High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel

Brian L. DeCost[1]   Bo Lei[2]   Toby Francis[2]
Elizabeth A. Holm[2]

[1]Material Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg MD, 20899, USA
[2]Materials Science and Engineering, Carnegie Mellon University, Pittsburgh PA, 15213, USA

Abstract

We apply a deep convolutional neural network segmentation model to enable novel automated microstructure segmentation applications for complex microstructures typically evaluated manually and subjectively. We explore two microstructure segmentation tasks in an openly-available ultrahigh carbon steel microstructure dataset [1, 2]: segmenting cementite particles in the spheroidized matrix, and segmenting larger fields of view featuring grain boundary carbide, spheroidized particle matrix, particle-free grain boundary denuded zone, and Widmanstätten cementite. We also demonstrate how to combine these data-driven microstructure segmentation models to obtain empirical cementite particle size and denuded zone width distributions from more complex micrographs containing multiple microconstituents. The full annotated dataset is available on materialsdata.nist.gov [3].

1   Introduction

Quantitative microstructure analysis is central to materials engineering and design. Traditionally this entails careful measurements of volume fractions,

1

size distributions, and shape descriptors for familiar microstructural features such as grains and second-phase particles. These quantities are connected to theoretical and/or empirical models for materials properties, e.g. grain boundary [4] or particle [5] strengthening mechanisms. Contemporary microstructure segmentation methods rely on specialized image processing pipelines that often require expert tuning for application to a particular microstructure system. Furthermore, the microstructures accessible to quantitative analysis are limited by the use of segmentation algorithms that rely on low-level image features (intensity and connectivity constraints). In this work, we apply deep learning methods for image segmentation to complex microstructure data, with the goal of extending the reach of quantitative analysis to microstructure systems that are currently evaluated subjectively or through laborious manual annotation.

Since 2012, deep learning methods [6] have dominated many computer vision applications[1], including object recognition and detection, scene summarization, semantic segmentation, and depth map prediction. The success of deep learning is often attributed to the ability of convolutional neural networks (CNNs) to learn to effectively represent the hierarchical structure of visual data, composing low-level image features (edges, color gradients) into higher level features corresponding to abstract qualities of the image subject (e.g. object parts). Recently, materials scientists have begun exploring a limited set of applications of contemporary computer vision techniques for flexible and generic microstructure representation. [8] and [9] explore these techniques in the context of microstructure classification. [10] and [11] use pretrained CNN representations to study relationships between processing conditions and microstructure via dimensionality-reduction and visualization techniques. [12] use a CNN segmentation model to identify constituent phases in steel microstructures.

In this report, we train a pixelwise CNN [13] to segment microstructures at a high level of abstraction, and investigate the potential for this technique to enable quantitative microstructure analyses that conventionally would require a large amount of hands-on image processing. We evaluate the feasibility of this approach on a subset of the openly available Utrahigh Carbon Steel (UHCS) microstructure dataset [1, 14, 15]. CNNs can distinguish between the four principal microconstituents in this heat-treated UHCS: proeutectoid cementite network, fields of spheroidite particles, the ferritic matrix in the particle-free denuded zone near the network, and Widmanstätten laths. We

---

[1] See [7] for a comprehensive introduction to deep learning methods, including architectural and training choices.

also train a network to segment individual spheroidite particles, and briefly explore automated microstructure metrology techniques enabled by this kind of powerful segmentation model. Our training data and annotations for both microstructure segmentation tasks will be publicly available through the NIST materials resource registry [3].

Our primary contributions are:

- Establishing two novel microstructure segmentation benchmark datasets **1**
- Connecting microstructure science to the deep semantic segmentation literature **2**
- Exploring novel means of expanding contemporary quantitative microstructure measurement techniques to more complex structures **3**

For microstructure scientists, CNN-based microstructure segmentation tools require an initial investment in annotation and training, but can enable longer-term or larger-scale research and characterization efforts. This trade-off is particularly attractive for its potential to enable microstructure-based material qualification by making it easier/cheaper to obtain statistical data on high-level microstructure features known to mediate critical engineering properties of materials (e.g. particle size distributions; denuded zone widths, and particle coarsening kinetics). In industrial settings where reliance on semi-automated segmentation techniques is common, the barrier to entry is even lower because the training data has already been collected. CNN-based microstructure segmentation tools also offer a path forward to high-throughput microstructure quantification techniques for accelerated alloy design and processing optimization, where acquisition and analysis of high-quality microstructure data is often a limiting factor.

2 Methods

2.1 Segmentation model

Recently a variety of deep CNN architectures have been developed for dense pixel-level tasks [16], such as semantic segmentation [17], edge detection, depth map, and surface normal prediction [18]. Conceptually, a modern deep CNN computes a highly nonlinear function through a layerwise composition of convolution, activation, and pooling (i.e. downsampling) functions, the parameters of which are learned from large annotated datasets by some variant of stochastic gradient descent [6, 7]. Classification CNNs reduce

an input image to a single latent feature vector, where CNNs designed for pixel-level tasks produce a latent representation for every pixel of the input image. This is typically accomplished by upsampling the intermediate feature maps via a fixed bilinear interpolation [13, 19] or a learned deconvolution operation [20]. In the latter class of networks, popular architectures include SegNet [17], Bayesian SegNet [21], U-Net [22] with heavy data augmentation, and fully-convolutional DenseNets [23]. In particular, U-Net [22] was designed for application to medical image segmentation tasks with small dataset sizes, relying on strong data augmentation to achieve good performance.

2.1.1 PixelNet architecture

The PixelNet [13] architecture is illustrated schematically in Figure 1. PixelNet applies bilinear interpolation to intermediate feature maps to form hypercolumn features $h(x) = [conv_1(x), conv_2(x), \ldots conv_5(x)]$, which represent each pixel in the input image with information drawn from multiple scales. A non-linear classifier implemented as a multi-layer perceptron (MLP, i.e. a traditional artificial neural network (ANN)) maps the hypercolumn features to the corresponding pixel-level target. Instead of computing dense high-dimensional feature maps at the input resolution as in other popular pixel prediction networks, at training time PixelNet performs a sparse upsampling to efficiently obtain hypercolumn features only for a small sample of the input pixels.[2] This is attractive for quickly training segmentation networks from scratch with small training sets because it reduces the memory footprint during training and makes training a non-linear predictor with high-dimensional latent representations feasible [13].

The feature extraction portion of our PixelNet variant uses the VGG-16 architecture [24] used by the original PixelNet [13]; this architecture consists of 13 convolution layers and two fully-connected layers $1_1, 1_2, 2_1, 2_2, 3_1, 3_2, 3_3, 4_1, 4_2, 4_3, 5_1, 5_2, 5_3, 6, 7$. The MLP layers in our PixelNet variant consist of 1024 neurons with rectified linear (ReLU) activations [25] ($ReLU(y_i) = \max(0, y_i)$) followed by batch normalization [26]. Following the original PixelNet implementation, our hypercolumn features consist of the highest convolution feature map within each block of the VGG architecture ($\{1_2, 2_2, 3_3, 4_3, 5_3, 7\}$), converting layer 7 to a 7×7 convolution filter as in [20] and [13]. We apply batch normalization [26] to each VGG-16 feature map before upsampling via bilinear interpolation, immediately after the ReLU activations.

---

[2] Our tensorflow implementation of PixelNet is available at https://github.com/bdecost/pixelnet
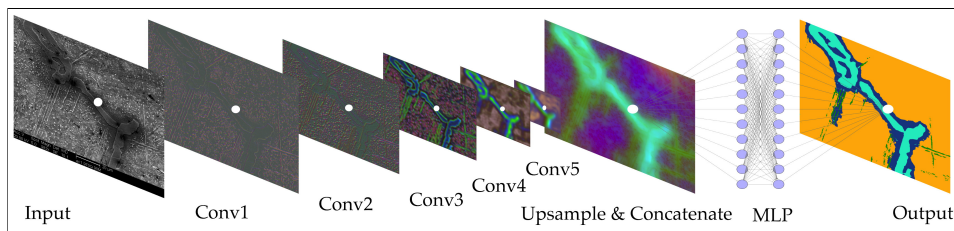
4

Figure 1: The PixelNet [13] image segmentation model. A pixel in the input image (left) is represented by the concatenation of its representations in each convolution layer (white dots). A multilayer perceptron (MLP) classifier is trained to associate the pixel representation with membership in a microstructure constituent (right).

2.1.2   Training details

We initialize the feature extraction portion of our networks with a pre-trained VGG-16 [24] network trained on the ImageNet [27] classification dataset. We train the pixel classification layers from scratch, randomly sampling initial weights from Gaussian distributions with zero mean and standard deviation $\sigma = \sqrt{2/c}$ [28], where c is the dimensionality of the input to the layer. To prevent overfitting, we use a combination of batch normalization [26], Dropout regularization [29], weight decay regularization [30] , and data augmentation. We set the weight decay strength to 0.0005 and apply Dropout regularization with a rate of 10% after the final MLP layer. Training images are subjected to local histogram equalization to mitigate differences in overall brightness across different samples and datasets. The training input and label images are augmented with random rotations in the range $[0, 2\pi)$, horizontal and vertical mirror symmetry, scaling in the range $[1, 2]$, and a $\pm 5\%$ random intensity shift. Rotated versions of the training input and label images are computed with mirror boundary conditions, with bilinear interpolation for the input images and nearest-neighbor interpolation for the (discrete) label images. We train the networks with the AdamW optimizer [30, 31] with the recommended default parameters. First we fix the parameters in the feature extraction portion of the network and train the pixel classification layers with an initial learning rate of 10^{-3} for 20 epochs (125 gradient updates). Each gradient update is computed from a random sample of 2048 pixels each from 4 augmented training images. We then fine-tune the entire CNN for 125 additional gradient updates using AdamW with an initial learning rate of 10^{-5}.

Our dataset has a heavy class imbalance (e.g. Widmanstätten cementite

5

only accounts for $\sim 3\%$ of pixels), so we compare a model trained using the standard cross-entropy classification loss with another trained using the focal loss [32], which is designed for unbalanced datasets. The focal loss extends the cross-entropy loss function $CrossEntropy(p_t) = -\log(p_t)$, where

$$p_t(p, y) = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{cases} \tag{1}$$

with ground truth y and predicted class probability $p = P(y = 1)$. The focal loss adds a modulating factor $(1 - p_t)^\gamma$ to emphasize examples about which the classifier is less confident during training, and a scaling parameter α to account for class imbalance:

$$FocalLoss(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{2}$$

We follow the recommendation of [32] in setting the focusing parameter $\gamma = 2$ and setting the class imbalance parameters α_t proportionally to the inverse frequency of each class.

2.2 Dataset

The semantic microstructure segmentation dataset consists of 24 manually annotated[3] micrographs from the open UHCS dataset [1, 2]; examples are shown in Figure 2 and in the online supplemental materials. These 645×484 pixel micrographs focus on the characteristic features of heat-treated UHCS: the proeutectoid cementite network and the associated denuded zone, and spheroidized and Widmanstätten cementite. Multiple heat treatment conditions and magnifications are represented in the semantic microstructure segmentation dataset.

The particle segmentation dataset consists of 24 micrographs collected at a single magnification in support of the particle coarsening analysis reported in [15]. Particle annotations were obtained through a partially-automated edge-based segmentation workflow [15]. A thresholded blur smooths contrast in the matrix surrounding particles before application of the Canny edge detector [34]. The particle outlines are filled in, and spurious edges (e.g. at grain boundaries) are removed by a 2px median filter. The final particle segmentations are verified and retouched manually where the contrast is insufficient for the Canny detector to identify particle edges. Particles intersecting the edge of the image are removed from the annotations to reduce bias in the estimated particle size distributions.

---

[3] We used the medical image annotation system MITK [33].

6

2.3 Performance evaluation

Because our set of annotated images is small (24 annotated micrographs total), we use cross-validation to estimate the generalization performance of the PixelNet architecture on our two microstructure segmentation tasks. We use a 6-fold cross-validation scheme [35]: each dataset is split into six validation sets of four micrographs each, and six PixelNet models are trained on each of the complementary training sets. The quantitative performance metrics reported in Tables 1 and 4 are averages over each validation image in the 6 validation sets; uncertainties are standard errors computed over the six validation images [35].

We report several standard evaluation metrics for semantic segmentation tasks: pixel accuracy (AC), precision, recall, and region intersection over union (IU) for individual microconstituents. For each of these metrics, a higher score indicates better performance.

Precision is the fraction of instances predicted to have class c that are correct:

$$Precision(c) = \frac{\sum_i \hat{y}_i = c \text{ and } y_i = c}{\sum_i \hat{y}_i = c} \tag{3}$$

where $\hat{y}_i$ indicates the predicted class label for each pixel i, and y_i indicates the corresponding ground truth class label. Equivalently, precision is the ratio of true positives to total (true and false) positives, which decreases when the model overpredicts the number of member pixels in a class.

Recall is the fraction of instances with ground truth class c that are predicted to have class c:

$$Recall(c) = \frac{\sum_i \hat{y}_i = c \text{ and } y_i = c}{\sum_i y_i = c} \tag{4}$$

Equivalently, recall is the ratio of true positives to the total number of pixels in a class, which decreases when the method underpredicts the member pixels in a class. Since the overall accuracy is defined as the number of true positives divided by the total number of pixels, it is straightforward to show that the classwise average recall or precision equals the overall accuracy.

The intersection over union metric $IU(c)$ for class c (also referred to as the Jaccard metric) is the ratio of correctly predicted pixels of class c (true positives) to the union of pixels with either ground truth or predicted class c (true and false positives plus false negatives):

$$IU(c) = \frac{\sum_i \hat{y}_i = c \text{ and } y_i = c}{\sum_i \hat{y}_i = c \text{ or } y_i = c} \tag{5}$$

7

For the spheroidite particle segmentation task, we also report performance metrics comparing particle size distributions (PSDs) obtained from the model predictions with those obtained from the ground truth annotations (as reported in [15]). We use the two-sample Kolmogorov-Smirnov (KS) test [36] to compare each pair of predicted and ground truth PSDs. The KS score reported in Table 4 is the fraction of micrographs where the KS test indicates that the predicted particle size distribution is consistent with the ground truth particle size distribution (i.e. the fraction of micrographs where we fail to reject (at the 95% confidence level) the null hypothesis that the distributions are equivalent).

2.4 Computing denuded zone widths

Given a microconstituent prediction map, we quantify the width of the denuded zone by computing the minimum distance to the network phase for each pixel on the matrix-particle interface. In practice, we compute a map of Euclidean distance to the network phase, and select the measurements at the denuded zone interface.

To obtain the denuded zone interface, we apply a series of image processing techniques to clean up the microconstituent prediction map, so that only the matrix predictions associated with the diffusion-limited denuded zone adjacent to the proeutectoid cementite network remain. A morphological filling operation removes any matrix pixels within the network. Matrix regions that are not connected to the network are identified by application of a morphological closing to matrix phase: any matrix segments that do not intersect the network phase after the morphological operation are removed. Finally, we remove any matrix predictions that are closer to a widmanstatten region than to a network region, and subsequently remove the widmanstatten regions. The region boundaries on the cleaned up label image (shown in Figure 5) include only the interface of the proeutectoid cementite network phase (indicated in blue) and the diffuse interface of the denuded zone (indicated in yellow).

3 Results and Discussion

3.1 Semantic microconstituent segmentation

Figure 2 shows microconstituent annotations (e-h) and predictions (i-p) for the four validation set micrographs (a-d) in one cross-validation iteration; results for all six validation sets are included in the online supplemental materials.

Microconstituent predictions using the focal loss function and the cross-entropy loss function are compared in Figure 2 (i-l) and (m-p) respectively. The predictions show reasonable correspondence with the annotations, despite nontrivial differences in features, such as particle size and appearance that arise from differences in heat treatment and magnification. Intensity variations and polishing damage evident in the input images have little impact on the predictive capability of the models. One notable exception is the cluster of spurious network predictions associated with the damaged areas in the lower left of Figure 2 c. Both models do a good job respecting the edges of the network carbide phase, with a few exceptions where the network is very fine or the contrast between network carbide and metal matrix is poor (see supplemental Figures S1.1 d and S1.5 d). Predicted boundaries between spheroidite particles and the denuded zone have little noise and tend to be smoother than in the annotations. The Widmanstätten predictions show the highest amount of noise, especially where the Widmanstätten lath are fine or are beginning to break up. The focal loss also tends to surround Widmanstätten cementite with wider swaths of the metallic matrix compared to the annotations. In addition to the low area fraction of Widmanstätten cementite, one potential contributing factor for these failure modes is labeling bias where the microstructure is ambiguous even to the human expert. For example, some areas with a low density of spheroidite particles are labeled by the model as metallic matrix where the annotation has made no such distinction. This phenomenon is evident in the lower half of Figure 2 i, where the model correctly identifies large patches of bare metal in the neighborhood of some large grain boundary cementite particles (refer to supplementary Figure S1.13 a for a more detail).

The cross-entropy model segmentation maps (Figure 2 (m-p)) tend to be more consistent with the annotations for the majority microconstituents. However, each model errs from the annotations in distinct ways. In general, the focal loss model seems to emphasize constituent contiguity, while the cross-entropy model tries to resolve fine features. For example, compared with the annotations in Figure 2 f, the focal loss model (Figure 2 j) more liberally identifies the very fine lath structures in the bottom right corner of the frame, consolidating them into a single patch, while the cross-entropy model (Figure 2 n) produces a noisier map that attempts to track finer-grained details of the lath structure but loses some area fraction. One conclusion from these results is that while both models give acceptable results, neither is necessarily the best that can be achieved. For a given segmentation problem, the user must select model parameters and loss functions to achieve the desired quantitative and qualitative performance.
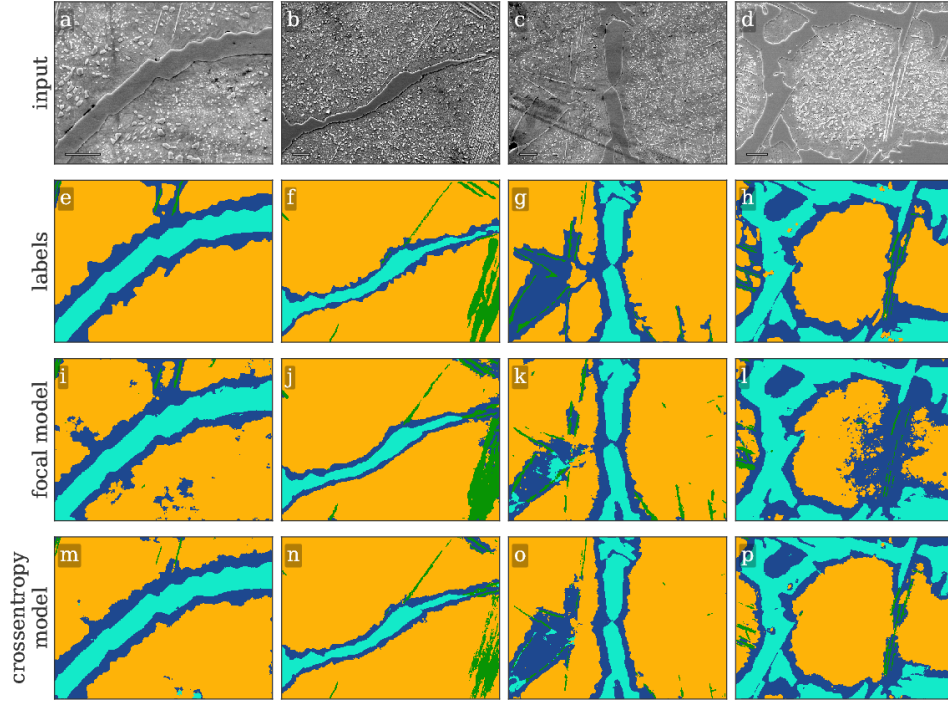
Figure 2: (a-d) Validation set micrographs, (e-h) microconstituent annotations, (i-l) PixelNet predictions using the focal loss segmentation model, and (m-p) PixelNet predictions using the standard crossentropy classification loss. Microstructural constituents include proeutectoid grain boundary cementite (light blue), ferritic matrix (dark blue), spheroidite particles (yellow), and Widmanstätten cementite (green). Scale bars indicate $10\mu m$.

Table 1: Focal model semantic segmentation performance averaged over validation images. Uncertainties are standard errors calculated across validation images.

	IU	precision	recall
matrix	49.1 ± 3.4	60.3 ± 4.4	72.3 ± 3.7
network	72.9 ± 5.3	85.5 ± 4.0	80.7 ± 5.9
spheroidite	85.7 ± 1.8	95.1 ± 1.2	89.8 ± 1.7
widmanstatten	42.7 ± 2.9	50.2 ± 3.6	73.5 ± 3.9
overall	62.6 ± 2.5	86.5 ± 1.6	86.5 ± 1.6

Table 1 shows the average validation set performance with standard errors for the focal loss model. The model obtains 86.5 ± 1.6 % overall accuracy (AC, equivalent to the average of the classwise recall or precision) in reproducing the pixel-level annotations. The model is consistently good at identifying spheroidite and network regions. The less prevalent microconstituents (matrix and Widmanstätten) are not as well captured, and show higher variation between images. For these microconstituents, the recall score is better than the precision score, meaning that the CNN tends to mistake other classes for matrix and Widmanstätten more than it tends to miss genuine matrix and Widmanstätten pixels. This effect is demonstrated on the fine Widmanstätten lath in the lower right portion of Figure 2 j, where the model fills in the fine spacing between Widmanstätten lath in its prediction. The low proportion of Widmanstätten pixels in the dataset enhances this effect. In the case of the matrix class, the difference in recall and precision scores is partly due to the overprediction of metallic matrix in areas containing a low density of spheroidite particles, as discussed in reference to Figure 2 i.

In contrast, the spheroidite and network classes have slightly higher precision compared with their recall scores. The standard error for the network scores is large, and is therefore likely accounted for by the small number of gross errors discussed in supplemental Figures S1.1 d and S1.5 d. Finally, the small difference in precision and recall score for the spheroidite class is likely also due to the overprediction of the metal matrix in regions with low particle density.

Table 2 shows the same performance metrics for the cross-entropy model trained with revised training hyperparameters. The crossentropy shows clearly superior overall numerical performance, including a nearly ten point bump in overall IU score. While most of the per-microconstituent scores are higher for the crossentropy model, the recall score for Widmanstätten cementite is

Table 2: Crossentropy model semantic segmentation performance averaged over validation images. Uncertainties are standard errors calculated across validation images.

	IU	precision	recall
matrix	67.3 ± 6.2	81.8 ± 8.0	79.4 ± 4.0
network	89.4 ± 3.6	96.1 ± 0.6	92.8 ± 3.5
spheroidite	91.9 ± 3.5	94.9 ± 2.6	96.6 ± 1.7
widmanstatten	52.9 ± 6.4	72.2 ± 7.1	66.8 ± 8.1
overall	75.4 ± 3.7	92.6 ± 2.5	92.6 ± 2.5

consistently depressed due to underprediction. We also briefly experimented with the popular U-Net [22] architecture, but found that this architecture slightly underperformed compared with Pixelnet. These results indicate that various CNN architectures and training schemes can achieve reasonable results, so the user can select an approach based on desired outcomes. Because of its excellent performance in the spheroidite segmentation task (reported in the next section), we present results only from the Pixelnet model trained with the focal loss function throughout the rest of this manuscript.

These quantitative metrics are useful for interpreting the strengths and weaknesses of a particular CNN model, but they do not necessarily directly quantify the quality of the predicted segmentation maps due to inherent subjectivity and bias in the labeling process. Even a single human annotator will not be able to consistently label an entire dataset, especially for ambiguous higher-level microconstituents such as the spheroidite class. For example, the annotator must decide how closely to track cementite particles when tracing out the edge of the denuded zone. In some cases, it is unclear whether a carbide should be labeled as grain boundary cementite or as a piece of Widmanstätten lath.

Furthermore, the low resolution of the input images relative to some of the finer features of interest also places a practical upper bound on these numerical performance scores, especially for microconstituents with large interfacial areas like the Widmanstätten lath. Many of the Widmanstätten lath in this dataset are just a few pixels wide, which can lead large shifts in numerical scores for what a human might consider a minor difference in labeling (e.g. dilating or eroding the Widmanstätten lath by one pixel).
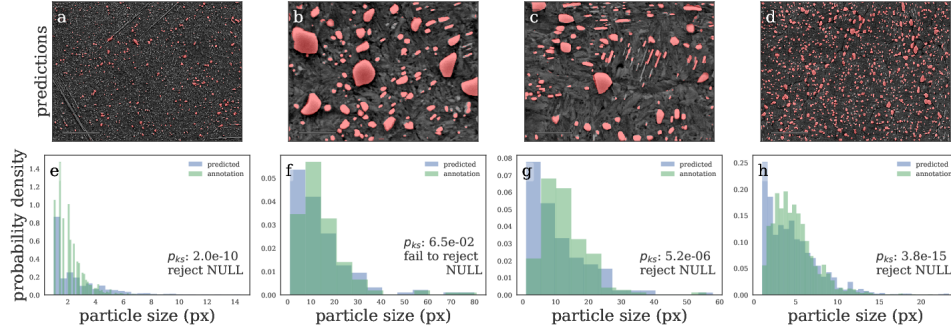
Figure 3: (a-d) Validation set predictions for the spheroidite particle segmentation task, along with (e-h) corresponding derived particle size distributions for the particle predictions (blue) and annotations (green). Scale bars indicate $5\mu m$.

3.2 Spheroidite particle segmentation

Figure 3 shows some validation results for the individual particle segmentation task, with numerical performance reported in Tables 3 and 4; additional examples are included in the online supplemental materials. Particle predictions are overlaid in red on the input micrographs (a-d). The second row (e-h) shows the empirical particle size distributions for both particle predictions and annotations, as well as the results of the two-sample Kolmogorov-Smirnov hypothesis test for distribution equivalence. Predictions for larger particles relative to the image frame (Figures 3 b and c) are consistently good, even where contrast gradients across particles and non-trivial background structure challenge thresholding and edge-based segmentation methods. The primary failure mode of the particle segmentation model is underprediction of very small particles, particularly in Figure 3 a and d. The vast majority of the fine particles in Figure 3 are missing entirely, and many are only partially labeled by the CNN with just one or two foreground pixels. These particles are typically one to five pixels in size, suggesting that higher- or multi-resolution inputs are necessary for general microstructure segmentation CNNs. However, the CNN does avoid spuriously labeling the small segments of Widmanstätten in Figure 3 as particles.

The PixelNet model performs slightly better than Otsu's thresholding method [37] on all metrics. One source of bias in these performance measurements are missing particles in the annotations, either from the removal of particles intersecting the image border, or from failure of the semi-automated annotation method itself. An additional source of bias stems from the ap-

13

Table 3: Particle segmentation performance averaged over validation images. Uncertainties are standard errors calculated across validation images.

	IU	precision	recall
matrix	90.0 ± 1.0	95.0 ± 0.6	94.5 ± 1.1
spheroidite	54.8 ± 3.4	74.6 ± 2.8	70.3 ± 4.3
overall	72.4 ± 3.1	91.1 ± 0.9	91.1 ± 0.9

Table 4: Particle segmentation performance metrics. Uncertainties are standard errors calculated across validation images.

model	$\text{IU}_{\text{matrix}}$	$\text{IU}_{\text{spheroidite}}$	IU_{avg}	AC	PSD KS
otsu	86.2 ± 7.2	53.7 ± 12.1	69.9 ± 9.3	88.1 ± 6.1	-
pixelnet	90.0 ± 1.0	54.8 ± 3.4	72.4 ± 3.1	91.1 ± 0.9	0.042

plication of the watershed algorithm [38] to split conjoined particles in the annotations; watershed segmentation is not presently applied to the particle predictions, increasing the relative rate of larger particles.

Despite good numerical performance on the particle segmentation task, the KS test suggests we reject the null hypothesis that the predicted and ground truth particle size distributions are equivalent for all but one of the 24 validation micrographs (shown in 3 b). The difficulty in detecting small particles explains the discrepancies between empirical particle size distributions that contribute to the KS score. For the two validation micrographs in Figure 3 containing fine particles, the particle size histograms and prediction maps show that the model often entirely misses particles with radii smaller than 5px. Many of these missing ~5px particles are partially labeled in the CNN predictions, leading to a severe overrepresentation of single-pixel particles, especially in Figure 3 h.

3.3 Quantitative analysis of higher-order features

High-quality automated segmentation techniques for complex microstructure constituents expand the scope of conventional quantitative microstructure analysis by reducing the manual labor required to obtain statistically meaningful amounts of data. In our UHCS case study, the CNN segmentation model allows us to collect volume and shape statistics for the proeutectoid carbide network, spheroidite particles, and Widmanstätten lath directly from SEM micrographs with no manual intervention. Additionally, the microconstituent prediction maps enable automated acquisition of interesting microstructural
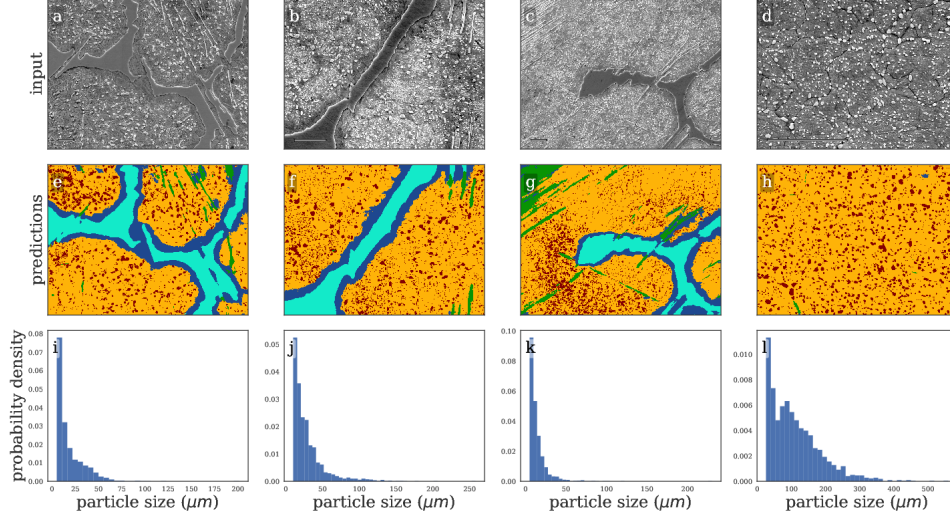
14

Figure 4: (a-d) Micrographs with (e-h) validation set microconstituent predictions and (i-l) derived particle size distributions obtained by applying the particle segmentation CNN to the semantic microstructure segmentation dataset. Scale bars indicate $10\mu m$.

statistics that were previously intractable, such as particle size distributions conditioned on spatial relationships with other microstructure features, or denuded zone widths [15].

Combining the two microstructure segmentation models allows us to filter out irrelevant microstructure features in order to estimate particle size distributions. Figure 4 shows combined microstructure predictions from both the abstract microstructure model and the particle model, using the same color scheme as Figures 2 and 3. We run the input image through separately-trained particle segmentation CNN and microconstituent CNN, suppressing particle predictions (red) outside of the predicted spheroidite regions (yellow). With an appropriate number of images, one could also compute particle size distributions spatially conditioned on other microstructure features (e.g. distance from the network phase), which could help lead to insights into operative microstructure evolution mechanisms (particle coarsening vs precipitation). The resolution of these input micrographs is insufficient to yield quantitatively accurate particle size distributions, especially with the underprediction of small particles discussed in Section 3.2, as evident in Figures 4 b and c. However, higher quality input and training micrographs will mitigate this effect.
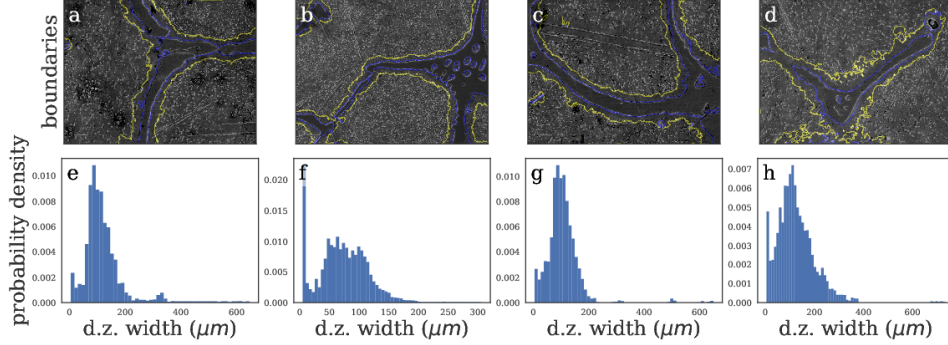
15

Figure 5: (a-d) Validation set microconstituent predictions with (e-h) corresponding denuded zone width distributions. The network interface is shown in blue and the particle matrix interface is shown in yellow. Scale bars indicate $10\mu m$

Figure 5 shows the predicted network and denuded zone boundaries for four validation images with corresponding computed denuded zone width distributions. The denuded zone width distributions are calculated by aggregating the minimum distance to the network interface for each pixel on the denuded zone boundary, as described in detail in Section 2.4. Generally, these empirical denuded zone widths are reasonable, but some care is required to interpret them. Specifically, the denuded zone width distributions in Figures 5 b and d have high frequencies at small spacings that result from spurious cementite network predictions. Figures 5 a and d also exhibit some overprediction of the denuded zone width where the particles are very fine, particularly in the upper portion of Figure 5 a.

The initial investment of micrograph annotation and training a CNN makes sense where a statistical number of samples must be characterized in the context of alloy and processing optimization studies, and in the context of microstructure and process validation or verification. Microconstituent annotation accounted for a substantial portion of the time we spent on this project. After an initial learning curve, a typical micrograph in our dataset cost between 20 and 30 minutes to annotate. In contrast, a commodity GPU performs Pixelnet-based segmentation at a rate of approximately one micrograph per second, after an initial training period of two to four hours depending on training hyperparameters. Success in a practical microstructure science setting will depend on establishing higher-quality training data and deeper understanding of the biases and variance of the labeling process.

The CNN predictions provide some useful feedback on these subjective

labeling decisions: consider the micrograph, annotation, and predictions in supplemental Figure S1.6 a, e, and i. In the bottom half of this micrograph (and in the other micrographs in this validation set), the annotator neglected to label the metal matrix surrounding the Widmanstätten lath as such, while the CNN consistently includes some matrix predictions associated with Widmanstätten predictions. This subjective labeling decision can be mitigated with higher-fidelity labeling of individual carbide particles – at much greater labeling expense. A high quality dataset might be obtained via crowd-sourcing (e.g. students in a microstructure analytics course), generation of realistic synthetic datasets through e.g. phase field modeling, or through the substantial expense of high-resolution elemental mapping with SEM+EDS (Energy-dispersive spectrometry). A large dataset might also be collected in a semi-supervised fashion through the development of smart microscopes with integrated microstructure recognition features.

Furthermore, it is critical to benchmark microstructure-specific tasks against other popular CNN architectures for semantic segmentation. Our approach of directly transferring the particle prediction CNN is tenuous, especially due to the disparity in magnification between the general UHCS and specific particle segmentation datasets. Rather than training two separate CNNs, it may be more appropriate train a single CNN in a multi-task setting, so that microstructures are mapped to a common numerical representation before the respective microconstituent and particle classification tasks.

Finally, microstructure data science is extremely data-limited in comparison to most general computer vision tasks. Though outside the scope of the present report, a detailed follow-on study to fully characterize the training data requirements of deep learning based microstructure segmentation models would be a valuable tool to enable experimental planning before significant investment for industrial application. In parallel, collaboration with computer scientists working on low-data deep learning, semi-supervised, and unsupervised techniques could also open the door to applicability in many more microstructure systems, especially where pixel-level annotations are expensive or difficult to consistently obtain.

4   Conclusions

We demonstrate microstructural segmentation and quantitative analysis at a high level of abstraction by applying an off-the-shelf deep neural network architecture for pixel-wise prediction tasks. We also present two new open microstructure segmentation benchmark datasets featuring the microstruc-

tures in ultra-high carbon steel at different length scales. This data-driven approach to microstructure segmentation expands the reach of traditional quantitative microstructure characterization to more complex industrially-relevant microstructure features that have, until now been, difficult to treat in an automated fashion. Combined with emerging automated microscopy capabilities, data-driven microstructure segmentation systems will enable future applications in high-throughput microstructure studies, including investigations of structure/processing relationships, microstructure design and optimization, and microstructure-based material qualification.

Acknowledgments

References

[1]   Brian L. DeCost et al. "UHCSDB (UltraHigh Carbon Steel micrograph DataBase): tools for exploring large heterogeneous microstructure datasets". In: *Integrating Materials and Manufacturing Innovation* 6 (2 2017), pp. 197–205. DOI: 10.1007/s40192-017-0097-0. URL: http://uhcsdb.materials.cmu.edu (cit. on pp. 1, 2, 6, 18).

[2]   Matthew D. Hecht et al. *Ultrahigh Carbon Steel Micrographs.* https://hdl.handle.net/11256/940. 2017 (cit. on pp. 1, 6).

[3]   Brian L. DeCost et al. *Ultrahigh Carbon Steel Micrographs: Microconstituent annotations.* https://hdl.handle.net/11256/964. 2018 (cit. on pp. 1, 3).

[4]   EO Hall. "The deformation and ageing of mild steel: III discussion of results". In: *Proceedings of the Physical Society. Section B* 64.9 (1951), p. 747 (cit. on p. 2).

[5]   C Zener. "quoted by CS Smith". In: *Trans. AIME* 175 (1948). quoted by CS Smith, p. 15 (cit. on p. 2).

[6]   Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: http://dx.doi.org/10.1038/nature14539 (cit. on pp. 2, 3).

[7]   Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016 (cit. on pp. 2, 3).

[8]   Brian L. DeCost and Elizabeth A. Holm. "A computer vision approach for automated analysis and classification of microstructural image data". In: *Computational Materials Science* 110 (2015), pp. 126–133. DOI: 10.1016/j.commatsci.2015.08.011 (cit. on p. 2).

[9]   Aritra Chowdhury et al. "Image driven machine learning methods for microstructure recognition". In: *Computational Materials Science* 123 (2016), pp. 176–187 (cit. on p. 2).

[10]   Nicholas Lubbers, Turab Lookman, and Kipton Barros. "Inferring Low-Dimensional Microstructure Representations Using Convolutional Neural Networks". In: *CoRR* (2016). arXiv: 1611.02764 [physics.comp-ph]. URL: http://arxiv.org/abs/1611.02764v1 (cit. on p. 2).

[11]   Brian L. DeCost, Toby Francis, and Elizabeth A. Holm. "Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures". In: *Acta Materialia* 133 (2017), pp. 30–40. DOI: 10.1016/j.actamat.2017.05.014. URL: http://arxiv.org/abs/1702.01117 (cit. on p. 2).

[12]   Seyed Majid Azimi et al. "Advanced Steel Microstructural Classification by Deep Learning Methods". In: *Scientific Reports* 8.1 (2018). ISSN: 2045-2322. DOI: 10.1038/s41598-018-20037-5. URL: http://dx.doi.org/10.1038/s41598-018-20037-5 (cit. on p. 2).

[13]   Aayush Bansal et al. "Pixelnet: Representation of the Pixels, By the Pixels, and for the Pixels". In: *CoRR* (2017). arXiv: 1702.06506 [cs.CV]. URL: http://arxiv.org/abs/1702.06506v1 (cit. on pp. 2, 4, 5).

[14]   Matthew D Hecht, Bryan A Webler, and Yoosuf N Picard. "Digital image analysis to quantify carbide networks in ultrahigh carbon steels". In: *Materials Characterization* 117 (2016), pp. 134–143 (cit. on p. 2).

[15]   Matthew D Hecht, Yoosuf N Picard, and Bryan A Webler. "Coarsening of inter and intragranular proeutectoid cementite in an initially pearlitic 2C-4Cr ultrahigh carbon steel". In: *Metallurgical and Materials Transactions A* (2017), pp. 1–16. ISSN: 1543-1940. DOI: 10.1007/s11661-017-4012-2 (cit. on pp. 2, 6, 8, 15).

[16]   Panqu Wang et al. "Understanding Convolution for Semantic Segmentation". In: *CoRR* (2017). arXiv: 1702.08502 [cs.CV]. URL: http://arxiv.org/abs/1702.08502v1 (cit. on p. 3).

[17]   V. Badrinarayanan, A. Kendall, and R. Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP.99 (2017), pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2644615 (cit. on pp. 3, 4).

[18]   A. Bansal, B. Russell, and A. Gupta. "Marr Revisited: 2D-3D Alignment via Surface Normal Prediction". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5965–5974. DOI: 10.1109/CVPR.2016.642 (cit. on p. 3).

[19]   Bharath Hariharan et al. "Hypercolumns for Object Segmentation and Fine-Grained Localization". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on p. 4).

[20]   Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). DOI: 10.1109/cvpr.2015.7298965. URL: http://dx.doi.org/10.1109/cvpr.2015.7298965 (cit. on p. 4).

[21]   Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. "Bayesian Segnet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: *CoRR* (2015). arXiv: 1511.02680 [cs.CV]. URL: http://arxiv.org/abs/1511.02680v2 (cit. on p. 4).

[22]   Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (2015), pp. 234–241. ISSN: 1611-3349. DOI: 10.1007/978-3-319-24574-4_28. URL: http://dx.doi.org/10.1007/978-3-319-24574-4_28 (cit. on pp. 4, 12).

[23] Simon Jégou et al. "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation". In: *CoRR* (2016). arXiv: 1611.09326 [cs.CV]. URL: http://arxiv.org/abs/1611.09326v2 (cit. on p. 4).

[24] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556 (cit. on pp. 4, 5).

[25] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, 2010, pp. 807–814. ISBN: 978-1-60558-907-7. URL: http://dl.acm.org/citation.cfm?id=3104322.3104425 (cit. on p. 4).

[26] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 448–456. URL: http://proceedings.mlr.press/v37/ioffe15.html (cit. on pp. 4, 5).

[27] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y. URL: http://dx.doi.org/10.1007/s11263-015-0816-y (cit. on p. 5).

[28] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015). DOI: 10.1109/iccv.2015.123. URL: http://dx.doi.org/10.1109/iccv.2015.123 (cit. on p. 5).

[29] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958. URL: http://dl.acm.org/citation.cfm?id=2670313 (cit. on p. 5).

[30] Ilya Loshchilov and Frank Hutter. "Fixing Weight Decay Regularization in Adam". In: *CoRR* (2017). arXiv: 1711.05101 [cs.LG]. URL: http://arxiv.org/abs/1711.05101v2 (cit. on p. 5).

[31] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* (2014). arXiv: 1412.6980 [cs.LG]. URL: http://arxiv.org/abs/1412.6980v9 (cit. on p. 5).

[32] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *CoRR* (2017). arXiv: 1708.02002 [cs.CV]. URL: http://arxiv.org/abs/1708.02002v2 (cit. on p. 6).

[33] Marco Nolden et al. "The Medical Imaging Interaction Toolkit: Challenges and Advances". In: *International Journal of Computer Assisted Radiology and Surgery* 8.4 (2013), pp. 607–620. DOI: 10.1007/s11548-013-0840-8. URL: https://doi.org/10.1007/s11548-013-0840-8 (cit. on pp. 6, 18).

[34] JOHN CANNY. "A Computational Approach to Edge Detection". In: *Readings in Computer Vision* (1987), pp. 184–203. DOI: 10.1016/b978-0-08-051581-6.50024-6. URL: http://dx.doi.org/10.1016/b978-0-08-051581-6.50024-6 (cit. on p. 6).

[35] Trevor Hastie, Jerome Friedman, and Robert Tibshirani. "The Elements of Statistical Learning". In: *Springer Series in Statistics* (2001). ISSN: 0172-7397. DOI: 10.1007/978-0-387-21606-5. URL: http://dx.doi.org/10.1007/978-0-387-21606-5 (cit. on p. 7).

[36] Frank J. Massey. "The Kolmogorov-Smirnov Test for Goodness of Fit". In: *Journal of the American Statistical Association* 46.253 (1951), pp. 68–78. ISSN: 1537-274X. DOI: 10.1080/01621459.1951.10500769. URL: http://dx.doi.org/10.1080/01621459.1951.10500769 (cit. on p. 8).

[37] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66. ISSN: 2168-2909. DOI: 10.1109/tsmc.1979.4310076. URL: http://dx.doi.org/10.1109/tsmc.1979.4310076 (cit. on p. 13).

[38] L. Vincent and P. Soille. "Watersheds in digital spaces: an efficient algorithm based on immersion simulations". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.6 (1991), pp. 583–598. ISSN: 0162-8828. DOI: 10.1109/34.87344. URL: http://dx.doi.org/10.1109/34.87344 (cit. on p. 14).

[39] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *The Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 18).

[40] Stéfan van der Walt et al. "Scikit-Image: Image Processing in Python".
      In: *PeerJ* 2.nil (2014), e453. DOI: 10.7717/peerj.453. URL: https:
      //doi.org/10.7717/peerj.453 (cit. on p. 18).

[41] François Chollet. *Keras*. https://github.com/fchollet/keras. 2015
      (cit. on p. 18).