).
- Kitahara, A. R. & Holm, E. A. Microstructure cluster analysis with transfer learning and unsupervised learning. *Integr. Mater. Manuf. Innov.* **7**, 148–156 (2018).
- DeCost, B. L. et al. UHCSDB: UltraHigh carbon steel micrograph database: tools for exploring large heterogeneous microstructure datasets. *Integr. Mater. Manuf. Innov.* **6**, 197–205 (2017).
- Aversa, R., Modarres, M. H., Cozzini, S., Ciancio, R. & Chiusole, A. Data descriptor: The first annotated set of scanning electron microscopy images for nanoscience. *Sci. Data* **5**, 1–10 (2018).
- DeCost, B. L. & Holm, E. A. A large dataset of synthetic SEM images of powder materials and their ground truth 3D structures. *Data Br.* **9**, 727–731 (2016).
- Karimi, A. M. et al. Automated pipeline for photovoltaic module electroluminescence image processing and degradation feature classification. *IEEE J. Photovolt.* **9**, 1324–1335 (2019).
- He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* 1026–1034 (2015).
- Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* 807–814 (2010).
- Szegedy, C. et al. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 1–9 (2015).
- Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv Prepr. arXiv1912.01703* (2019).
- Tan, M. et al. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2820–2828 (2019).
- Simonyan, K., & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv Prepr. arXiv1409.1556* (2014).
- Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* 448–456 (PMLR, 2015).
- Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 4700–4708 (2017).
- Chen, Y. et al. Dual path networks. *Adv. Neural Inf. Process. Syst.* **2017-December**, 4468–4476 (2017).
- 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).
- Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 31 (2017).

45. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 1251–1258 (2017).
46. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 4510–4520 (2018).
47. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 1492–1500 (2017).
48. Hu, J., Shen, L. & Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 7132–7141 (2018).
49. Yakubovskiy, P. Segmentation Models Pytorch. *GitHub Repos.* (2020).
50. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv Prepr. arXiv1412.6980* (2014).
51. Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S. & Cardoso, M. J. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support* 240–248 (Springer, 2017).
52. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* vol. 9351 234–241 (Springer, 2015).
53. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N. & Liang, J. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* **39**, 1856–1867 (2019).
54. Chaurasia, A. & Culurciello, E. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)* 1–4 (IEEE, 2017).
55. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125 (2017).
56. Zhao, H., Shi, J., Qi, X., Wang, X. & Jia, J. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 2881–2890 (2017).
57. Li, H., Xiong, P., An, J. & Wang, L. Pyramid attention network for semantic segmentation. *arXiv Prepr. arXiv1805.10180* (2018).
58. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)* 801–818 (2018).
59. Van der Walt, S. et al. scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014).
60. Fiorio, C. & Gustedt, J. Two linear time union-find strategies for image processing. *Theor. Comput. Sci.* **154**, 165–181 (1996).

## ACKNOWLEDGEMENTS

This work was supported by the NASA Transformational Tools and Technologies (TTT) project under the Transformative Aeronautics Concept Program within the

Aeronautics Research Mission Directorate. Computational resources were provided by the Scientific Computing and Visualization Team in the Information and Applications Division within the Office of the CIO at NASA Glenn. Inhouse MicroNet images were provided by researchers from the Materials and Structures Division at NASA Glenn Research Center, many of which were captured using facilities provided by the NASA Glenn ASG lab.

## AUTHOR CONTRIBUTIONS

J.S. conceived and designed the study, developed the software, evaluated results, and contributed to formal analysis and writing of the original draft. T.S. and B.H. performed SEM experiments, provided datasets, and evaluated results. All authors reviewed the final manuscript.

## COMPETING INTERESTS

The authors declare no competing interests.

## ADDITIONAL INFORMATION

**Correspondence** and requests for materials should be addressed to Joshua Stuckner.

**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/>.

This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2022