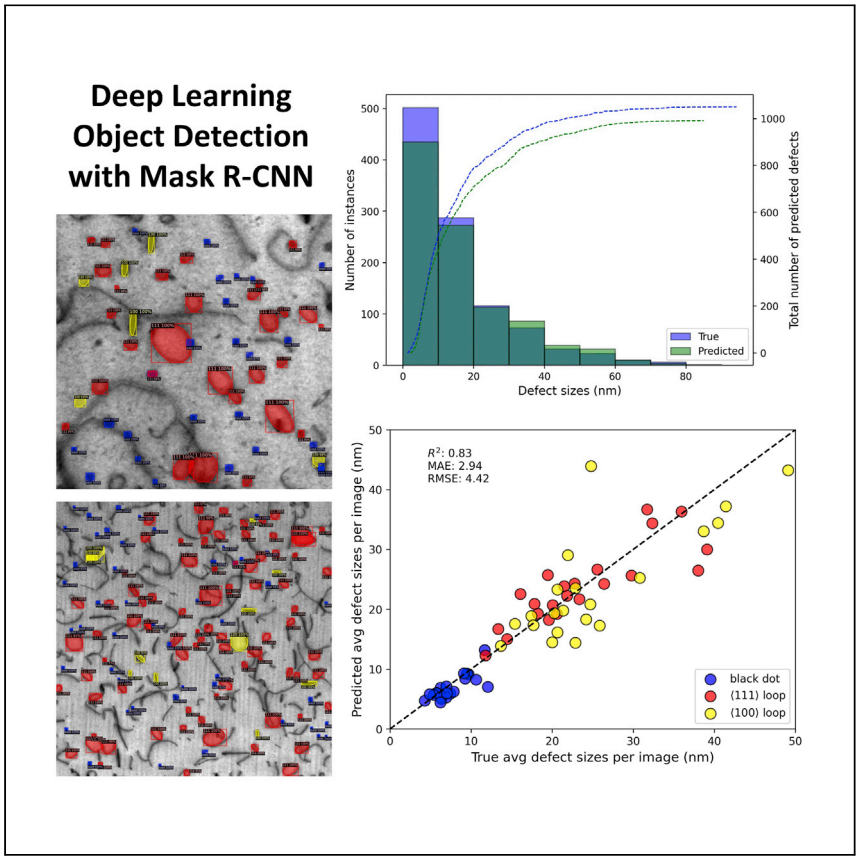
Cell Reports
Physical Science

CellPress
OPEN ACCESS

Article

Performance and limitations of deep learning semantic segmentation of multiple defects in transmission electron micrographs



Jacobs et al. use deep learning to detect and quantify defects in electron microscopy images of irradiated FeCrAl alloys. The model accurately predicts alloy hardening, a key property of nuclear reactor materials. Implications of model performance based on training data size and domain are discussed.

Ryan Jacobs, Mingren Shen, Yuhan Liu, ..., Chao Wang, Kevin G. Field, Dane Morgan

rjacobs3@wisc.edu

Highlights

Semantic segmentation of multiple defects in electron microscopy images of FeCrAl

In-depth analysis of defect sizes, shapes, areal densities, and material hardening

Model weak points investigated with targeted cross-validation tests

Model yields accurate hardening predictions, important for nuclear materials

Jacobs et al., Cell Reports Physical Science 3, 100876
May 18, 2022
https://doi.org/10.1016/j.xcrp.2022.100876

Cell Reports
Physical Science

CellPress
OPEN ACCESS

Article

Performance and limitations of deep learning semantic segmentation of multiple defects in transmission electron micrographs

Ryan Jacobs,[1,5,6,*] Mingren Shen,[1,5] Yuhan Liu,[2,5] Wei Hao,[2] Xiaoshan Li,[2] Ruoyu He,[2] Jacob R.C. Greaves,[1] Donglin Wang,[2] Zeming Xie,[2] Zitong Huang,[3] Chao Wang,[2] Kevin G. Field,[4] and Dane Morgan[1]

SUMMARY

Transmission electron microscopy (TEM) is a popular method for characterizing and quantifying defects in materials. Analyzing digitized TEM images is typically done manually, which is a time-consuming and potentially error-prone task that is not scalable to large dataset sizes, motivating development of automated methods for quantifying and analyzing defects in TEM images. In this work, we perform semantic segmentation of multiple defect types in electron microscopy images of irradiated FeCrAl alloys using a deep learning mask regional convolutional neural network (Mask R-CNN) model. We evaluate the performance of the model based on distributions of defect shapes, sizes, and areal densities relevant to informing physical modeling and understanding irradiated Fe-based materials properties. To better understand the performance and present limitations of the model, we provide examples of useful evaluation tests, which include a suite of random splits and dataset-size-dependent and domain-targeted cross-validation tests, exposing potential weak points in the model applicability domain. Our model predicts the expected irradiation-induced material hardening to within 10–20 MPa (about 10% of total hardening), on par with experimental error. Finally, we discuss the first phase of an effort to provide an easy-to-use, open-source object detection tool to the broader community for identifying defects in new images.

INTRODUCTION

Extended defects in materials are critical for determining their properties and performance. The role of defects is particularly important for materials performance in extreme environments, where a cornerstone of advanced materials discovery and development is understanding the production and evolution of defects. In many cases, extreme environments include elevated temperatures, stress, corrosion rates, and radiation, which can lead to production of defects, including point defects, line dislocations, dislocation loops, cavities/voids, stacking fault tetrahedra, and precipitates, to name a few. The nucleation, growth, and evolution of these various defect types can lead to deleterious changes in materials performance, including loss of strength and ductility. Common, simplified structure-property relationships, such as the dispersed barrier hardening model,[1] show that these changes in properties are directly related to the size, number, density, and type of defects present. As a

[1]Department of Materials Science and Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

[2]Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI 53706, USA

[3]Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

[4]Nuclear Engineering and Radiological Sciences, University of Michigan-Ann Arbor, Ann Arbor, MI 48109, USA

[5]These authors contributed equally

[6]Lead contact

*Correspondence: rjacobs3@wisc.edu

https://doi.org/10.1016/j.xcrp.2022.100876

result, a significant portion of the materials discovery, development, and deployment cycle for extreme environments is spent characterizing and quantifying these defects after simulated exposure. Characterization and quantification of defects are critical for predicting and understanding materials performance in an array of complex and aggressive environments.

Transmission electron microscopy (TEM) is a popular method for characterizing and quantifying defects in materials. Analyzing digitized TEM images is commonly done with software packages like ImageJ,[2] which enable a user to manually quantify the size, shape, and location of defects in the images. This purely manual, human-based task is very time-consuming, error-prone, and inconsistent. It generally requires many hours of training and expertise to do well and is not scalable to large dataset sizes. The latter point is particularly important, considering that modern TEM instruments can now routinely collect tens of thousands of images or hours of video content whose manual analysis is not feasible.[3] Therefore, development of automated methods for quantifying and analyzing defects in TEM images as well as understanding the advantages, shortcomings, and potential pitfalls of these methods can be used to establish a set of best practices for the community as these automated methods witness increased adoption.

The rise in popularity of deep learning methods in 2012 revolutionized the field of computer vision,[4,5] and the maturation of these methods has direct implications for the present problem of automatically characterizing and quantifying defects in TEM images. Deep learning techniques typically involve use of convolutional neural networks (CNNs) and have enabled stunning advances ranging from superhuman facial recognition to self-driving vehicles. As a prime example, yearly object classification competitions such as the Pattern Analysis, Statistical Modeling, and Computational Learning Visual Object Classes (PASCAL VOC)[6] and ImageNet[7] Large-Scale Visual Recognition Challenge (ILSVRC)[8] witnessed a significant advance in prediction accuracy after 2012, when the first deep-learning-based image classification network, AlexNet,[5] enabled a performance increase from about 40% correct in the prior 2 years to nearly 60% correct in the PASCAL VOC challenge.[9] In the following few years, the advances in deep learning object classification methods made these models so adept at classifying the test set images at these competitions that, as of 2018, the average classification performance was at or above 90% for PASCAL VOC and greater than 80% for ILSVRC.[9]

Coupled use of traditional computer vision and machine learning methods, such as a workflow incorporating a sequence of blurring, thresholding, and masking operations combined with clustering algorithms or random forest classification models, has yielded numerous successes in analyzing and quantifying an assortment of features in microscopy images.[10–12] However, traditional computer vision methods tend to suffer from reliance on empirically chosen parameters, making them useful for limited sets of cases and, thus, less general and less transferable than deep-learning-based methods. Deep learning methods are increasingly being adopted in materials science.[13–17] In microstructure characterization in materials science,[18–20] the advances of these deep learning methods have enabled a shift from combined use of manually implemented and tuned traditional computer vision and machine learning techniques to more automatic deep learning methods. Deep learning methods have shown success in tasks ranging from highlighting defective regions of crystalline materials in high-resolution scanning TEM (STEM) images,[21] segmenting different microstructural phases,[22] finding locations of individual atoms in a material,[23] counting and analyzing nanoparticles,[24,25] identifying and classifying

surface defect types in steel,[26,27] and classifying types of dislocation loops at the microscale.[28,29]

Over the past few years, there have been a handful of pioneering studies employing deep learning methods to characterize and quantify defects in electron microscopy images. Li et al.[28] used a standard CNN architecture coupled with traditional computer vision methods to quantify defects in FeCrAl alloys. Two key limitations of their work were the ability to only identify a single type of defect and the lack of pixel-level segmentation information from the model, prompting use of traditional computer vision methods that required extensive manual tuning to obtain the desired performance. Shen et al.[29] extended the work of Li et al.[28] by using the Faster R-CNN (regional CNN) algorithm on the data from Li et al.[28] and were able to characterize multiple defect types with a full deep learning approach. However, this work still used traditional computer vision methods to extract details of predicted defect size.[29] In a similar vein, the work of Anderson et al.[30] also used the Faster R-CNN algorithm to detect He bubbles, which are sometimes called cavities or voids, in irradiated Ni-based alloys. Like the studies of Li et al.[28] and Shen et al.,[29] this study also used additional post-processing methods separate from the deep learning model to extract materials property information such as void sizes because the Faster R-CNN model does not provide pixel-level segmentation information.[30] In addition, Shen et al.[29] also employed the YOLO (you only look once) object detection model to demonstrate real-time identification and tracking of defect loops in FeCrAl alloys for sets of TEM images extracted from a video.[31] As a final example, Roberts et al.[32] employed a model called DefectSegNet, based on the U-net model architecture, as the first study to demonstrate pixel-level segmentation of multiple defect types in electron microscopy images. This work, although very encouraging, does not conclusively demonstrate the widespread effectiveness of pixel-wise segmentation models for two reasons. First, images were gathered for only a single material alloy and single sample, and two large 2,048 × 2,048 images were used for each defect type, which, after augmentation, amounted to 48 individual smaller training images, likely indicating a narrow model domain and a small amount of training data. Second, the output of U-net models consists of a single mask for the entire image, denoting whether individual pixels are part of a defect or part of the background, making quantification of per-defect statistics, such as size, shape, and density, more difficult, necessitating use of additional techniques beyond the deep learning approach used for detection.[32]

In this study, we employ pixel-level segmentation models to create an automated, fully deep-learning-based approach to classify and analyze multiple defect types in irradiated FeCrAl alloys (an example micrograph is shown in Figure S1). We highlight analysis of key model performance statistics with a focus on quantities such as predicted distributions of defect shapes, defect sizes, and defect areal densities relevant to informing modeling and understanding irradiated alloy materials properties. In addition, to better understand the performance and present limitations of the model, we provide examples of useful evaluation tests, which include a suite of random splits, and dataset size-dependent and domain-targeted cross-validation tests. Finally, a significant expansion of the labeling in the image database from the studies of Li et al.[28] and Shen et al.,[29] including more labeled images and pixel-level segmentation, enables us to make a current best-fit segmentation model for identifying defect loops in irradiated FeCrAl alloys, which can be used by other researchers to make predictions of defects in new images. We provide the final model fit of all images in the latest database and a Google Colab notebook to allow users to easily make predictions on new test images. This automated analysis

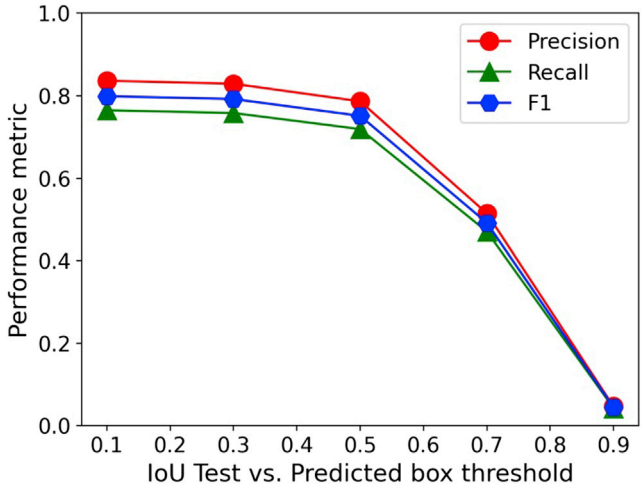


Figure 1. Summary of model classification performance

Model performance as a function of IoU cutoff between predicted and ground truth. The model was fit and evaluated using dataset 1 "initial split."

provides an output of numbers and locations of each defect and the test images with the predictions overlayed (see Data and code availability under Experimental procedures).

RESULTS

Assessing model performance on a single dataset

In this section, we assess the mask regional convolutional neural network (Mask R-CNN) model performance using the best set of hyperparameters obtained from a preliminary survey of roughly 25 Mask R-CNN model runs (Note S1). All fits in this section are performed on a single dataset, dataset 1 "initial split." An overview of the database, including nomenclature for the different data splits assessed in this section, and methods used are provided under Experimental procedures. Figure 1 provides a graphical representation of the calculated precision, recall, and F1 score for the test of finding defects (regardless of whether the type is correct) as a function of this intersection-over-union (IoU) cutoff. We found that an IoU value of 0.3 provides a reliable balance of model performance for this defect find test while providing reliable predictions of defect sizes, shapes, and densities (to be discussed later). In Figure 1, the Mask R-CNN overall F1 score at an IoU value of 0.3 is about 0.8, which is nearly identical to the value obtained by Shen et al.,[29] who used the Faster R-CNN model as implemented in the ChainerCV package.[33] This result indicates that the Mask R-CNN model used in this work can provide defect find statistics at the same level of quality as Faster R-CNN and that the use of Detectron2 versus ChainerCV and different backbone structure (ResNet 50 here, VGG16 in Shen et al.[29]) does not appreciably alter the model quality, at least for this case. Figure 2 provides three sets of images comparing the ground truth labels with the Mask R-CNN model predictions. Similar to what was observed by Shen et al.,[29] from manual inspection, the object detection model does well overall at correctly categorizing and placing defect locations on the image relative to the ground truth. There are some observable errors in the prediction versus the ground truth, such as missing some defects that should be present (false negative), predicting some defects to be present that should not be (false positive), and miscategorizing some defects. These types of errors are to be expected, and more details regarding their discussion and quantification are provided in Shen et al.[29]

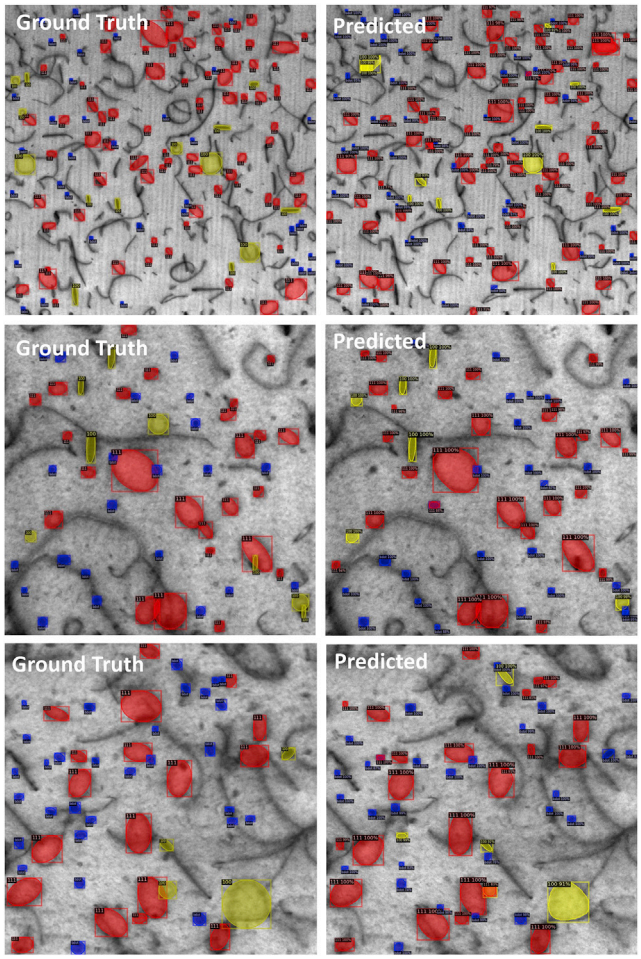


Figure 2. Examples of images with true and predicted labels

Examples of labeled ground truth (left columns) and Mask R-CNN-predicted (right column) images. The red, yellow, and blue masks denote 111 loops, 100 loops, and black dot defects, respectively. The predictions shown here were made with an IoU value of 0.3 from a model fit and evaluated on dataset 1 "initial split."

Detailed materials-centric property statistics obtainable from the mask R-CNN model

In this section, we present a discussion of materials-centric properties obtained from the Mask R-CNN model predictions, specifically the distributions of predicted versus true defect sizes, shapes, and areal densities, and an approximation of the expected increase in yield stress based on a dispersion hardening model. Throughout this section, fits to dataset 1 "initial split" are used, and an IoU value of 0.3 is used based on the discussion in the previous section. Figure 3 shows histogram distributions of true and predicted values of defect shape and defect size. We examine two cases for each distribution: a case where all true and predicted defects are used in the analysis, and a second case examining only instances where a defect was found in the correct location, based on the implemented IoU value of 0.3. These two situations provide us with slightly different information regarding model performance. For the situation assessing all defects (Figures 3A and 3C), this comparison is indicative of the errors one may expect when applying the model to new test images where the number and locations of defects are not known *a priori*. For the situation of assessing only found defects (Figures 3B and 3D), this comparison is indicative of how well the

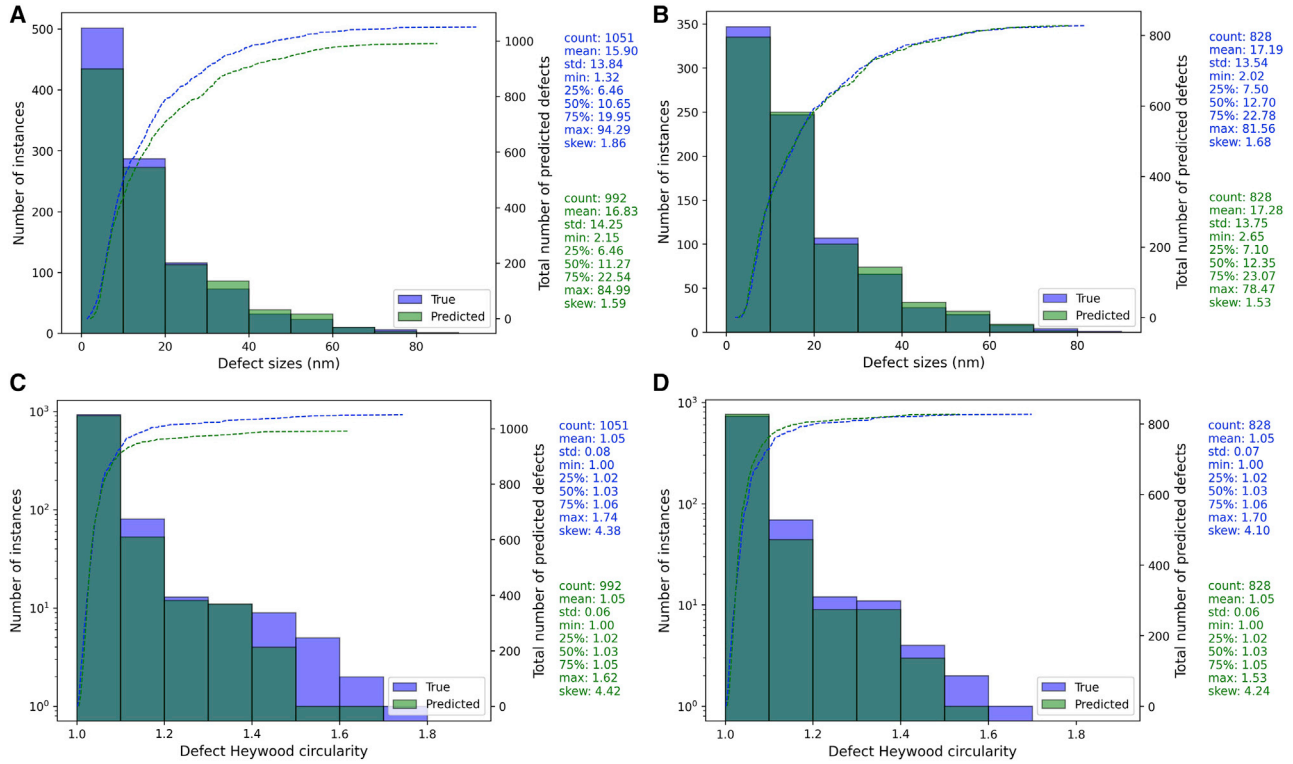


Figure 3. True and predicted defect size and shape distributions

(A–D) Histograms comparing distributions of true and predicted defect sizes (A and B) and defect shapes (C and D), computed as the Heywood circularity, for all true and predicted defects (A and C) and only defects found in the correct location (IoU = 0.3) (B and D). The defect number histograms in (C) and (D) are log scale. The dashed lines indicate the cumulative distributions of defect sizes and shapes, with object totals denoted by the right axis.

model can predict the size and shape of defects for the case where it has explicitly found a defect in the correct location. Although an analysis such as that shown in Figures 3B and 3D requires the defect positions to be known *a priori*, it is a useful analysis because it removes the effect of false positives and false negatives when considering how well the model is able to predict defect size and shape. In Figure 3, the error in the mean values of defect shape and size, when considering all true and predicted defects, are nearly 0% (accurate to two decimal places) and 7.1%, respectively. Qualitatively, in Figure 3, the distributions between true and predicted defects are generally in very good agreement, and the distributions match more closely for the case of comparing found defects only. This result makes sense, given that comparing the distributions between all true and predicted defects will have contributions from false positives and false negatives, which is expected to alter the overall distribution compared with only comparing correctly identified defects. The large fractional errors observed for larger values of Heywood circularity above about 1.5 are for bins with less than 10 defects and, therefore, sensitive to small counting errors. Also, defects with Heywood circularity above about 1.5 almost always consist of long edge-on ⟨100⟩ loops. We speculate that the model undercounts the edge-on ⟨100⟩ loops because of a class imbalance in the dataset, where there are fewer ⟨100⟩ loops compared with the other defect types, and we later show that the present model can still be improved by adding labeled data of ⟨100⟩ loops. It is possible that the model may confuse the edge-on ⟨100⟩ loops with pre-existing line dislocations, which are considered part of the image background and not a feature of interest. The line dislocations are considered part of the background and not a defect of interest to detect and quantify because these line dislocations are present in the

material prior to irradiation.[34] For the present application of detecting and quantifying defects in FeCrAl alloys, the focus was placed on detecting and quantifying the dislocation loops and black dot (sometimes abbreviated as "bdot" in this work) defects, which arise as a consequence of irradiation, resulting in hardening of the material.

In Figure 4, we take the defect size distribution data for all true and predicted defects from Figure 3A and break it up to be on a per-defect-type basis. In Figure 4, the shapes of the predicted defect size distributions well match the true distributions, although two deviations are notable. First, in Figure 4A, the predicted black dot size distribution skews toward values smaller, on average, than the true values. Second, in Figure 4B, the number of predicted instances of ⟨111⟩ loops is slightly overestimated, and in Figure 4C, the instances of ⟨100⟩ loops are slightly underestimated even though the shape of the predicted size distribution well matches the true distribution.

Another useful way to represent the comparisons of true and predicted defect statistics is by way of parity plots. In Figure 5, we present parity plots of the true versus predicted defect shape, size, and density split out by defect type. Each data point plotted in Figure 5 represents the calculated defect statistics from an individual test image. This analysis is useful for picking out particular images that may perform better or worse than others as well as identifying problematic outlier images. For example, this analysis enabled us to pick out a single test image with a very large number of true black dot defects whose count was severely underestimated by the model (lower right corner in Figure 5E). This single test image thus contributed to most of the observed errors for the black dot defect densities. Although there is some variation in how well individual images are predicted, the model does quite well on the scale of individual images, with mean absolute error values of the per-image defect size of about 3 nm and per-image defect density of about 0.5×10^4 #/nm^2 (we use # to denote "number of defects"). It is also notable that, when taken as an average over the entire test image set, the model predictions improve and become excellent for all three properties of interest. Instead of representing the defect size in nanometers, one could also assess the error using units of pixels. In addition, instead of assessing defect densities as number of defects per square nanometer, one could examine the errors in defect counts by counting the total true and predicted defects of each type for each image. We also examined the errors in the model performance for this dataset using pixels and total defects per image as an assessment of defect size and defect density, respectively (Figures S2 and S3).

As a final visualization to help further quantify and better understand per-image and overall model errors, we took the same per-image data from above and re-cast the values in terms of percent error for each defect type. An example of this result is given in Figure 6 for the case of defect size errors. Analogous plots of defect shape and defect density errors can be found in Figures S4 and S5. Figure 6 enables further comparison between per-image and overall expected errors. For instance, in Figure 6, it is evident that the defect size percent errors are typically about 30% or lower and that a single test image shows particularly poor prediction of ⟨100⟩ loop sizes. Further examination of predictions made on this poorly predicted test image show that this large percentage error is not due to the model predicting many ⟨100⟩ loops poorly in terms of their size but, rather, that the model predicts one large loop in particular as ⟨100⟩ when the ground truth indicates that it is a ⟨111⟩ loop. This loop is much larger than the other ⟨100⟩ loops in the image, resulting in a large size error. It is also worth noting that, when taken as an average, the per-image errors for

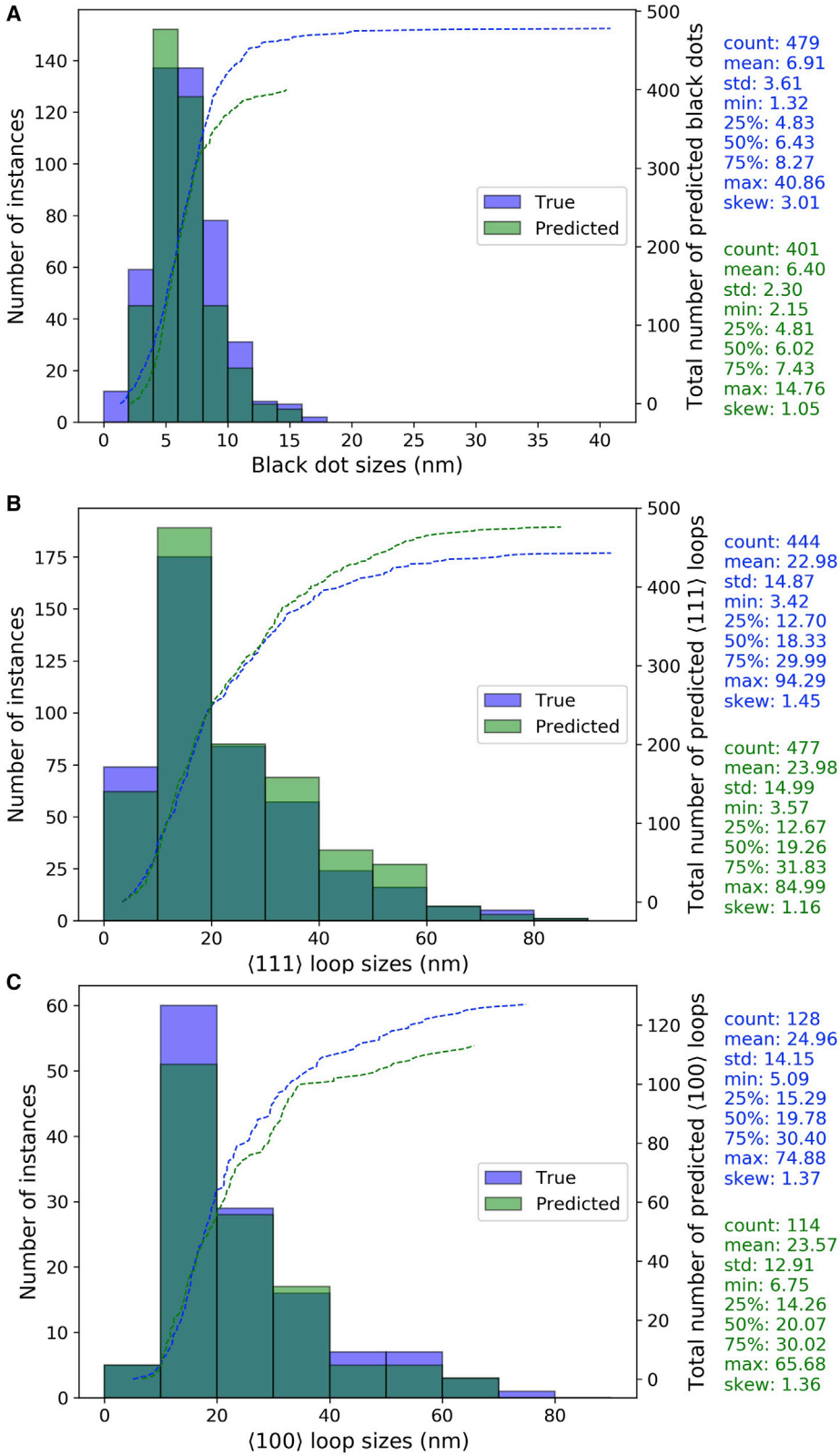


Figure 4. True and predicted defect size distributions by defect type

(A–C) Histograms of defect size distributions for all found defects split out by defect type: (A) black dot defects, (B) ⟨111⟩ loop defects, and (C) ⟨100⟩ loop defects. The dashed lines indicate the cumulative distributions of defect sizes, with object totals denoted by the right axis.

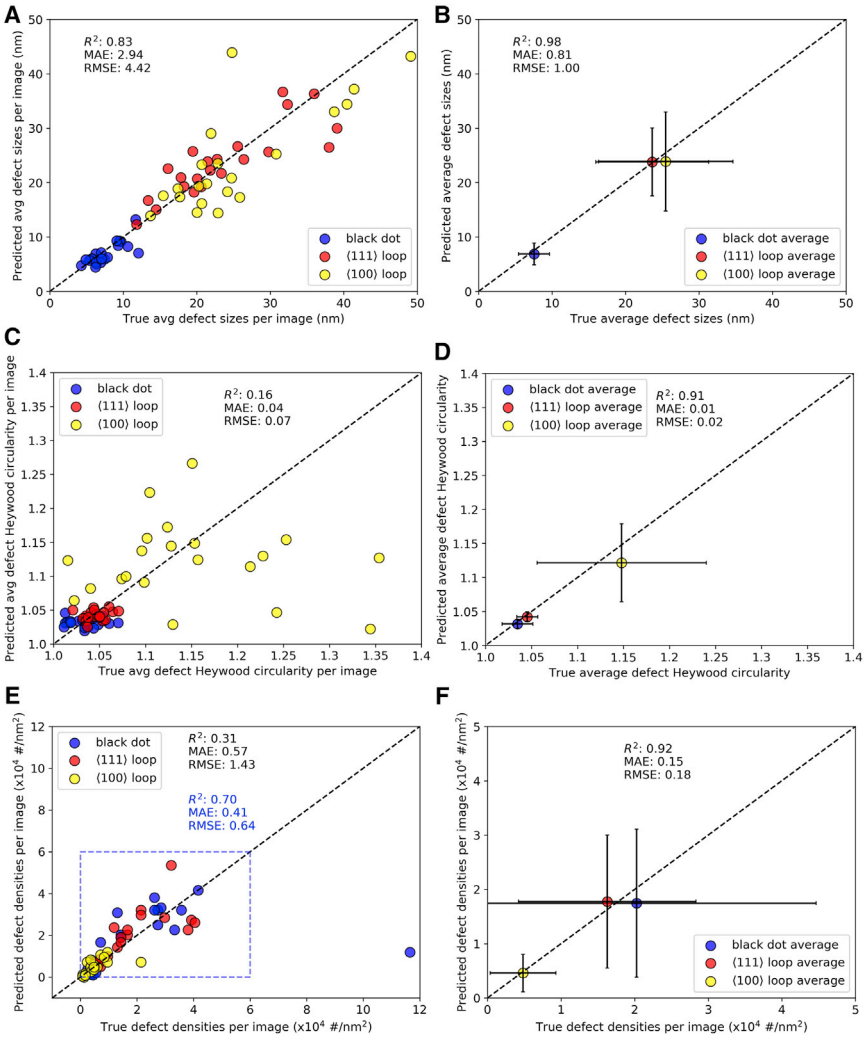


Figure 5. Parity plots of per-image and average defect property predictions

(A–F) Parity plots comparing true and predicted defect sizes (A and B), shapes (C and D), and densities (E and F) on a per-validation image basis (A, C, and E, left column) and averaged over all validation images (B, D, and F, right column). In all panels, blue, red, and yellow points represent values for black dots, ⟨111⟩ loops, and ⟨100⟩ loops, respectively. For panels averaged over all validation images (B, D, and F), the points denote the average value for the respective defect type, and the error bars indicate standard deviations in the true and predicted values. In (E), the statistics listed in blue correspond to the data points enclosed in the dashed blue box, which removed the single outlier image with a significantly underestimated number of black dot defects.

defect sizes are under 20%. Further, if the entire distribution of defect sizes is taken together and not separated on a per-image basis, then the average errors drop further and are consistently under 10%.

A major reason for quantifying the defect type, size, shape, and density is that these properties play a role in determining alloy mechanical properties. As mentioned in the Introduction, the dispersed barrier hardening model uses information of defect type, size, and number density to determine the increase in material yield or ultimate tensile strength (hardening) resulting from the creation of defects. Typically, only average size and density information is readily available. However, with using the present data and models, the full size distributions and more detailed defect

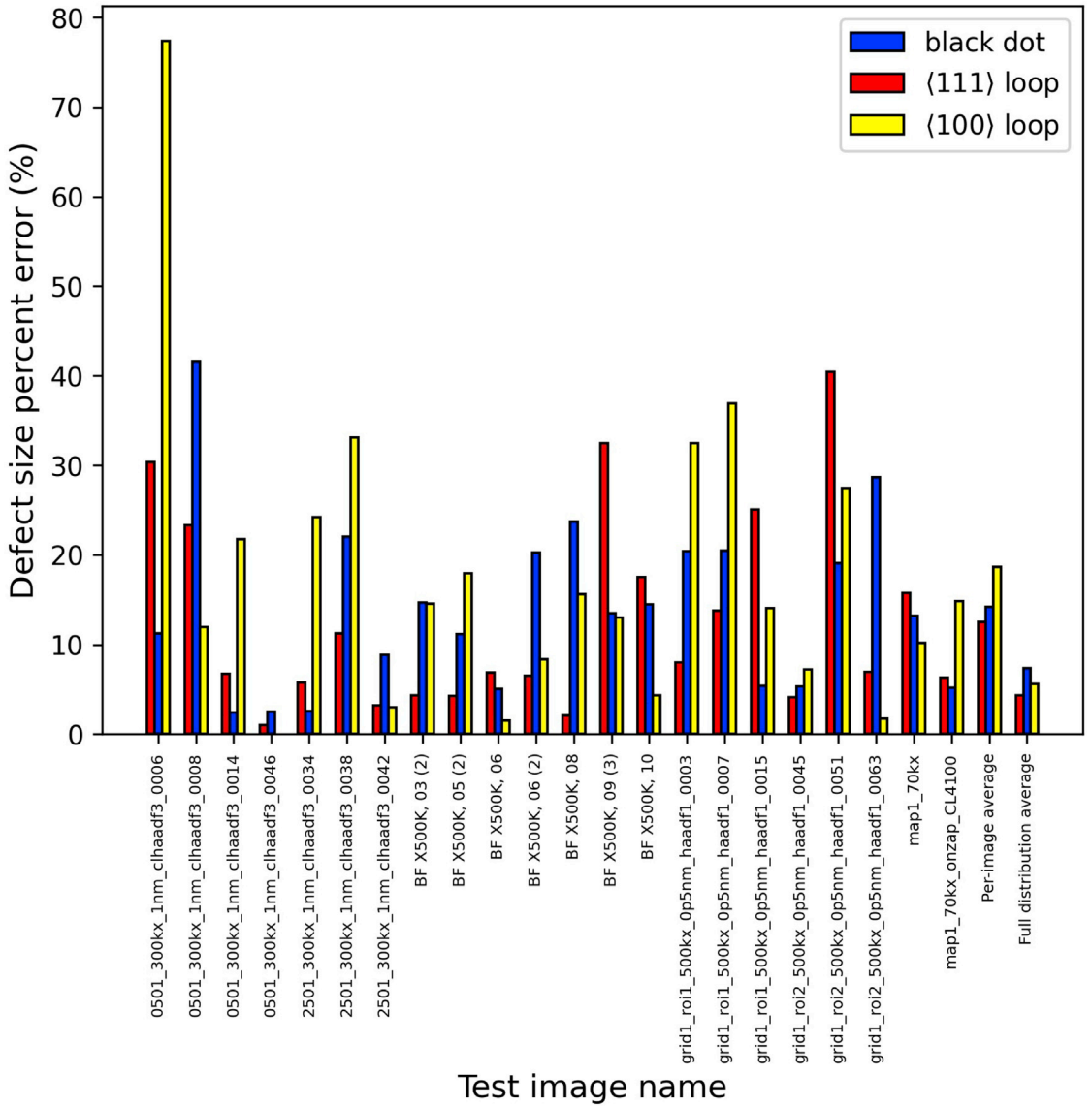


Figure 6. Defect size percent errors by test image

The bar plot shows the per-image predicted defect size percent error for each defect type. Also provided on the right side of the plot are the per-image average and the values obtained from the full distribution. The test images shown here are from dataset 1 "initial split." The labels along the x axis denote the test image names for test images comprising the dataset 1 "initial split" test image set. The label "per-image average" consists of the averaged per-image defect size percent error, and the label "full distribution average" corresponds to the average percent error of every individual defect, as if considering all test images constitutes one large image.

density data for each defect type are available, enabling a more detailed analysis of hardening. Here, we compare the machine-learning-predicted radiation-induced hardening for the present data with the hand counting ground truth value. Following the work of Field et al.,[34] we use the simplified dispersed barrier hardening model with materials constants from Field et al.[34] (see Note S2 and Tables S1 and S2 for more details) and calculate the expected (from the ground truth) and predicted (from the Mask R-CNN predictions on test images) hardening. In practice, this is done by calculating the hardening contribution of each defect type for each image and then summing the contributions to obtain the total hardening. This summing step can be done by simply adding all contributions (linear

sum) or adding the squares of the contributions and taking the square root of this sum of squares (quadrature sum), and it is often unclear which method is best when mixed features are present in the microstructure, so here we did both.[34] From this analysis, we find that, depending on the image examined, the hardening amount ranges from about 50–200 MPa. We find that the mean absolute error between true and predicted hardening is 16.05 (11.05) MPa based on linear (quadrature) sum. These absolute error values translate into mean absolute percent errors of 12.9% (13.7%) for linear (quadrature) sum. These findings indicate that the present Mask R-CNN model predictions of defect sizes and densities can be used to predict the expected hardening with an average error in the range of 10–20 MPa, which is approximately 10% of the total expected hardening based on the observable defects in the images. Other, non-observable features, such as small vacancy and interstitial clusters that exist under the resolution limit of the TEM, as well as precipitates, are not considered.

Understanding variations in model performance based on training and testing data choice

The performance of machine learning models of all types can be sensitive to the choice of datasets used for training and testing. In object detection, cross-validation is not typically performed because the dataset sizes for training and testing are often very large (e.g., a few million instances). Within the limit of large datasets, cross-validation will typically not yield significantly different results in the model predictions because the training and test sets are sampled from the same domain, and cross-validation can become computationally impractical. However, for more specific object detection applications, such as the present work of finding defects in irradiated alloys, the volume of data is typically much smaller, often on the order of a few thousand instances instead of a few million.

Here, to assess the sensitivity of model performance to the choice of which images are used for training and testing, we perform random cross-validation of the train and test sets. This process consists of making five random splits of the images, always holding out 21 images for testing and using the remaining images for training. Splitting the images in this way makes it so that about 15%–20% of the total defects are reserved for testing and that the training and testing sets are drawn from roughly the same domain.

The full results of the random leave out cross-validation test are shown in Table S4, and here we summarize our key findings. It is evident that the effect of different images used in training and for testing is moderate in scale, with ranges (standard deviations) of the overall defect find F1 score, overall defect type F1 score, average defect size error (all defects), and average defect density error of 0.04 (0.02), 0.05 (0.02), 9.25% (3.80%), and 13.65% (5.31%), respectively. These ranges and standard deviations in key statistics are larger than what was found from running the same model multiple times to assess model randomness (Note S3; Table S3), which indicates that the choice of training and test images, at least for this particular application, may yield meaningfully different predictions of model performance.

Figure 7 provides parity plots visualizing these best and worst cross-validation splits for predicting defect size and defect density. An observation from Figure 7 is that the error values between the best and worst cross validation split differ by factors ranging from about 1.5–2.5. More specifically, the root mean squared error (RMSE) of defect density changing from 0.70×10^4 #/nm^2 (best) to 1.74×10^4 #/nm^2 (worst) is a factor of 2.5, and RMSE of defect size changing from 6.00 nm (best) to 8.83 nm (worst) is a factor of 1.5. For the defect size error, one test image is the main culprit for the

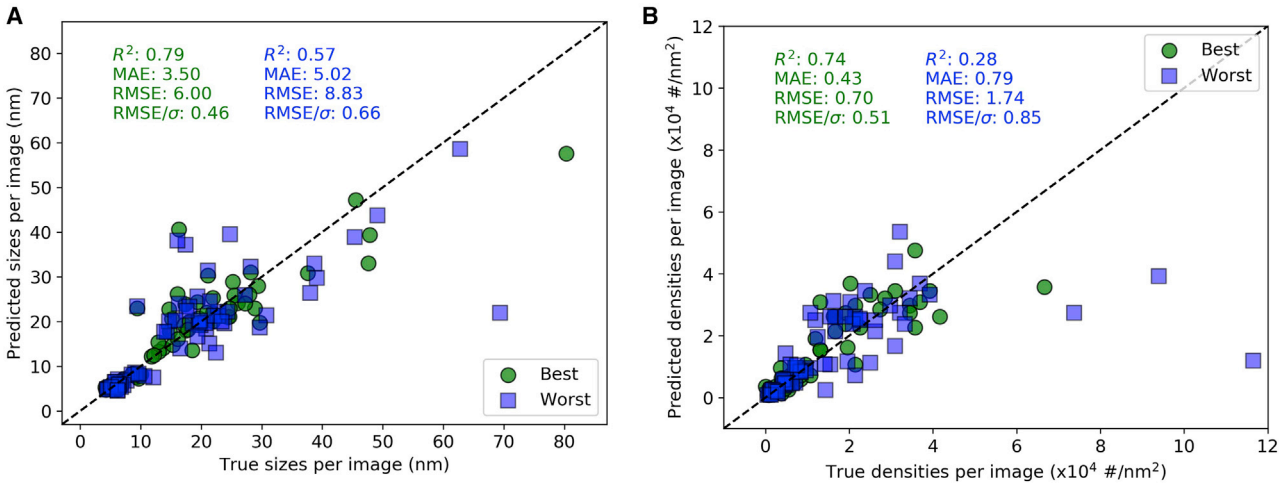


Figure 7. Parity plots showing best and worst model performance

(A and B) Parity plots showing the predicted versus true defect sizes (A) and densities (B), where each data point results from a specific test image. The green circle and blue square data denote the best and worst cross validation split for each quantity, respectively.

worsened trend, which can be traced to poor predictions of ⟨100⟩ loop defect sizes for this one image. We speculate that this error is due to missing instances of ⟨100⟩ loops and misidentifying other defect types as ⟨100⟩ loops, pushing the average ⟨100⟩ loop size for this image to a small value. For the defect density error, three test images showed significant underprediction of defects, which, for all cases, were instances of the model significantly underestimating the number of black dot defects. Overall, this analysis indicates that, just as in the case of non-deep-learning machine learning applications, performing numerous splits of cross-validation is useful for obtaining a more informed assessment of model performance.

Understanding the limitations of the model performance and domain based on training and testing data choice

In addition to random leave out tests, it has been demonstrated in other machine learning applications of materials science that leaving out physically motivated groups of data is a useful method to more selectively probe model performance.[35–37] Therefore, our second cross-validation test consists of leaving out physically motivated groups of images to more rigorously evaluate the domain of applicability of our model. These leave out group (LOG) tests are described as follows. LOG test 1 (leave out irradiation condition) keeps the alloys consistent between train and test image sets, but the irradiation conditions between the train and test sets are different. These irradiation conditions differences make it so that the training set will be on smaller ⟨111⟩ loops and ⟨100⟩ loops and a higher density of black dots on two alloys compared with the larger loops and lower density of black dots in the test set. LOG test 2 (leave out alloy test) keeps the irradiation conditions consistent between train and test image sets, but the alloys are different. These compositions and sink density differences make it so that the training set will have large loops compared with the test set. LOG test 3 (leave out sample and microscope type) keeps groups in the domain based on the microscope and sample used. The training dataset images were acquired on an older microscope (Philips CM200) with simple starting microstructures, whereas the test dataset images were acquired on a newer microscope (FEI Talos F200X or JEOL 2100F) with samples that had a more complicated microstructure. The training dataset was obtained entirely by K.G.F., and the test dataset had two microscopists, K.G.F. and Dalong Zhang.[38]

Table 1. Summary of LOG cross-validation test results

Group test	Dataset type	Number of train images (defects), number of defects per type	Number of test images (defects)	Defect ID F1 at IoU = 0.3	Defect find F1 at IoU = 0.3
Leave out irradiation	dataset 2	12 (370) bdot: 117 ⟨111⟩: 195 ⟨100⟩: 58	9 (649)	bdot: 0.86 ⟨111⟩: 0.85 ⟨100⟩: 0.26 overall: 0.66	0.79
Leave out irradiation	dataset 2 expanded	21 (1,340) bdot: 707 ⟨111⟩: 423 ⟨100⟩: 210	9 (649)	bdot: 0.85 ⟨111⟩: 0.81 ⟨100⟩: 0.22 overall: 0.63	0.80
Leave out alloy	dataset 2	9 (649) bdot: 268 ⟨111⟩: 334 ⟨100⟩: 47	51 (6,837)	bdot: 0.80 ⟨111⟩: 0.50 ⟨100⟩: 0.36 overall: 0.55	0.69
Leave out alloy	dataset 2 expanded	18 (1,732) bdot: 767 ⟨111⟩: 651 ⟨100⟩: 314	51 (6,837)	bdot: 0.87 ⟨111⟩: 0.62 ⟨100⟩: 0.43 overall: 0.64	0.66
Leave out microscope/ sample	dataset 2	18 (1,606) bdot: 598 ⟨111⟩: 792 ⟨100⟩: 216	70 (3,285)	bdot: 0.81 ⟨111⟩: 0.68 ⟨100⟩: 0.33 overall: 0.60	0.75
Leave out microscope/ sample	dataset 2 expanded	69 (8,569) bdot: 4,038 ⟨111⟩: 2,493 ⟨100⟩: 2,038	70 (3,285)	bdot: 0.82 ⟨111⟩: 0.68 ⟨100⟩: 0.57 overall: 0.69	0.75

Table 1 summarizes the results for the LOG cross-validation tests. From Table 1, a few key results emerge. First, the overall defect type F1 scores for the LOG tests are generally lower, in the range of 0.55–0.69, than the overall defect type F1 scores obtained from the random leave out cross validation tests, which were in the range of 0.77–0.82. The lower values of the overall defect type F1 scores and their larger range for the LOG tests versus the random leave out tests make sense. The F1 scores are lower for the LOG test because it is a more demanding test of the model because the test images are farther outside the domain of the training data than for the random cross-validation test, where the training and test data are drawn from the same domain of images. Because the training and test image sets are more similar for each iteration of random cross-validation, the range of reported F1 scores is smaller. The LOG tests examined here contain different train/test splits that differ markedly in their character, resulting in a larger range of model performance quality.

In addition to differences in model performance between random versus LOG cross-validation tests, we can assess the change in model performance for the LOG test when the training dataset for each test is changed from using the initial dataset 2 to the newer dataset 2 expanded dataset. The performance differences in the LOG tests between the use of dataset 2 versus dataset 2 expanded for training suggests that, for these more demanding tests, the larger amount of training data contained in dataset 2 expanded are useful from the standpoint of broadening the domain of applicability of the model. For example, for the leave out alloy test, the overall defect type F1 score increased from 0.55 to 0.64 when training on dataset 2 versus dataset 2 expanded, and for the leave out microscope/sample test, the overall defect type F1 score increased from 0.60 to 0.69. For the leave out irradiation

test, the F1 score remained approximately unchanged between training for the two different datasets. By inspecting the defect type F1 score per defect type, we can see that the improvement in model performance for the leave out alloy and leave out microscope/sample tests is due to different factors. For the leave out alloy test, the improvement in defect type F1 stems from improvements in F1 scores of all three defect types. In contrast, for the leave out microscope/sample test, the improvement in defect type F1 comes from improvement in correctly identifying the ⟨100⟩ loops only.

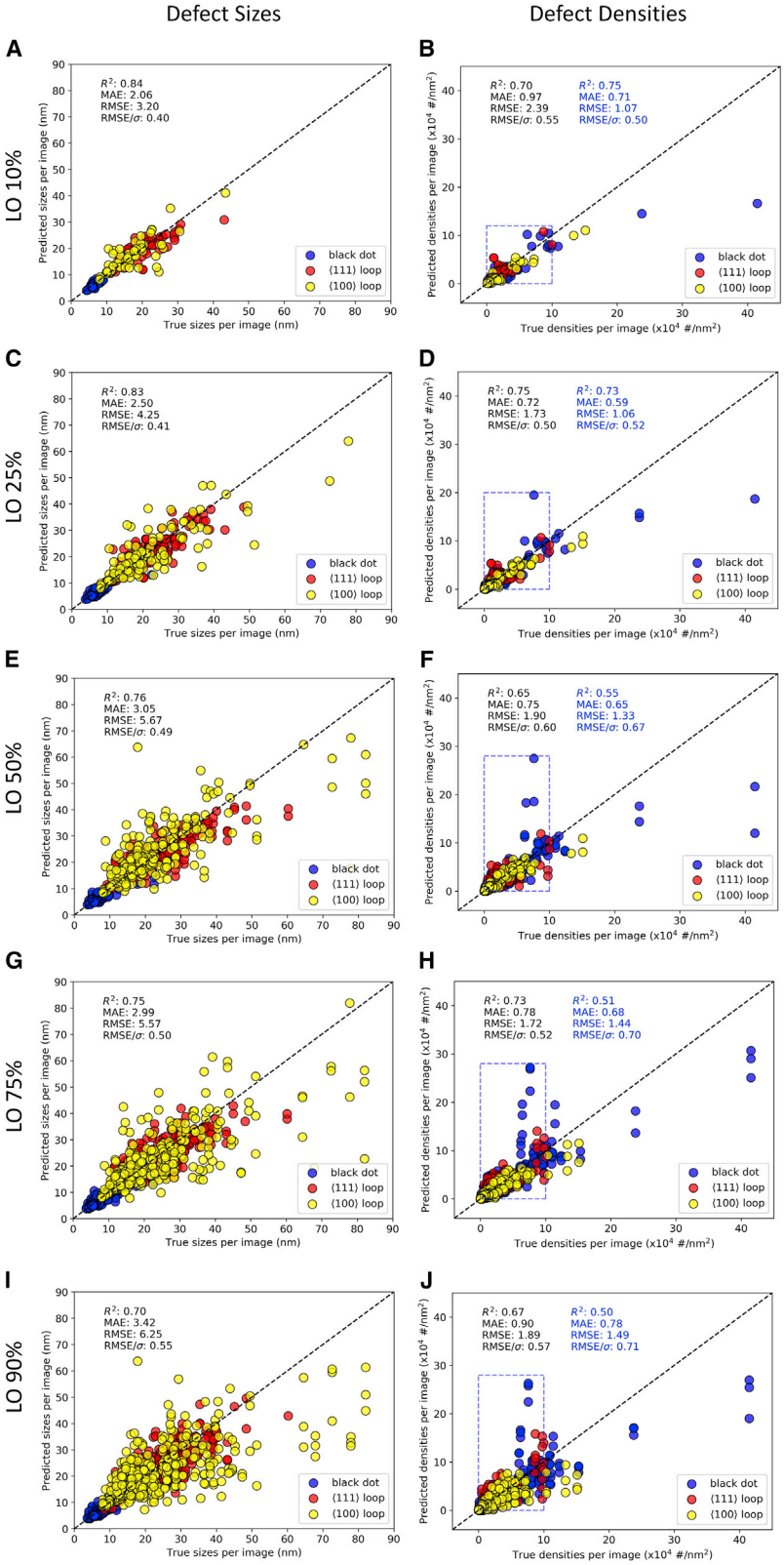
Examining the effect of ground truth labeling on the model performance by domain experts

As discussed in the Introduction, one issue when characterizing and quantifying defects in electron microscopy images is that establishment of the ground truth labels is done manually by human domain expert labelers. This labeling process inherently carries some level of subjectivity with it because different human labelers may disagree about whether a feature in an image constitutes a defect being present and the type of defect. In addition, some labelers may exhibit labeling patterns notably distinct from other labelers. For example, in the work of Li et al.,[28] when comparing the results of five human labelers quantifying the number and size of defects in a set of images, two labelers differed in their labeling systematically, with one labeler tending to categorize many more image features as defects compared with the other labeler.

Here we assess the performance of Mask R-CNN models trained on different ground truth datasets. The full results of this test are shown in Table S5, and here we summarize our key findings. Overall, the results of both datasets show very similar levels of average accuracy for all test statistics (e.g., defect find F1 scores of 0.81 and 0.82 for prediction on dataset 1 and dataset 2, respectively), where the differences in scores between the two datasets is of the same magnitude as observed from our test assessing model randomness (Note S3). One notable difference is that the dataset 1 model tends to show higher density errors for black dots (16.28% error versus 5.06% error for dataset 1 and dataset 2, respectively), and the dataset 2 model tends to show higher size errors for black dots (7.40% and 15.37% for dataset 1 and dataset 2, respectively). The cause of these differences is not clear, but we speculate that it may relate to the nature of the ground truth labels, where dataset 1 contains many instances of image features labeled as black dot defects that were not labeled as a defect at all in dataset 2. Mask R-CNN models trained using different ground truth labels perform very similarly, indicating that, at least for this case, the labeling performed by a particular domain expert may not hold obvious advantages compared with another expert. However, it is worth noting here that, if certain biases exist in the ground truth labels (for example, a labeler who systematically labels certain ambiguous image features as being black dot defects), then this bias will likely translate to the trained model. Because the predictive ability of a model can, as an upper bound, only become as accurate as the ground truth data it is trained on, future work should be devoted to establishing publicly available curated datasets that can be labeled and analyzed by many researchers in the field. This process will then involve subsequent model re-training to converge on the most accurate and predictive model of the most relevant metrics as agreed upon by the greater community.

Examining the effect of dataset size on the model performance

Analyzing the effect of training dataset size on model performance enables one to identify the amount of training data required for the model performance to saturate. In addition, even when the model performance does not improve beyond a certain amount of training data, it is likely that the domain of applicability of the model is



expanded, as discussed above in the context of the LOG cross-validation tests. In this section, we assess model performance as a function of training dataset size in two different ways. First, we use our largest dataset, dataset 2 expanded, to generate multiple splits of different leave out percent cross-validation tests, ranging from leave out 10% to leave out 90% of the images as test data. With these leave out percent cross-validation tests, we assess the performance of the model using parity plots of predicted versus true defect sizes and defect densities of all test set images. For the second test, we construct learning curves that plot per-defect type F1 scores as a function of number of defects of each defect type used in the training data. For this second test, to construct the learning curves, data from the previously discussed LOG tests, the leave out percent tests to be discussed in this section, and additional runs using dataset 1 and random cross-validation to construct training sets of varying sizes were used.

For our first assessment of the effect of dataset size using leave out percent cross-validation, Figure 8 presents parity plots of defect sizes and defect densities split out by defect type for five cases of different dataset sizes. The dataset size was modified by performing multiple iterations of leave out percent cross-validation, with the leave out fraction consisting of 10%, 25%, 50%, 75%, and 90% of the images. Each leave out amount was performed three times, where each time a different random portion of the data was left out for testing. Several findings are evident from Figure 8. In general, the model performance generally improves as less data are held out (equivalently, as the amount of training data increases). More specifically, as the leave out fraction becomes larger, the ability of the model to predict defect sizes becomes significantly worse on a per-image basis, with the RMSE increasing from 3.20 nm (average of 3 iterations of leave out 10%) to 6.25 nm (average of 3 iterations of leave out 90%), an increase of nearly a factor of two. Interestingly, although the model performance worsens when leaving out up to 90% of the images, the predictive performance is still impressively robust in the limit of small amounts of training data. This finding may suggest that object detection models like Mask R-CNN may offer useful insights and predictions on rather sparse datasets containing fewer than 1,000 training instances, and this will be discussed in more detail below. Regarding the predictions of defect density with different leave out amounts, the trends when examining all of the data as a function of leave out amount do not show as clear a trend as the case of defect sizes, and the trend might be affected by the presence of a few images with very high black dot defect densities. However, if the analysis is instead focused on the region where the true defect density is less than 10×10^4 #/nm^2 (blue dashed boxes in Figure 8), which constitutes the vast majority of the images studied in this work, then the errors in defect density clearly increase from 1.07×10^4 #/nm^2 (leave out 10%) to 1.49×10^4 #/nm^2 (leave out 90%). As observed in past studies, increasing the amount of training data generally results in reduced prediction errors[30] and may also help broaden the applicability domain of the model.

For our second assessment of the effect of dataset size using all of the cross-validation tests described in this work, Figure 9 contains learning curve plots representing

Figure 8. Model performance with varying numbers of test images

(A–J) Parity plots comparing true and predicted defect sizes (left) and densities (right) for three random splits of 10% (A and B), 25% (C and D), 50% (E and F), 75%, (G and H), and 90% (I and J) cross-validation. The blue, red, and yellow data points denote average values from an individual test image for black dot, $a_0/2\langle 111\rangle$, and $a_0\langle 100\rangle$ loops, respectively. For the plots of defect density, the blue dashed box and corresponding statistics are for images where the true densities are less than 10×10^4 #/nm^2.

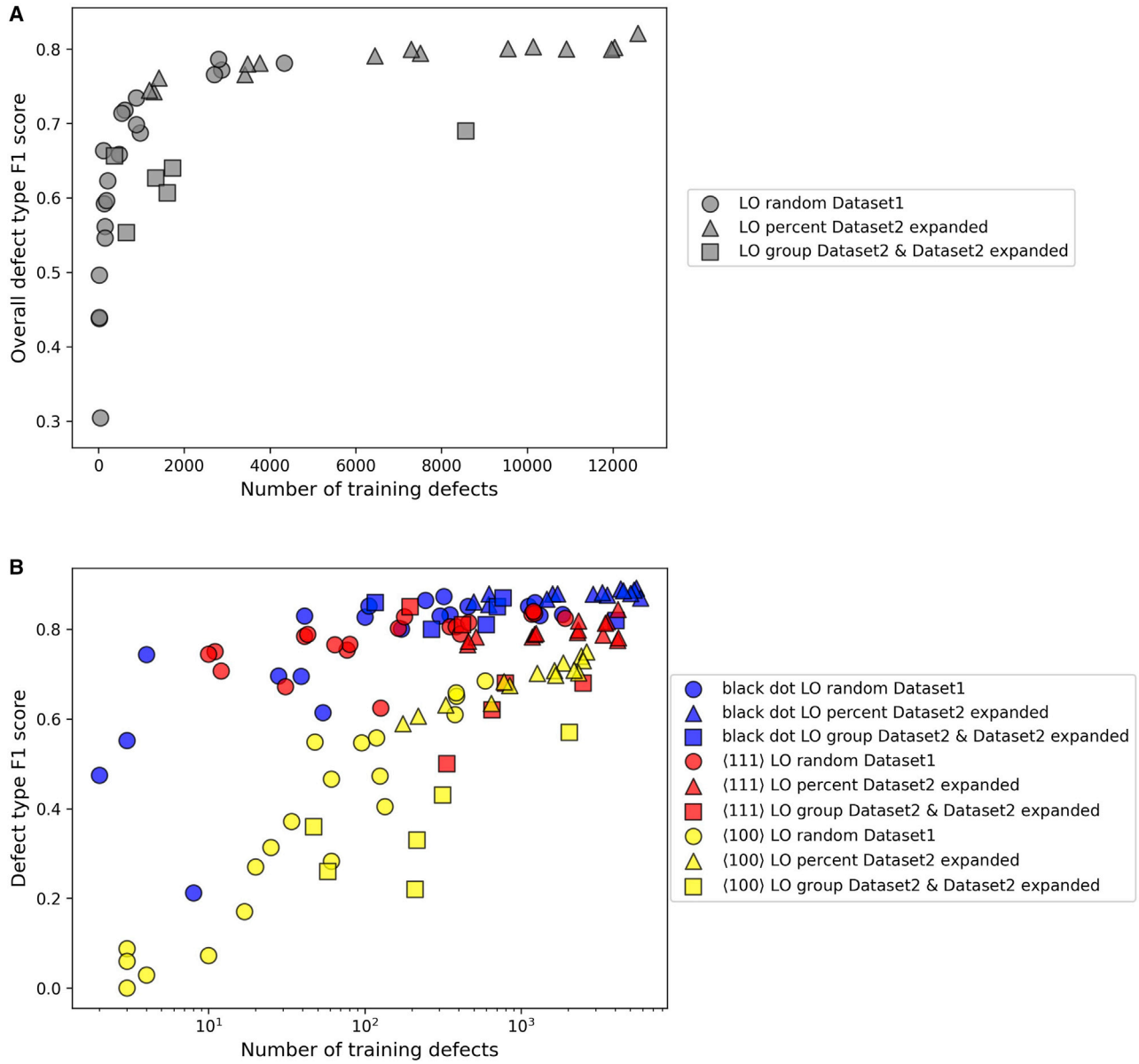


Figure 9. Model classification performance as a function of training set size

(A and B) Learning curve plots of (A) overall defect type F1 score as a function of the number of training defects and (B) defect type F1 score split out by defect type as a function of the number of training defects. The x axis in (B) is on a log scale.

the overall defect type F1 score versus the number of training defects (Figure 9A) and the defect type F1 score broken out by defect type versus the number of training defects, this time on a log scale (Figure 9B). There are a few key pieces of information we can extract from Figure 9A. First, the ability of the model to correctly identify defects quickly increases with the number of training defects, with a defect ID F1 score approaching 0.7 for models trained on fewer than 1,000 defect instances. After 1,000 defects, improvement is incremental with significantly diminishing returns, and a defect ID F1 score of about 0.8 is achievable using more than 6,000 defects. Extrapolating these results suggests that achieving a meaningful defect ID score above 0.8 may require a dataset with more than 50,000 defects. In Figure 9A, the data points for our tests of leave out percent cross-validation using dataset 2

expanded (gray triangles) and random leave out cross-validation using dataset 1 (gray circles) fall on the same curve. This result makes sense because both of these methods select training and test images at random. These two datasets differ in the criterion used to select how large the training sets were and the test image sets used. The random leave out tests (gray circles) used dataset 1, the test image set was the same in all cases, and the number of training images varied. The leave out percent tests (gray triangles) used dataset 2 expanded, and the test image set changed for each test. The data points corresponding to the LOG tests (gray squares), except for one instance, always fall below the random cross-validation data points for the same amount of training data. This is to be expected, given that the LOG test is more demanding, and the test data are generally farther from the domain of the training data compared with the random cross-validation tests.

In Figure 9B, we take the same data from Figure 9A but break out the defect ID F1 scores by defect type; for easier examination of the differences of F1 score between defect types, we plot the number of training defects (i.e., the x axis) using a log scale. Examining the data in this manner shows that, within the limit of very small datasets (e.g., around only ⟨100⟩ defects), the model still performs surprisingly well at correctly identifying black dots and ⟨111⟩ loops, whereas there is very poor predictive ability of the ⟨111⟩ loops. When the number of black dots and ⟨111⟩ loops used for training is in the range of a few hundred, the defect ID F1 score is already above 0.8 for these defect types. Thus, expanding the amount of labeled data in our database mainly resulted in the model performing better on the ⟨111⟩ loops, as evidenced by the collection of yellow triangle data points with F1 scores in the range of 0.7–0.75 for the highest defect counts. The increasing trend of ⟨111⟩ loop ID F1 score suggests that the model performance on identifying this defect type still has room for improvement with inclusion of additional labeled data, even beyond the expanded dataset prepared for this study.

From Figure 9B, we can see that the performance of the model in identifying black dots is highest, followed by ⟨111⟩ loops, followed by ⟨100⟩ loops being the worst. This trend is in agreement with the qualitative visual complexity of these defect types; black dots are the most uniform in size, shape, and overall appearance and should thus be easiest to categorize. ⟨111⟩ loops are more varied in their size and appearance than black dots but are not as visually diverse as ⟨100⟩ loops, where ⟨100⟩ loops have edge-on and face-on orientations, yielding a wider range of visually distinct sizes, shapes, and contrasts, and the similarity of the edge-on orientation with background line dislocations result in a harder classification task. These qualitative comparisons are also in line with the LOG test results, where the black dot predictions between random and LOG cross-validation were effectively identical, whereas the ⟨111⟩ loop and, in particular, the ⟨100⟩ loop F1 scores were markedly lower for the LOG tests compared with the random cross-validation tests. This performance trend is indicative of black dot defects appearing visually very similar between different groups assigned here, whereas the size, shape, and prevalence of the ⟨111⟩ and ⟨100⟩ loops change more dramatically between the train and test sets used for the LOG tests compared with the random cross-validation tests.

DISCUSSION

This work and others like it provide an avenue for deep learning models to improve and accelerate materials modeling efforts. Understanding the effect of different irradiation-induced defects in metal alloys on the resulting materials properties and performance hinges on quantifying the numbers, sizes, and shapes of different defect

types in the material. The present Mask R-CNN model enables fast, automatic quantification of all of these quantities as well as refinements to enable more accurate materials modeling by including quantitative data of defect size and shape distributions instead of just commonly used average values or models that do not typically include effects related to defect shape.

This work highlights not only the successes and usefulness of deep learning object detection methods for finding defects in microscopy images, but also lays out some of the current limitations and potential issues to be aware of when evaluating the performance of a model. In particular, some high-level findings that may be broadly useful for evaluating model performance can be summarized as follows.

- Understanding variations in model performance based on data choice. We found that the choice of training and test images yields meaningfully different predictions of model performance. As an example, we found that the error values for defect size and density errors between best and worst cross-validation split differ by factors ranging from about 1.5–2.5. This finding indicates that, just as in traditional machine learning evaluations, cross-validation is a useful tool for evaluating performance of object detection models.
- Understanding limitations of model performance and domain. The LOG tests examined in this work contain groups of train and test images that differ markedly in their character (for example, separating sets of images based on alloy type), resulting in a larger range of model performance quality compared with random cross-validation. For these more demanding tests, we found that the larger amount of training data contained in our expanded database was useful for broadening the domain of applicability of the model but did not improve the model performance in random cross-validation. This finding suggests that expansions of present databases should be focused on including data that exist in different domains from what is already present and that reducing cross-validation score may not be a good metric to assess the value of additional data because it misses gains in the domain of applicability of the model.
- Impact of domain expert labeling to make ground truth. The generation of ground truth labels can be subjective, leading to different labels from different domain experts. When considering model performance on the same dataset labeled by different experts, we found that very similar levels of average accuracy for all test statistics were obtained, where the differences in scores between the two datasets is of the same magnitude as observed from our test assessing model randomness. However, if certain biases exist in the ground truth labels (for example, when a labeler systematically labels certain ambiguous image features as being black dot defects), then this bias will likely translate to the trained model.
- Effect of dataset size on model performance. We found that, when leaving out up to 90% of the images, the predictive performance is still impressively robust in the limit of small amounts of training data. More specifically, we found that a defect ID F1 score approaching 0.7 for models trained on fewer than 1,000 defect instances, although achieving scores significantly above 0.8, was estimated to potentially require more than 50,000 instances. This finding suggests that these models may be reliably trained on datasets that can be generated with modest human labeling efforts of even just a few hours.

We would like to point out that one shortcoming of the present work is that our model is restricted to a single material class (FeCrAl alloys) and uses data for a single

STEM imaging condition (bright field, [100] on zone). Regarding material type, defects like the dislocation loops studied here will manifest with different geometries when the material is changed from, for example, a ferritic steel with the body-centered-cubic crystal structure like the FeCrAl alloys studied here, to an austenitic steel with a face-centered-cubic crystal structure. This change in defect geometry will necessitate training a new model or re-training the present model with these defect instances to increase the model domain and enable accurate predictions on a new material. Regarding imaging condition, analyzing images where the imaging was conducted using a different zone axis (e.g., [111] instead of [100] used here), even for the FeCrAl alloys studied here, will result in the defect loops having different orientations and shapes (e.g., a loop being in plan view versus edge on), and varying image contrasts will change what the model feature map perceives as indicating defected versus background regions, again necessitating model re-training. Model re-training on new datasets may be very time consuming because of the need for acquiring sufficient labeled data. In this regard, state-of-the-art methods, such as single- or few-shot learning, may be promising avenues for training new models using very few instances of new labeled data.[39]

We believe the potential of using object detection models for analyzing electron microscopy images is far from being realized. One area of future work in this space might focus on developing a more general defect model for irradiated alloys that incorporates more than the three defect types considered here and is able to classify dislocation lines, cavities, and voids formed from gas bubbles and precipitates, perhaps also taking into account different imaging conditions. Another area of promising future work centers around the exploration and development of methods for synthetic training data generation, including physics-based modeling such as the common "multi-slice" simulations, lower-order models based on simplified assumptions and physical descriptions, and machine-learning-centric methods of synthetic data generation such as through use of generative adversarial networks (GANs).[40,41,42] These methods may enable more robust and rapid model training and evaluation because the reliance on costly and time-consuming experimental data labeling would be reduced, perhaps significantly. A key development to support adoption of these new methods is developing community-based software packages that enable rapid cloud-based dissemination of automated detection packages. To accomplish this, it will be essential to establish a minimum performance metric for adoption and use of any developed automated defect detection framework that is agreed on by the community. Formation of a robust, community-driven database of labeled TEM images for rapid development and qualification of automated defect detection frameworks will greatly accelerate the development and assessment of new models. Improved data sharing frameworks, such as the Materials Data Facility (MDF)[43] and cloud-based services for hosting machine learning models such as DLHub,[44,45] are enabling the intersection of materials data and trained machine learning models in a manner that will likely be transformative to the materials research community in the coming years. As a step toward this goal, and in the same spirit as similar efforts of democratization of deep learning models like that of von Chamier et al.,[46] we made the final trained Mask R-CNN model, images, and analysis scripts publicly available, along with an easy-to-use Google Colab notebook for running the trained model on user-provided images and for re-training the model-provided additional labeled data (see Data and code availability under Experimental procedures).

The results of the present study demonstrate that the use of standard, off-the-shelf object detection models is extremely effective at quantifying the average size,

shape, and density of different object types in the context of defects in electron microscopy images. The findings of this work and findings in recent similar studies[29–32] suggest that maturation of computing hardware (e.g., faster graphics processing units [GPUs]) and object detection software (e.g., the open-source Detectron2 package) has reduced the barrier required to perform meaningful object detection tasks. Consistent with these advancements, several companies have developed software packages to aid in performing traditional computer vision analysis and deep learning analysis of images, including semantic segmentation of objects in images. These tools include Reactiv IP's Smart Image Processing package, Object Research Systems' Dragonfly package, and EPFL's DeepImageJ package, to name a few. Application-specific use of object detection methods with these commercial packages or open-source packages like Detectron2, such as model evolution via re-training on newly available data and cloud-based model hosting for broad dissemination, along with implementation of new state-of-the-art object detection methods, such as few-shot learning[39] or vision transformers (ViTs),[47–49] may enable a transformative leap in the manner in which electron microscopy image analysis is performed.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests should be directed to and will be fulfilled by the lead contact, R.J. (rjacobs3@wisc.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The datasets generated and/or analyzed during the current study are available on Figshare (Database: https://doi.org/10.6084/m9.figshare.14691207.v3). The trained model on all images comprising dataset 2 expanded, a Google Colab notebook, and associated Python scripts to make predictions on new images and save the associated data is also available on Figshare (Database: https://doi.org/10.6084/m9.figshare.14691207.v3). Supplemental information discussing the effect of model randomness and model hyperparameters on initial model performance, additional analysis plots of predicted materials properties, and more information about the hardening calculations are also available.

FeCrAl image database

The image database used in this study consists of images of FeCrAl alloys that have undergone neutron or ion irradiation. The images are exactly those available derived from a series of published studies from Field et al.,[34,50,51] although some of the data have yet to be summarized in a publication, and we extended the labeling, as discussed below. The samples are FeCrAl alloys but vary in composition, microstructure (including grain size and line dislocation density), and irradiation conditions. All images are from a single TEM imaging condition, specifically [100] on-zone bright-field STEM. These imaging conditions produce defects appearing as black contrast features on a white background. In the case of irradiated FeCrAl, on-[100] zone imaging results in open single-edge elliptical loops that are dislocation loops with a Burgers vector of $a_0/2\langle 111\rangle$ (referred to as $\langle 111\rangle$ loops), open double-edge elliptical loops and closed elliptical solid loops that are dislocation loops with a Burgers vector of $a_0 100$ (referred to as $\langle 100\rangle$ loops), and closed circular solid dots that are typically called black dot defects with a Burgers vector of either $a_0/2\langle 111\rangle$ or $a_0\langle 100\rangle$

(referred to as black dots). An example experimental micrograph showing the visual characteristics of each labeled defect type is shown in Figure S1.

As mentioned above, the image database used in this work has been used previously in the studies of Li et al.[28] and Shen et al.,[29] but these studies did not include pixel-level segmentation information. For this study, the image database was updated to include new labeling; specifically, new ground truth pixel-level segmentation annotations. We developed three datasets of labels. The first considered a set of 107 images that were labeled with pixel-level segmentation by a first group of domain experts who found 5,382 defect instances. These are not all images in the full set of images. We call this set of 107 images and 5,382 defect instances dataset 1. Then, to better understand how the labeling might affect the results, the same 107 images were labeled by a second set of domain experts, this time finding 5,053 defect instances. We call this set of 107 images and 5,053 labels dataset 2. Finally, to explore how using a larger set of labeled images might affect the results, we labeled additional images and joined them with dataset 2. This led to a new dataset with 182 annotated images and 13,675 defect instances, which we call dataset 2 expanded. Table S6 contains a summary of the basic characteristics of each dataset, including the number of images and number of each labeled defect type. Numerous different splits of train and test images and their associated defects are used throughout this work. Table S7 provides a summary of the number of images and each defect type present in the various train and test datasets analyzed in this study. All segmentation mask annotations for both image datasets were made using the VGG Image Annotator web application.[52] All of the data for these three datasets are available on Figshare (see Data and code availability).

Mask R-CNN methods

Throughout this study, we use the Mask R-CNN model as implemented in the Detectron2 package, which uses PyTorch as the backend. The Detectron2 package was developed by the Facebook AI Research (FAIR) team.[53] Detectron2 is freely available and enables implementation of many object detection models, such as Faster R-CNN,[54] Mask R-CNN,[55] and Cascade R-CNN.[56] These object detection models have been pre-trained on the ImageNet[7] or Microsoft COCO[57] (Common Objects in Context) image databases, enabling use of the transfer learning technique. When using transfer learning, the model backbone weights are frozen to those obtained from the previous ImageNet or Microsoft COCO image training, except for a small number of terminal layers (2 throughout this work). The weights in these terminal layers are then updated during the training process to tune the model for the particular application of interest; in this case, detecting certain defect types in electron microscopy images. All post-processing of Mask R-CNN model predictions and associated analyses were performed using in-house Python scripts, which we have made available on Figshare (see Data and code availability).

In this work, we evaluate the performance of our Mask R-CNN models on a number of different application-specific tests central to understanding the effect of different defect types on the mechanical properties of an irradiated alloy. These tests include how well the model can predict the areal density and size of defects in an image and how well the model can discern the location and type of defects in an image. Explanations of the key we quantify to evaluate the overall performance of the Mask R-CNN model are summarized in Table S8. The Heywood circularity factor is defined as the perimeter of an object divided by the circumference of a circle of the same area.

When training and using object detection models, a key performance parameter to choose is that of the IoU score. The IoU score is used as a threshold value to decide whether a predicted object mask overlaps sufficiently with a ground truth mask so that the prediction can be considered a successfully "found" object. When evaluating an image, there is a list of true defect masks and predicted defect masks. To decide whether a defect has been found in the correct location, the IoU of every predicted defect is calculated for each true defect, and the defect with the highest IoU score is considered the best possible match. When this computed IoU score is above the designated threshold, this predicted defect is considered to be found. Each true defect can only be found one time, so if multiple predicted defects are found to pass the IoU threshold with a particular true defect, then the predicted defect with the highest IoU score is considered the found defect, and the other defect(s) would then be considered false positives.

In addition to the particular set of application-specific test statistics as summarized in Table S8, we performed a number of different detailed test types. A summary of the different types of tests performed, what aspects of the model or data are changed in each test, and the rationale for performing each test is provided in Table S9. These different test types, particularly assessing the effect of different train/test image splits, dataset size, and effect of ground truth labels, may serve as a basis for better understanding the successes and limitations of object detection models, especially in the context of characterizing and quantifying objects in electron microscopy images.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.xcrp.2022.100876.

ACKNOWLEDGMENTS

We would like to thank the Wisconsin Applied Computing Center (WACC) and Colin Vanden Heuvel for providing access to the CPU/GPU cluster, Euler. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant ACI-1548562. Specifically, it used the Bridges-2 system through allocation TG-DMR090023, which is supported by NSF award ACI-1928147, at the Pittsburgh Supercomputing Center (PSC).[58] Research was sponsored by the Department of Energy (DOE) Office of Nuclear Energy, Advanced Fuel Campaign of the Nuclear Technology Research and Development program (formerly the Fuel Cycle R&D Program). Neutron irradiation of FeCrAl alloys at the Oak Ridge National Laboratory's High Flux Isotope Reactor user facility was sponsored by the Scientific User Facilities Division, Office of Basic Energy Sciences, DOE. Additional support for K.G.F., D.M., and R.J. was provided by Idaho National Laboratory as part of the DOE Office of Nuclear Energy, Nuclear Materials Discovery, and Qualification Initiative (NMDQi). Additional support for R.J. was provided by the National Science Foundation (NSF) under NSF award 1931298.

AUTHOR CONTRIBUTIONS

R.J. performed the model analysis and wrote the manuscript. M.S. and Y.L. acquired and annotated the data and performed model analyses. W.H., X.L., R.H., J.R.C.G., D.W., Z.X., Z.H., and C.W. annotated the data and performed preliminary model analyses. K.G.F. and D.M. oversaw the project. All authors reviewed the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: January 19, 2022
Revised: March 24, 2022
Accepted: April 12, 2022
Published: May 3, 2022

REFERENCES

1. Seeger, A., Diehl, J., Mader, S., and Rebstock, H. (1957). Work-hardening and work-softening of face-centred cubic metal crystals. Philos. Mag. *2*, 323–350. https://doi.org/10.1080/14786435708243823.

2. Schneider, C.A., Rasband, W.S., and Eliceiri, K.W. (2012). NIH Image to ImageJ: 25 years of image analysis. Nat. Methods *9*, 671–675. https://doi.org/10.1038/nmeth.2089.

3. Jesse, S., Chi, M., Belianinov, A., Beekman, C., Kalinin, S.V., Borisevich, A.Y., and Lupini, A.R. (2016). Big data analytics for scanning transmission electron microscopy ptychography. Sci. Rep. *6*, 26348. https://doi.org/10.1038/srep26348.

4. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning (MIT Press).

5. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. *1*, 9.

6. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., and Zisserman, A. (2014). The pascal visual object Classes challenge: a retrospective. Int. J. Comput. Vis. *111*, 98–136. https://doi.org/10.1007/s11263-014-0733-5.

7. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: a large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (IEEE). https://doi.org/10.1109/CVPR.2009.5206848.

8. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. *115*, 211–252.

9. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M. (2020). Deep learning for generic object detection: a survey. Int. J. Comput. Vis. *128*, 261–318. https://doi.org/10.1007/s11263-019-01247-4.

10. DeCost, B.L., Francis, T., and Holm, E.A. (2017). Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures. Acta Mater. *133*, 30–40. https://doi.org/10.1016/j.actamat.2017.05.014.

11. Groom, D.J., Yu, K., Rasouli, S., Polarinakis, J., Bovik, A.C., and Ferreira, P.J. (2018). Automatic segmentation of inorganic nanoparticles in BF TEM micrographs. Ultramicroscopy *194*, 25–34. https://doi.org/10.1016/j.ultramic.2018.06.002.

12. DeCost, B.L., and Holm, E.A. (2015). A computer vision approach for automated analysis and classification of microstructural image data. Comput. Mater. Sci. *110*, 126–133. https://doi.org/10.1016/j.commatsci.2015.08.011.

13. Goh, G.B., Hodas, N.O., and Vishnu, A. (2017). Deep learning for computational chemistry. J. Comput. Chem. *38*, 1291–1307. https://doi.org/10.1002/jcc.24764.

14. Dimiduk, D.M., Holm, E.A., and Niezgoda, S.R. (2018). Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering. Integr. Mater. Manuf. Innov. *7*, 157–172. https://doi.org/10.1007/s40192-018-0117-8.

15. Nash, W., Drummond, T., and Birbilis, N. (2018). A review of deep learning in the study of materials degradation. Npj Mater. Degrad. *2*, 1–12. https://doi.org/10.1038/s41529-018-0058-x.

16. Agrawal, A., and Choudhary, A. (2019). Deep materials informatics: applications of deep learning in materials science. MRS Commun. *9*, 779–792. https://doi.org/10.1557/mrc.2019.73.

17. Morgan, D., and Jacobs, R. (2020). Opportunities and challenges for machine learning in materials science. Annu. Rev. Mater. Res. *50*, 71–103. https://doi.org/10.1146/annurev-matsci-070218-010015.

18. Holm, E.A., Cohn, R., Gao, N., Kitahara, A.R., Matson, T.P., Lei, B., and Yarasi, S.R. (2020). Overview: computer vision and machine learning for microstructural characterization and analysis. Metall. Mater. Trans. A. *51A*, 1–22.

19. Ge, M., Su, F., Zhao, Z., and Su, D. (2020). Deep learning analysis on microscopic imaging in materials science. Mater. Today Nano *11*, 100087. https://doi.org/10.1016/j.mtnano.2020.100087.

20. Park, C., and Ding, Y. (2019). Automating material image analysis for material discovery. MRS Commun. *9*, 545–555. https://doi.org/10.1557/mrc.2019.48.

21. Dennler, N., Foncubierta-Rodriguez, A., Neupert, T., and Sousa, M. (2021). Learning-based defect recognition for quasi-periodic HRSTEM images. Micron *146*, 103069. https://doi.org/10.1016/j.micron.2021.103069.

22. Kim, H., Inoue, J., and Kasuya, T. (2020). Unsupervised microstructure segmentation by mimicking metallurgists ' approach to pattern recognition. Sci. Rep. 1–11. https://doi.org/10.1038/s41598-020-74935-8.

23. Ziatdinov, M., Dyck, O., Maksov, A., Li, X., Sang, X., Xiao, K., Unocic, R.R., Vasudevan, R., Jesse, S., and Kalinin, S.V. (2017). Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations. ACS Nano *11*, 12742–12752. https://doi.org/10.1021/acsnano.7b07504.

24. Oktay, A.B., and Gurses, A. (2019). Automatic detection, localization and segmentation of nano-particles with deep learning in microscopy images. Micron *120*, 113–119. https://doi.org/10.1016/j.micron.2019.02.009.

25. Okunev, A.G., Mashukov, M.Y., Nartova, A.V., and Matveev, A.V. (2020). Nanoparticle recognition on scanning probe microscopy images using computer vision and deep learning. Nanomaterials *10*, 1–16. https://doi.org/10.3390/nano10071285.

26. Fu, G., Sun, P., Zhu, W., Yang, J., Cao, Y., Yang, M.Y., and Cao, Y. (2019). A deep-learning-based approach for fast and robust steel surface defects classification. Opt. Lasers Eng. *121*, 397–405. https://doi.org/10.1016/j.optlaseng.2019.05.005.

27. Liu, Y., Xu, K., and Xu, J. (2019). Periodic surface defect detection in steel plates based on deep learning. Appl. Sci. *9*, 3127. https://doi.org/10.3390/app9153127.

28. Li, W., Field, K.G., and Morgan, D. (2018). Automated defect analysis in electron microscopic images. Npj Comput. Mater. *4*, 1–9. https://doi.org/10.1038/s41524-018-0093-8.

29. Shen, M., Li, G., Wu, D., Liu, Y., Greaves, J., Hao, W., Krakauer, N.J., Krudy, L., Perez, J., Srrenivasan, V., et al. (2021). Multi defect detection and analysis of electron microscopy images with deep learning. Comput. Mater. Sci. *199*, 110576.

30. Anderson, C.M., Klein, J., Rajakumar, H., Judge, C.D., and Béland, L.K. (2020). Automated detection of helium bubbles in irradiated X-750. Ultramicroscopy *217*, 113068. https://doi.org/10.1016/j.ultramic.2020.113068.

31. Shen, M., Li, G., Wu, D., Yaguchi, Y., Haley, J.C., Field, K.G., Morgan, D., Ridge, O., and Ridge, O. (2021). A deep learning based automatic defect analysis framework for In-situ TEM ion irradiations. Comput. Mater. Sci. *197*, 110560. https://doi.org/10.1016/j.commatsci.2021.110560.

32. Roberts, G., Haile, S.Y., Sainju, R., Edwards, D.J., Hutchinson, B., and Zhu, Y. (2019). Deep learning for semantic segmentation of defects

in advanced STEM images of steels. Sci. Rep. *9*. https://doi.org/10.1038/s41598-019-49105-0.

33. Niitani, Y., Ogawa, T., Saito, S., and Saito, M. (2017). ChainerCV: a library for deep learning in computer vision. In MM '17: Proceedings of the 25th ACM International Conference on Multimedia, pp. 1217–1220.

34. Field, K.G., Hu, X., Littrell, K.C., Yamamoto, Y., and Snead, L. (2015). Radiation tolerance of neutron-irradiated model Fe-Cr-Al alloys. J. Nucl. Mater. *465*, 746–755. https://doi.org/10.1016/j.jnucmat.2015.06.023.

35. Meredig, B., Antono, E., Church, C., Hutchinson, M., Ling, J., Paradiso, S., Blaiszik, B., Foster, I., Gibbons, B., Hattrick-Simpers, J., et al. (2018). Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery. Mol. Syst. Des. Eng. *3*, 819–825. https://doi.org/10.1039/c8me00012c.

36. Lu, H.-J., Zou, N., Jacobs, R., Afflerbach, B., Lu, X.-G., and Morgan, D. (2019). Error assessment and optimal cross-validation appraoches in machine learing applied to impurity diffusion. Comput. Mater. Sci. *169*, 109075.

37. Ward, L., Keeffe, S.C.O., Stevick, J., Jelbert, G.R., Aykol, M., and Wolverton, C. (2018). A machine learning approach for engineering bulk metallic glass alloys. Acta Mater. *159*, 102–111. https://doi.org/10.1016/j.actamat.2018.08.002.

38. Zhang, D., Briggs, S.A., Edmondson, P.D., Gussev, M.N., Howard, R.H., and Field, K.G. (2019). Influence of welding and neutron irradiation on dislocation loop formation and α′ precipitation in a FeCrAl alloy. J. Nucl. Mater. *527*, 151784. https://doi.org/10.1016/j.jnucmat.2019.151784.

39. Akers, S., Kautz, E., Trevino-Gavito, A., Olszta, M., Matthews, B.E., Wang, L., Du, Y., and Spurgeon, S.R. (2021). Rapid and flexible segmentation of electron microscopy data using few-shot machine learning. Npj Comput. Mater. *7*. https://doi.org/10.1038/s41524-021-00652-z.

40. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Advances in Neural Information Processing Systems.

41. Ma, W., Kautz, E.J., Baskaran, A., Chowdhury, A., Joshi, V., Yener, B., and Lewis, D.J. (2020). Image-driven discriminative and generative machine learning algorithms for establishing microstructure-processing relationships. J. Appl. Phys. *128*, 134901. https://doi.org/10.1063/5.0013720.

42. Hsu, T., Epting, W.K., Kim, H., Abernathy, H.W., Hackett, G.A., Rollett, A.D., Salvador, P.A., and Holm, E.A. (2021). Microstructure generation via generative adversarial network for heterogeneous, topologically complex 3D materials. JOM *73*, 90–102. https://doi.org/10.1007/s11837-020-04484-y.

43. Blaiszik, B., Chard, K., Pruyne, J., Ananthakrishnan, R., Tuecke, S., and Foster, I. (2016). The materials data facility: data services to advance materials science research. JOM *68*, 2045–2052. https://doi.org/10.1007/s11837-016-2001-3.

44. Chard, R., Li, Z., Chard, K., Ward, L., Babuji, Y., Woodard, A., Tuecke, S., Blaiszik, B., Franklin, M.J., and Foster, I. (2019). DLHub: model and data serving for science. In Proceedings of 2019 IEEE International Parallel & Distributed Processing Symposium IPDPS (IEEE), pp. 283–292. https://doi.org/10.1109/IPDPS.2019.00038.

45. Chard, R., Ward, L., Li, Z., Babuji, Y., Woodard, A., Tuecke, S., Chard, K., Blaiszik, B., and Foster, I. (2019). Publishing and serving machine learning models with DLHub. ACM Int. Conf. Proceeding Ser. https://doi.org/10.1145/3332186.3332246.

46. von Chamier, L., Laine, R.F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., Lerche, M., Hernández-Pérez, S., Mattila, P.K., Karinou, E., et al. (2021). Democratising deep learning for microscopy with ZeroCostDL4Mic. Nat. Commun. *12*, 1–18. https://doi.org/10.1038/s41467-021-22518-0.

47. Zhang, C., Li, H., Wan, X., Chen, X., Yang, Z., Feng, J., and Zhang, F. (2022). TransPicker: a transformer-based framework for particle picking in cryoEM micrographs. In 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1179–1184.

48. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020). Deformable DETR: Deformable Transformers for End-To-End Object Detection. Preprint at arXiv, 1–16. https://doi.org/10.48550/arXiv.2010.04159.

49. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-End object detection with transformers. In European Conference on Computer Vision (ECCV), pp. 213–229.

50. Field, K.G., Briggs, S.A., Hu, X., Yamamoto, Y., Howard, R.H., and Sridharan, K. (2017). Heterogeneous dislocation loop formation near grain boundaries in a neutron-irradiated commercial FeCrAl alloy. J. Nucl. Mater. *483*, 54–61. https://doi.org/10.1016/j.jnucmat.2016.10.050.

51. Field, K.G., Briggs, S.A., Sridharan, K., Yamamoto, Y., and Howard, R.H. (2017). Dislocation loop formation in model FeCrAl alloys after neutron irradiation below 1 dpa. J. Nucl. Mater. *495*, 20–26. https://doi.org/10.1016/j.jnucmat.2017.07.061.

52. Dutta, A., Gupta, A., and Zisserman, A. (2020). VGG Image Annotator (VIA).

53. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2.

54. Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.

55. He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask R-CNN. Int. Conf. Comput. Vis.

56. Cai, Z., and Vasconcelos, N. (2018). Cascade R-CNN: delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern Recognition (IEEE), pp. 6154–6162.

57. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, C.L. (2014). Microsoft COCO: Common objects in context. In European Conference on Computer Vision, pp. 740–755.

58. Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G., et al. (2014). XSEDE: accelerating scientific discovery. Comput. Sci. Eng. *120*, 4–5.