

Full Length Article

Convolutional neural networks for expediting the determination of minimum volume requirements for studies of microstructurally small cracks, Part I: Model implementation and predictions

Karen J. DeMille*, Ashley D. Spear

Department of Mechanical Engineering, University of Utah, Salt Lake City, 84112, UT, USA

ARTICLE INFO

Keywords:

Microstructure
Machine learning
Computational fracture mechanics
Polycrystalline material
Representative volume element

ABSTRACT

Convolutional neural networks (CNNs) are implemented to expedite the determination of representative volume elements for microstructurally small cracks (RVE_{MSC}). By definition, RVE_{MSC} is the minimum volume of microstructure required around a microstructurally small crack (MSC) to achieve convergence of crack-front parameters with respect to volume size. In a previous study, RVE_{MSC} was determined using a computationally expensive finite-element (FE) framework involving the simulation of many microstructural instantiations. With the aim of increasing the computational efficiency of determining RVE_{MSC} , CNNs are leveraged herein to reduce the number of FE simulations required to determine RVE_{MSC} . Using data from the previous FE-based RVE_{MSC} study, CNNs are trained to predict $RVE_{MSC,ip}$ values, which quantify crack-front parameter convergence with respect to volume size for microstructural instantiation i evaluated at individual crack-front points p , given local microstructural and geometrical information. Predicted $RVE_{MSC,ip}$ values are subsequently used to estimate RVE_{MSC} values. Studies are carried out to determine the optimal amount of training data, assess CNN-based RVE_{MSC} estimation performance, and demonstrate the use of CNNs as microstructural-instantiation screening tools by enabling downselection of microstructures that are considered critical in terms of volume requirements. Individual and ensemble CNN predictions are compared. While CNNs are not found to be accurate enough to replace all FE simulations, CNNs are found to be effective as a rapid screening tool for improving the efficiency of the FE-based RVE_{MSC} determination framework and for expediting future RVE_{MSC} studies.

1. Introduction

In the context of investigations of microstructurally small cracks (MSCs), or cracks whose lengths are on the order of the predominant microstructural features of a material [1], a critical question that must be answered is: *What is the minimum microstructural volume that should be included in studies involving MSCs?* Whether through experimental characterization or computational representation, the consideration of microstructural features in studies of MSCs is critical given the strong influence of microstructural features on MSC behavior [2–6]. To clarify the requisite volume of microstructure that should be included or represented in MSC studies, DeMille and Spear [7] introduced RVE_{MSC} , or “the smallest heterogeneous volume containing an MSC such that local crack-front parameters are converged with respect to volume size”, and established the size of RVE_{MSC} for linear-elastic materials as a function of boundary condition and crack size normalized with respect to microstructure-feature size.

The concept of RVE_{MSC} is closely related to other volume elements used in analyzing heterogeneous material behavior. Other volume elements include representative volume elements (RVEs) and statistical volume elements (SVEs). An RVE is the minimum volume of heterogeneous material required to capture an average material property that does not vary with volume size [8]. Heterogeneous material volume elements smaller than RVE, whose average material properties vary with volume size, are referred to as SVEs [9]. In other words, increasing the size of a volume beyond RVE will not impact the average material properties of the volume, while increasing the size of a volume beyond an SVE size may affect the average properties of the volume. Although most often used in homogenization applications [8,10–15], RVEs and SVEs have been used in studies of local fatigue indicator parameters (FIPs) [16]; maximum principal stresses and strains [17,18]; and crack-tip opening displacements [19]. Here, we consider local J-integral values along a crack front. In this work, the minimum volume of interest is designated RVE_{MSC} because, similar to an RVE, the J-integral

* Corresponding author.

E-mail address: karen.demille@utah.edu (K.J. DeMille).

value at a given crack-front point does not vary with volume size for volumes larger than RVE_{MSC} .

In previous work by the authors [7], a finite-element (FE) framework was used to find RVE_{MSC} under assumptions of a linear-elastic constitutive model and a semi-circular surface crack. In the framework, various instantiations of idealized heterogeneous microstructures containing MSCs were simulated. Crack-front parameter values (viz., J-integrals) were then tracked as the volume of microstructure around an MSC was systematically varied. Despite important outcomes from that study, the framework proved computationally intractable for further studies due to the large number of FE simulations required. Given the assumptions of the previous RVE_{MSC} study, an expedited framework for determining RVE_{MSC} under different conditions (e.g., elastic-plastic constitutive model, non-circular internal crack) is needed to better understand minimum volume requirements for studies of MSCs.

Machine learning (ML) is a potential tool for expediting computationally expensive, simulation-based frameworks, such as the aforementioned RVE_{MSC} framework. Given sufficient data, ML models can capture complex, non-linear relationships between material features and material behavior [20]. Various studies have shown the ability of ML to predict material behavior with high accuracy in significantly less time than high-fidelity, physics-based simulations [21–23]. Some ML models, including artificial neural networks (ANNs), random forests, and Bayesian networks, take one-dimensional inputs and make one-dimensional predictions. These types of models have been used to predict stress and strain distributions [24,25], stress hotspots [26–28], effective properties [22,29], crack growth [30,31], and crystal orientation evolution [32]. Microstructural descriptors used as input to the ML models included shape encodings [24], grain shape and size [22, 26–28], and volume-fraction porosity and average pore sizes [25]. Spatial relationships between microstructural features were input to the ML models in various ways. Mangal and Holm [26] encoded spatial relationships of grains through neighborhood features, including distances to special points and the number of nearest neighbors. Pathan et al. [29] used two-point correlations and principal component analysis (PCA) to describe the spatial relationships between fibers in fiber-reinforced composites. Pandey et al. [32] accounted for the spatial relationships between voxels by providing grain orientations for surrounding voxels alongside the orientation of the voxel of interest to their model. However, manually encoding spatial relationships between many microstructural features as single-dimensional inputs is a complex, non-trivial task.

Convolutional neural networks (CNNs) offer an alternative to manually encoding spatial relationships between material features for ML model input [22,23,33–35]. By design, CNNs take in multidimensional arrays of data, detect hierarchical patterns of features, extract single-dimensional descriptors of the input, and make a prediction [36,37]. Cang and Ren [33] and Lubbers et al. [38] extracted low-dimensional representations of heterogeneous materials using CNNs. In multiple studies [22,23,39–43], CNNs were trained to predict effective properties of various materials given images of heterogeneous microstructures. Other studies [34,44,45] predicted stress or strain fields from images of microstructures. Pierson et al. [46] predicted the path of a crack through a heterogeneous microstructure, and Kantzos et al. [47] predicted stress concentrations from surface height maps. Given the abilities of CNNs to rapidly predict material behavior from 3D microstructural data, it is hypothesized that the bottleneck in the original RVE_{MSC} framework due to a large number of required FE simulations can be addressed by introducing CNNs into the determination of RVE_{MSC} .

The objective of this work is to demonstrate the potential of CNNs to expedite a computationally expensive, simulation-based study of material behavior, specifically in the context of establishing RVE_{MSC} . Given the computational expense associated with obtaining RVE_{MSC} data, the data generated in the previous RVE_{MSC} study [7] is harnessed

for this work. A task-specific microstructural sampling strategy, involving 3D grids of microstructural features ahead of crack-front points, is selected to quantify the MSC-containing microstructures. The amount of training data provided to the CNN is varied to establish an optimal amount of training data. Subsequently, ensemble CNN performance is compared with individual CNN performance. CNN performance is evaluated on two tasks: directly estimating RVE_{MSC} values from CNN predictions and screening microstructural instantiations prior to FE simulation. Although this work focuses on using CNNs to expedite the determination of minimum volume requirements for studies of MSCs, the methods and findings from the work could be applicable to other computational materials science applications involving computationally expensive simulations.

2. Methods

2.1. Previous work: finite-element models

All of the data used in this work come from FE simulations performed with Abaqus 6.14 [48] for the original RVE_{MSC} determination framework [7]. In the simulations, the volume of heterogeneous material surrounding an MSC was varied systematically while crack-front parameters (viz., J-integral values) were tracked along the crack front. Details of the FE simulations reported in previous work are provided next for completeness.

The FE models consisted of idealized, heterogeneous microstructures, each containing a semi-circular surface crack. The microstructures comprised cube-shaped grains whose elastic moduli (E) values were varied to implicitly represent different grain orientations. As shown in Fig. 1, the cube-shaped grains had side lengths of g , which was arbitrarily defined given that the constitutive model did not account for grain size. The crack had a half-length of a . Each grain was assigned an elastic modulus between 75 and 225 GPa and a Poisson's ratio of $\nu = 0.32$. The microstructural volume surrounding the crack was parameterized by d_1 , the minimum distance from the crack-front to the sides of the volume in the crack plane, and d_2 , the distance from the crack plane to the lower and upper surfaces. For the remainder of the paper, d_1 and d_2 will be expressed in terms of the number of grains, i.e., the physical distance normalized by the grain size, g .

Cracked microstructural instantiations were created for each of four normalized crack sizes ($a/g = 0.25, 0.45, 1.0$, and 3.0). For each crack size, 22 different microstructural instantiations were generated, resulting in 88 unique cracked microstructural instantiations. Most instantiations were generated by randomly selecting an elastic modulus for each grain. Fig. 1 shows an example of an instantiation, in which colors reflect the elastic moduli of grains. In some instantiations, select grains along the crack front were strategically assigned either the maximum or minimum elastic modulus. For each crack size, one microstructural instantiation had all grains assigned a uniform value of $E = 138$ GPa.

The FE models were analyzed under two different sets of boundary conditions: full submodeling or free sides. Both of these boundary conditions incorporated a submodeling approach, where the displacements applied to the boundaries of the microstructure were interpolated from displacement results of a global (homogeneous) FE model. The global FE model, shown in Fig. 2a, represents the gauge section of a tensile specimen loaded to 800 MPa in the y -direction. The global model had dimensions of $25\text{ mm} \times 10\text{ mm} \times 2.1\text{ mm}$, an elastic modulus of $E = 138$ GPa, and a Poisson's ratio of $\nu = 0.32$. For the full-submodeling boundary condition, x -, y -, and z -displacements were applied to all faces of the microstructural model except the surface intersecting the crack, as shown in green in Fig. 2b. This boundary condition is intended to represent the behavior of an MSC embedded within a large volume of material. For the free-sides boundary condition, y -displacements were applied to only the top and bottom faces of the microstructural model, as shown in green in Fig. 2c. This boundary condition is

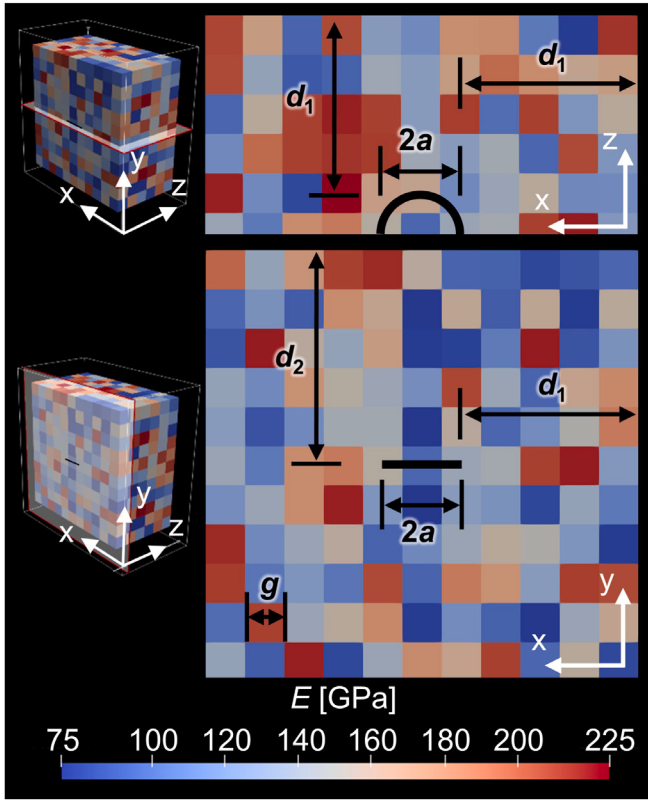


Fig. 1. Idealized, heterogeneous microstructural FE models used in the determination of RVE_{MSC} . A semi-circular surface crack of half-length a is embedded in a microstructure consisting of cube-shaped grains with side length g . Different elastic moduli are assigned to each grain. Two parameters, d_1 and d_2 , are used to define the volume of microstructure around the crack.

intended to represent the response of an MSC embedded in a very narrow specimen, such as the matchstick specimens often used in X-ray tomography experiments [49–52]. In total, 176 unique combinations of microstructural instantiation, crack size, and boundary condition were simulated. Results from the FE simulations in Ref. [7] showed that volume requirements ranged from 4 to 3439 grains (for a 5% tolerance), depending on material constraint from boundary condition, microstructure arrangement, and crack size.

2.2. Convolutional neural networks

To expedite the determination of RVE_{MSC} , CNNs are implemented in this work to predict (or estimate) volume requirements for J-integral convergence with respect to volume size. Volume requirements are considered at three levels: individual crack-front points, individual microstructural instantiations, and across all crack-front points in all microstructural instantiations. The CNNs are trained to predict $RVE_{MSC,ip}$, or the minimum volume required for a crack-front parameter (viz., J-integral) to converge with respect to volume size at crack-front point p in microstructural instantiation i . Predictions of $RVE_{MSC,ip}$ are subsequently used to estimate $RVE_{MSC,i}$, or the minimum volume for which J-integral values are converged with respect to volume size at all crack-front points (except those at grain boundaries¹) in microstructural instantiation i . Finally, estimates of $RVE_{MSC,i}$ are used to estimate RVE_{MSC} , or the minimum volume of microstructure required around an MSC to achieve J-integral convergence with respect to volume size at all crack-front points in all microstructural instantiations.

¹ Spurious J-integral values occur at grain boundaries.

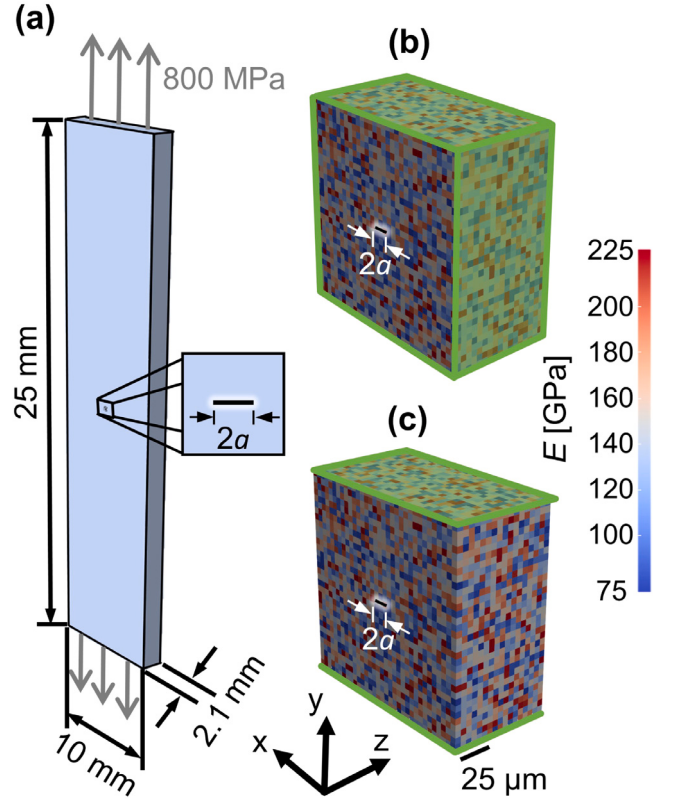


Fig. 2. Boundary conditions used in FE simulations for determining RVE_{MSC} [7]: (a) global model used to compute displacements for boundary conditions of the microstructure models; (b) full-submodeling and (c) free-sides boundary conditions, where surfaces highlighted in green have applied displacements interpolated from the global model.

A total of four CNNs (all having the same architecture) are used to predict one of two $RVE_{MSC,ip}$ parameters ($d_{1,ip}$ or $d_{2,ip}$) under one of two boundary conditions. All data for the CNNs (both input and target data) are obtained from the FE simulations completed during the previous RVE_{MSC} determination study [7]. The following sections describe the CNN inputs, targets, architecture, and data.

2.2.1. CNN inputs

For a given crack-front point p , the inputs to the CNN model consist of two 3D arrays and two scalar values. The 3D arrays are sampled from a regular grid of $N \times N \times N$ points located at p such that the grid is tangent to the crack front, as shown in Fig. 3. The size, location (relative to the crack-front point), and dimensions of the sampling grids used to generate the 3D array inputs for the CNN models are specified by three parameters: the distance ahead of the crack-front point (d_{ahead}), the distance behind the crack-front point (d_{behind}), and the number of grid points in each direction (N). From these grid parameters, the distance between grid points can be determined using the equation

$$\Delta_{grid} = \frac{d_{ahead} + d_{behind}}{N - 1}. \quad (1)$$

The same grid spacing is used in each of the three directions. The grid parameters used for this work are provided in Table 1.

The input arrays provide the CNN with a 3D “image” of the local microstructure in the neighborhood of each crack-front point. The use of a local microstructure sampling method is similar to that used by Yang et al. [34]. In this work, the sampling grids are designed to probe grains primarily ahead of the crack front, as Rovinelli et al. [53] showed that clusters of grains ahead of a crack tip influence short crack behavior; hence $d_{ahead} > d_{behind}$. The selected values of d_{ahead} and d_{behind}

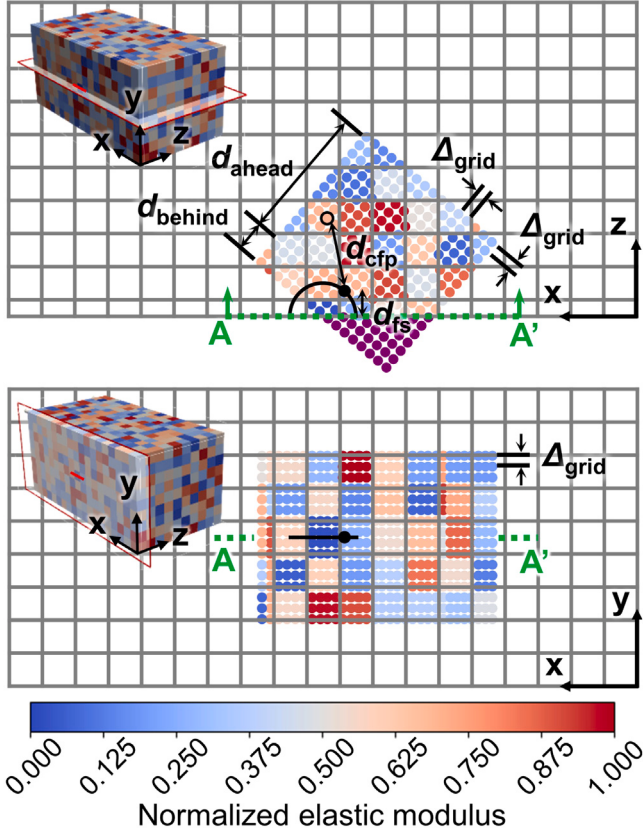


Fig. 3. Sampling strategy used to extract microstructural information from around a crack-front point p . The top figure shows a top-down view of the sampling grid, while the bottom figure shows a front view of the sampling grid. The $16 \times 16 \times 16$ grid is defined using parameters $d_{\text{ahead}} = 4$ grains, $d_{\text{behind}} = 0.95$ grains, and $\Delta_{\text{grid}} = 0.33$ grains. Grid points are colored according to the normalized elastic modulus. Purple points represent grid points located outside of the physical volume.

Table 1

Parameters used to define the sampling grids used to extract data for the 3D array inputs of the CNNs. Sampling grids are placed tangent to the crack front at each crack-front point, as shown in Fig. 3.

Grid parameter	Value
d_{ahead}	4 grain widths
d_{behind}	0.95 grain widths
N	16
Δ_{grid}	0.33 grain widths

result in grids that include slightly more grains than the number of nearest-neighbor grains shown to have the strongest influence on local crack-front parameter convergence behavior; namely, the previous RVE_{MSC} study showed that the second and third nearest-neighbor grains ahead of a crack front have the greatest impact on the local crack-front parameter convergence rates [7]. In other studies, local FIPs or mechanical fields in a given grain were found to be most sensitive to the first through third nearest-neighbor grains [16,18,54].

Input to the CNN model for crack-front point p includes both microstructural and geometrical features. Elastic moduli and Euclidean distances between grid points and crack-front point p sampled from the grid comprise the E and distance to crack front (d_{cfp}) input arrays, respectively. If a grid point falls outside of the physical volume of the FE model, then a value of -1 is assigned to the corresponding array component. The scalar inputs are the normalized crack size (a/g) and the distance from the crack-front point to the free surface (d_{fs}), which is simply the z -coordinate of the crack-front point (Fig. 3). The inclusion of d_{fs} as an input is intended to inform the CNN of the degree of

material constraint (ranging from plane-stress to plane-strain) acting on crack-front point p , which may influence the convergence trends of crack-front parameters [55]. To improve CNN model performance, the CNN inputs are normalized. The components of 3D array inputs are normalized to range from 0 to 1 for grid points inside the FE model and set to -1 for grid points outside the FE model. The scalar inputs are normalized to range from 0 to 1.

2.2.2. CNN targets

For a given crack-front point p , the targets of the CNN models consist of $\text{RVE}_{\text{MSC},ip}$ parameter values, $d_{1,ip}$ and $d_{2,ip}$. A total of four different CNNs are used in predicting the $\text{RVE}_{\text{MSC},ip}$ parameter values. A separate CNN is trained for each combination of boundary condition (full submodeling or free sides) and $\text{RVE}_{\text{MSC},ip}$ parameter ($d_{1,ip}$ or $d_{2,ip}$). Separate CNN models for different combinations of boundary condition and $\text{RVE}_{\text{MSC},ip}$ parameter reflect the approach used by Rovinelli et al. [30,31], where separate ML models were used to predict crack-growth direction and rate.

Fig. 4a and b illustrate the process of finding $d_{1,ip}$ and $d_{2,ip}$, respectively, for microstructure i at crack-front point p under a given boundary condition from previously completed FE simulations [7]. In finding $d_{1,ip}$ (Fig. 4a), d_1 is varied, while d_2 is fixed at $d_2 = S$, where $S \gg d_{2,\text{MSC}}$. In Fig. 4a, $S = 16$ grains. For each crack-front point, or node along the crack-front, a coarsely-sampled set of J-integral values is extracted from FE simulation results for volumes with d_1 values incremented in steps of 2 grains ($d_1 = \{2, 4, 6, 8, \dots, 24\}$ grains). The coarsely-sampled J-integral values are shown as circles in the plot of Fig. 4a. Bounds of $\pm 5\%$ are established based on the J-integral value corresponding to the largest volume (e.g., $d_1 = 24$ grains in Fig. 4a). The largest d_1 value whose corresponding J-integral value falls outside of the $\pm 5\%$ bounds is identified as the divergence point (e.g., $d_1 = 2$ grains in Fig. 4a). The set of sampled J-integral values is locally refined around the divergence point, as J-integral values are extracted from FE simulation results for volumes with d_1 values incremented in steps of 0.5 grains between the divergence point and the next-largest volume ($d_1 = \{2.5, 3, 3.5\}$ grains in Fig. 4a). The locally refined J-integral values are shown as crosses in the plot of Fig. 4a. An updated divergence point is identified ($d_1 = 2.5$ grains), and the volume immediately to the right of this volume is identified as $d_{1,ip}$ ($d_{1,ip} = 3$ grains). A similar process is used to determine $d_{2,ip}$ (Fig. 4b). In the case of determining $d_{2,ip}$, d_2 is varied and d_1 is fixed at $d_1 = T$, where $T \gg d_{1,\text{MSC}}$ ($T = 14$ grains² in Fig. 4b). Note that in Fig. 4b, the divergence point for the coarsely sampled J-integral value set is 0 grains since no coarsely sampled J-integral values fall outside of the $\pm 5\%$ convergence bounds. Given that d_1 and d_2 values are incremented in minimum steps of 0.5 grains, $d_{1,ip}$ and $d_{2,ip}$ are determined with resolutions of 0.5 grains. To improve CNN model performance, the targets for the CNNs are normalized to a range of 0 to 1.

2.2.3. CNN-based estimates of $\text{RVE}_{\text{MSC},i}$ and RVE_{MSC}

Predictions of $\text{RVE}_{\text{MSC},ip}$ parameter values are used to make CNN-based estimates of $\text{RVE}_{\text{MSC},i}$ and RVE_{MSC} . The limiting (or maximum) $\text{RVE}_{\text{MSC},ip}$ parameter values define the $\text{RVE}_{\text{MSC},i}$ parameter values for a given microstructure:

$$d_{1,i} = \max_p(d_{1,ip}) \quad (2)$$

$$d_{2,i} = \max_p(d_{2,ip}). \quad (3)$$

² Selected values of S and T vary based on crack size and boundary condition due to variations in $d_{2,\text{MSC}}$ and $d_{1,\text{MSC}}$ for different crack sizes and boundary conditions.

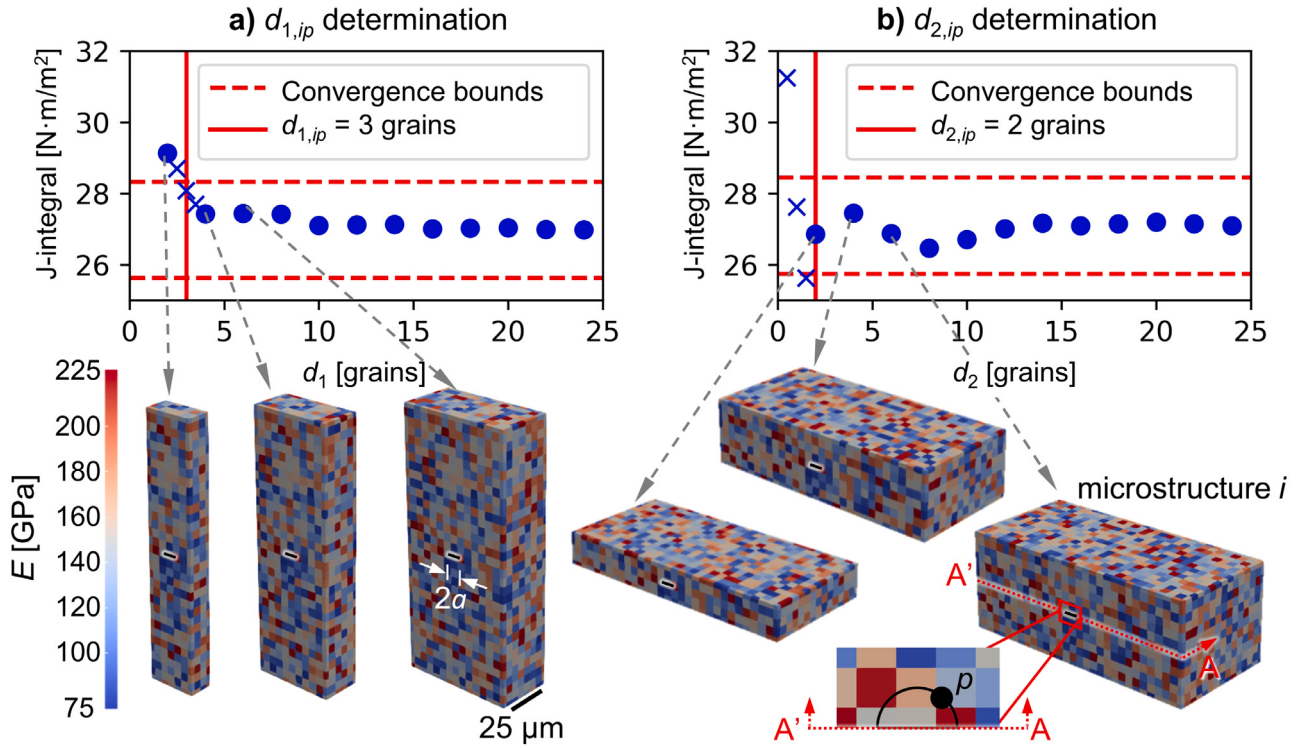


Fig. 4. Procedure used in determining (a) $d_{1,ip}$ and (b) $d_{2,ip}$ via FE simulations [7]. To determine $d_{1,ip}$ or $d_{2,ip}$, respectively, the volume of microstructure around a crack is varied in d_1 or d_2 while tracking the convergence of J-integral values with respect to volume size at crack-front point p in microstructure i .

Given that RVE_{MSC} requires J-integral values to be converged at all crack-front points in all microstructural instantiations, RVE_{MSC} is determined by taking the maximum $RVE_{MSC,i}$ parameter values among all microstructures:

$$d_{1,MSC} = \max_i(d_{1,i}) \quad (4)$$

$$d_{2,MSC} = \max_i(d_{2,i}). \quad (5)$$

2.2.4. CNN architecture

The 3D CNN architecture used in each of the four CNN models is based on the VGG-16 CNN model [56] and is shown in Fig. 5. Fig. 5 specifies the number of filters, convolutional filter sizes, and activation function for each convolutional layer. For each maximum pooling layer, the pooling window size is shown. All of the convolutional layers and maximum pooling layers use strides of $1 \times 1 \times 1$. Same padding is used for each convolutional layer, while valid padding is used for each maximum pooling layer. Following the third maximum pooling layer, the layer output is flattened in preparation for the fully connected layers. Fig. 5 specifies the number of output units and activation functions for the fully connected layers. Max-norm regularization [57] is applied to each convolutional filter, fully connected weight vector, and bias vector with a weight constraint of $c = 4$. Dropout layers [57] are placed prior to both fully connected layers and retain layer units with a probability of 50%. The CNN models are built using Keras 2.3 [58] and TensorFlow 1.14 [59]. The architecture is selected using Hyperas [60] and a greedy optimization approach.

The CNNs are trained using the ADAM optimizer [61] and mean square error (MSE) loss. A mini-batch size of 64 and an ADAM learning rate of 0.001 are used. Training is stopped once the validation loss has failed to decrease over ten epochs. The model weights corresponding to the lowest validation loss are restored for the final trained model.

2.2.5. CNN data overview and splitting

Recall, previous FE simulations [7] provide data for the CNNs; in the FE simulations, four different crack sizes ($a/g = 0.25, 0.45, 1.0$, and 3.0) and 22 microstructural instantiations per crack size were considered for each combination of boundary condition and $RVE_{MSC,ip}$ parameter. Thus, each of the four CNN models has 88 crack-front-point (CFP) groups of $d_{1,ip}$ or $d_{2,ip}$ values associated with it. Each CFP group contains $d_{1,ip}$ or $d_{2,ip}$ values for all crack-front points (except points that lie on grain boundaries) in a given microstructural instantiation containing an MSC of size a/g . Each crack contains between 91 and 269 crack-front points, depending on the crack size. By considering J-integral convergence at individual crack-front points, 88 FE simulation sets are effectively expanded to a total of 14 168 available data points per CNN model.

In the studies discussed in Section 2.3, the 14 168 data points are split into training, validation, and testing sets. During the splitting, data points are split such that all crack-front points from a given CFP group are kept together. This splitting approach ensures that similar data from neighboring crack-front points in a given CFP group do not introduce data duplicity between the training, validation, and testing sets.

2.3. CNN evaluation studies

Four studies are performed to evaluate the performance of the CNN models in expediting the determination of RVE_{MSC} . First, the optimal amount of training data for the CNN models is found. Second, individual CNN models are evaluated on their performance in predicting $RVE_{MSC,ip}$ and making estimates of $RVE_{MSC,i}$ and RVE_{MSC} . Third, individual CNN model performance is compared to ensemble CNN model performance in predicting $RVE_{MSC,ip}$ and making estimates of $RVE_{MSC,i}$ and RVE_{MSC} . Fourth, CNN models are evaluated on their ability to screen for cracked microstructural instantiations that have large volume requirements.

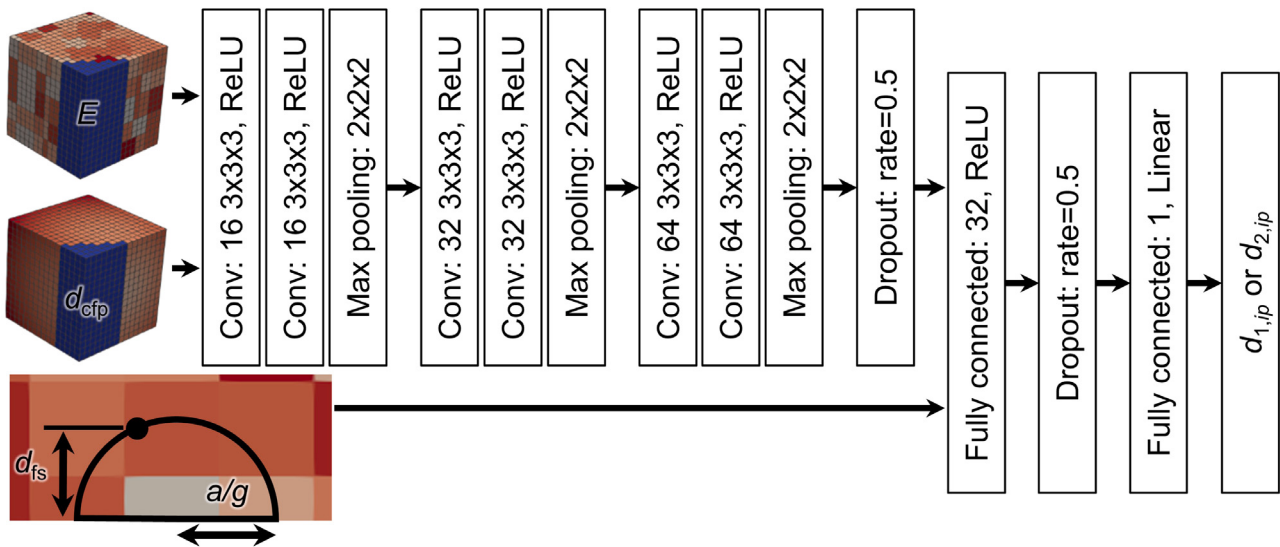


Fig. 5. The 3D CNN architecture, based on VGG-16 [56], used to predict $RVE_{MSC,ip}$ parameter values. Convolutional filter sizes, maximum pooling window sizes, fully connected layer units, dropout rates, and activation functions are shown. In addition to model specifications shown, strides of $1 \times 1 \times 1$ and max-norm regularization are used.

2.3.1. Identification of optimal amount of training data

The first study aims to identify the optimal amount of training data for the CNN models. In this case, the optimal amount of training data represents the number of CFP groups that balances model performance and computational cost. To identify the optimal amount of training data, the performance of the CNN models on a constant test set is tracked as the amount of training data is systematically increased. Seven cases of increasing training/validation data, shown in Table 2, are compared. The following process is repeated for each of the four CNN models. First, approximately 20% (or 18) of 88 CFP groups are set aside as a test set. Recalling that each CFP group contains between 91 and 269 data points (Section 2.2.5), the test set contains ~2900 data points. From the remaining 70 CFP groups, 80% (56 CFP groups) and 20% (14 CFP groups) are placed in training and validation data pools, respectively. The training and validation data pools contain ~9000 and ~2300 data points, respectively. For each of the seven training/validation data amounts in Table 2, training and validation CFP groups are randomly selected from the training and validation data pools. The model is trained with the selected training/validation sets, then evaluated on the test set. Given that CNN performance varies from one training run to another, the training and testing process is repeated ten times for each amount of training data to capture the variability in CNN performance. For the ten repetitions, the test set remains fixed, while the training and validation data sets are randomly drawn during each repetition.

The variation of training and validation data amounts is repeated for a total of five random test sets. From ten training repetitions for five test sets, CNN performance is evaluated 50 different times for each amount of training data with three sources of variation: the random selection of training/validation data, the random selection of test data, and the random initialization of model weights.

2.3.2. Individual CNN model performance evaluation

The second study seeks to assess the performance of an individual CNN model in predicting $RVE_{MSC,ip}$ and, hence, estimating $RVE_{MSC,i}$ and RVE_{MSC} parameter values. A repeated five-fold cross-validation [62] is performed to: first, evaluate $RVE_{MSC,ip}$ predictions; second, evaluate $RVE_{MSC,i}$ parameter estimates among all microstructural instantiations; and third, evaluate RVE_{MSC} parameter estimates. To perform a single five-fold cross-validation for a given combination of boundary condition and $RVE_{MSC,ip}$ parameter type, the 88 CFP groups are randomly split into five subsets containing 17, 17, 17, 17, and 20 groups, respectively. In each of five training folds, one of the five subsets is

Table 2

Training and validation data set sizes compared during the identification of the optimal amount of training data. Each CFP group contains data for all crack-front points in a given microstructural instantiation i containing a crack of size a/g .

Training data case	Number of CFP groups (Approx. number of data points)		
	Training	Validation	Test
1	8 (~1300)	2 (~300)	18 (~2900)
2	16 (~2600)	4 (~600)	18 (~2900)
3	24 (~3900)	6 (~1000)	18 (~2900)
4	32 (~5200)	8 (~1300)	18 (~2900)
5	40 (~6400)	10 (~1600)	18 (~2900)
6	48 (~7700)	12 (~1900)	18 (~2900)
7	56 (~9000)	14 (~2300)	18 (~2900)

reserved for testing. From the remaining four subsets, 50 CFP groups (the optimal number identified from the previous study, as presented later in Section 3.1) are randomly selected for training and validation data. The training and validation data are used to train the CNN. The trained CNN then predicts $RVE_{MSC,ip}$ parameter values for the test set. During the five-fold cross-validation, each subset of data is held out as the test set exactly once, resulting in one prediction of $RVE_{MSC,ip}$ for each crack-front point. The $RVE_{MSC,ip}$ predictions are used with Eqs. (2) and (3) to estimate $RVE_{MSC,i}$ parameters. With $RVE_{MSC,i}$ parameter estimates for all microstructural instantiations, Eqs. (4) and (5) are used to estimate RVE_{MSC} parameters. To improve the accuracy of the model performance evaluation, the five-fold cross-validation process described above is completed a total of five times.

2.3.3. Ensemble CNN model performance evaluation

The third study compares the performance of individual CNN predictions with ensemble CNN predictions. A modified five-fold cross-validation approach is performed to evaluate ensemble predictions of $RVE_{MSC,ip}$ and estimates of $RVE_{MSC,i}$ and RVE_{MSC} . The same cross-validation subsets from Section 2.3.2 are used. During one modified cross-validation, each of the five cross-validation subsets is held out as a test set. For each test set, ten individual CNN model $RVE_{MSC,ip}$ predictions are used to find ensemble model $RVE_{MSC,ip}$ predictions. First, 50 CFP groups (the optimal number identified in Section 3.1) are randomly selected from all data not in the test set. For each of the ten CNN model predictions, the training/validation data are split randomly into training and validation sets consisting of 80% and 20% of the training/validation data, respectively. Using these training and

validation sets, the CNN model is trained, and $d_{1,ip}$ or $d_{2,ip}$ predictions for the test set are recorded. The previously selected training/validation data are then shuffled randomly into new training and validation sets (maintaining the 80% - 20% split) and the CNN training process is repeated. The shuffling of training and validation data incorporates data diversity into the ensemble model [63]. After ten different prediction sets have been obtained, ensemble model predictions are made by averaging the ten different $d_{1,ip}$ or $d_{2,ip}$ predictions for each crack-front point [64]. $RVE_{MSC,i}$ and RVE_{MSC} estimates are found with Eqs. (2), (3), (4), and (5). As in Section 2.3.2, the modified five-fold cross-validation is completed a total of five times.

2.3.4. CNN-based microstructure screening performance evaluation

The goal of the fourth study is to test the ability of the CNN models to identify cracked microstructural instantiations with large $RVE_{MSC,i}$ values (i.e., large volume requirements to guarantee convergence of crack-front parameters). For each combination of boundary condition and $RVE_{MSC,i}$ parameter ($d_{1,i}$ or $d_{2,i}$), ensemble $RVE_{MSC,ip}$ predictions (Section 2.3.3) are used to screen instantiations for those with large $RVE_{MSC,i}$ values. First, all 88 cracked microstructural instantiations are labeled as having large, medium, or small $RVE_{MSC,i}$ parameter values. For a given crack size, the range of all actual $RVE_{MSC,i}$ parameter values is divided equally into three bins: small (bottom 33% of range), medium (middle 33% of range), and large (top 33% of range) volume requirements, as shown in Fig. 6. Each cracked microstructural instantiation is labeled according to its actual $d_{1,i}$ or $d_{2,i}$ values. Three test sets are selected, with each test set consisting of a small-, medium-, and large-volume-requirement instantiation for each crack size ($a/g = 0.25, 0.45, 1.0$, and 3.0). In other words, each test set contains data points from 12 of the 88 CFP groups (1932 of 14 168 data points) associated with the given combination of boundary conditions and $RVE_{MSC,i}$ parameter.

Ensemble CNN-based $RVE_{MSC,i}$ estimates (Section 2.3.3) are made for each of the three test sets. Fifty CFP groups (the optimal number identified in Section 3.1) drawn randomly from all non-test CFP groups are selected as training/validation data. Ten different models are trained using different random splits of the training/validation data into training (40 CFP groups) and validation (10 CFP groups) sets. Ensemble $d_{1,ip}$ or $d_{2,ip}$ predictions are made by averaging the predictions of the ten individual models. Estimates of $d_{1,i}$ or $d_{2,i}$ values are determined for each instantiation (or CFP group) using Eqs. (2) and (3). The estimated $d_{1,i}$ or $d_{2,i}$ values derived from CNN predictions of $d_{1,ip}$ and $d_{2,ip}$ are used to label instantiations as having large, medium, or small predicted volume requirements using the same binning strategy used for the actual (ground-truth) results. For the labeling based on estimated $RVE_{MSC,i}$ parameters, the volume requirement bin boundaries are adjusted to reflect the range of $RVE_{MSC,i}$ parameter values seen in only the training/validation data set (all instantiations are considered in the labeling when selecting test sets).

3. Results

3.1. Optimal amount of training data

Fig. 7 shows the variation of coefficient of determination (R^2) as the amount of training/validation data is increased. Results are shown for the four CNN model types corresponding to different combinations of boundary condition and $RVE_{MSC,ip}$ parameter type discussed in Section 2.2. Each blue point represents the performance of one of 50 different CNN model runs described in Section 2.3.1. The black dotted line represents the trend of the average R^2 value of the 50 model runs. Considering the plateau in average R^2 that begins to develop between 40 and 70 CFP training/validation groups, the optimal amount of training and validation data for predicting $RVE_{MSC,ip}$ is identified as 50 CFP groups. Further discussion is given in Section 4.1.

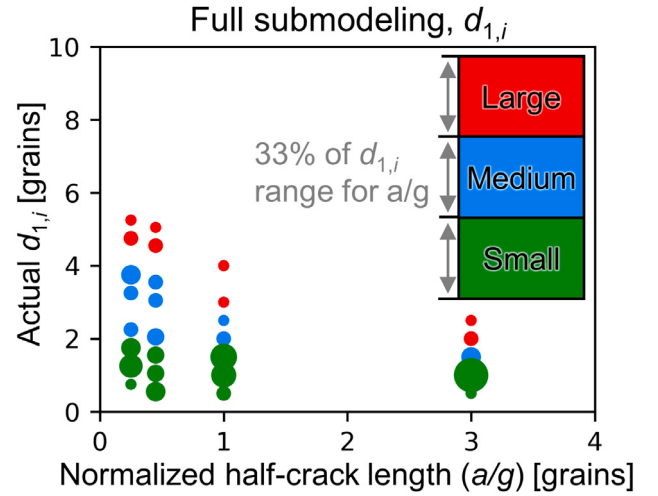


Fig. 6. Binning strategy used in labeling microstructural instantiations as having large, medium, or small volume requirements. The range of $d_{1,i}$ or $d_{2,i}$ values for each crack size is split into three bins, each containing 33% of the total range. Bins for $d_{1,i}$ and the full-submodeling boundary condition are shown. Each point represents one or more microstructural instantiations, with the size of the points reflecting the number of instantiations for which the $d_{1,i}$ value is repeated.

3.2. CNN model regression performance

Metrics for CNN model performance in predicting and estimating $RVE_{MSC,ip}$ and $RVE_{MSC,i}$ parameter values, respectively, are shown in Table 3. Repeated cross-validation predictions (Section 2.3.2) are made using one CNN model, while ensemble predictions of $RVE_{MSC,ip}$ parameters (Section 2.3.3) are equal to the average prediction from ten CNN models. Estimates of $RVE_{MSC,i}$ parameters are derived from predicted $RVE_{MSC,ip}$ parameters using Eqs. (2) and (3). The R^2 and root mean square error (RMSE) scores in Table 3 are equal to the average of 25 cross-validation training runs (five-fold cross-validation repeated five times) for each of the four models discussed in Section 2.2.

Fig. 8 shows $RVE_{MSC,i}$ parameter values determined based on ensemble predictions of $RVE_{MSC,ip}$ parameters for one of the modified cross-validations described in Section 2.3.3. For each of the two boundary conditions, $d_{1,i}$ and $d_{2,i}$ values for 22 microstructural instantiations are shown for normalized crack sizes of $a/g = 0.25, 0.45, 1.0$, and 3.0 . The actual $RVE_{MSC,i}$ parameter values from the previous FE simulations are shown as blue points. The CNN-estimated $RVE_{MSC,i}$ parameter values are shown in red. The sizes of the points reflect the number of times that a given value is repeated in the data set for the normalized crack size.

Fig. 9 provides an example of comparisons between actual and CNN-derived estimates of RVE_{MSC} parameter values for one data split. Results for crack sizes of $a/g = 0.25, 0.45, 1.0$, and 3.0 for free-sides and full-submodeling boundary conditions are shown in Figs. 9a,c and b,d, respectively. The actual RVE_{MSC} parameter values are shown as black circles with ± 1 grain error bars. Estimates of RVE_{MSC} parameter values from ten individual CNN model runs are shown as gray crosses. Ensemble RVE_{MSC} parameter value estimates, obtained as described in Section 2.3.3, are shown as red squares.

3.3. Microstructure screening via CNN

Table 4 provides the performance metrics for assessing the CNN's ability to detect microstructural instantiations with large volume requirements. Microstructure screening metrics are shown for three different test sets. For large/medium/small volume-requirement binning, accuracies presented represent the percent of instantiations whose actual and CNN-estimated $RVE_{MSC,i}$ parameter values fall within the

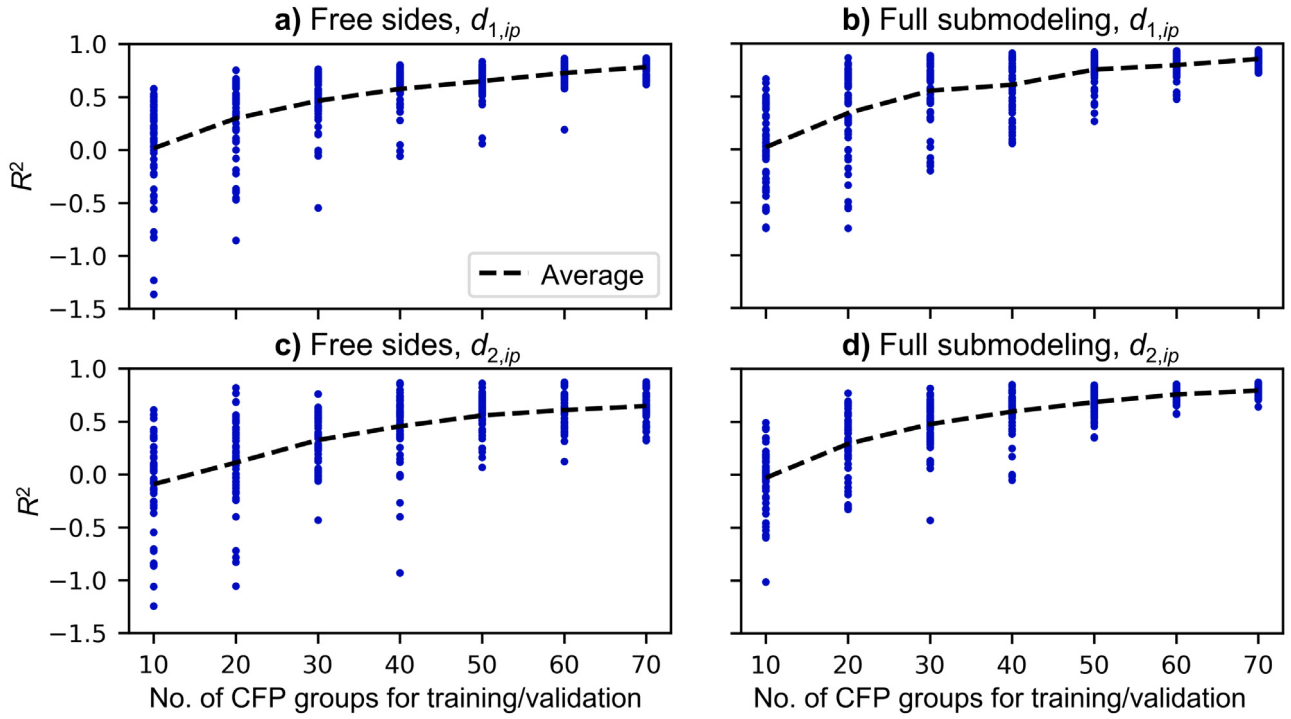


Fig. 7. Variation of CNN model performance as the amount of training and validation data provided to the CNN model is increased. Each CFP group comprises between 91 and 269 distinct data points. Results are shown for four different types of CNN models trained to predict a specific $RVE_{MSC,ip}$ parameter under a particular type of boundary condition: (a) $d_{1,ip}$ with free sides, (b) $d_{1,ip}$ with full submodeling, (c) $d_{2,ip}$ with free sides, and (d) $d_{2,ip}$ with full submodeling.

Table 3

Average R^2 and RMSE values for $RVE_{MSC,ip}$ and $RVE_{MSC,i}$ parameter predictions and CNN-based estimates, respectively, from repeated cross-validations. The CNN models are trained using the optimal training data amount. The repeated cross-validation uses individual CNN model predictions (Section 2.3.2), while the ensemble cross-validation uses ensemble model predictions (Section 2.3.3).

Model type	Individual CNN cross-validation				Ensemble CNN cross-validation			
	$RVE_{MSC,ip}$		$RVE_{MSC,i}$		$RVE_{MSC,ip}$		$RVE_{MSC,i}$	
	R^2	RMSE [grains]	R^2	RMSE [grains]	R^2	RMSE [grains]	R^2	RMSE [grains]
Free sides, d_1	0.64	1.11	0.65	1.23	0.70	1.02	0.71	1.11
Free sides, d_2	0.62	1.31	0.45	1.43	0.67	1.21	0.52	1.34
Full submodeling, d_1	0.83	0.55	0.79	0.55	0.82	0.57	0.80	0.54
Full submodeling, d_2	0.68	1.20	0.59	1.21	0.73	1.09	0.67	1.08

same bin. Accuracy, sensitivity, and precision for labeling large and not large (i.e., small or medium) volume requirements are also shown. The large/not large accuracies represent the percent of instantiations whose actual and CNN-estimated $RVE_{MSC,i}$ parameter values fall within the same bin. The sensitivity represents the percent of instantiations with *actual* large $RVE_{MSC,i}$ parameter values whose CNN-estimated $RVE_{MSC,i}$ parameter values are also in the large bin range. The precision represents the percent of instantiations with *CNN-estimated* $RVE_{MSC,i}$ parameter values in the large volume bin whose actual $RVE_{MSC,i}$ parameter values are also in the large volume bin.

3.4. Computational cost of CNNs

The computational cost of using CNNs comes from two sources: a one-time training cost and a forward prediction cost. Each CNN model takes between 3 min (2 CPU hours) and 17 min (10 CPU hours) to train on 32 cores. The training time accounts for the bulk of the computational cost, as it takes less than 1 s to predict $RVE_{MSC,ip}$ parameter values for all crack-front points in a microstructural instantiation.

4. Discussion

4.1. Optimal amount of training data

The comparisons of model performance as increasing amounts of training and validation data are provided to the CNN models are used to determine the amount of data that best balances between maximizing CNN performance and minimizing the expense of obtaining training data. These comparisons are shown in Fig. 7. Two general trends are observed in the performance comparison plots: a decrease in the scatter of R^2 and an increase in average R^2 as the amount of training/validation data increases. Similar trends were found in studies on training set size by Kim [65] and Cho et al. [66]. These observations demonstrate that the model is making better, more consistent predictions given more data. However, for each cracked microstructural instantiation, at least 40 CPU hours worth of FE simulations are required to determine either $d_{1,ip}$ or $d_{2,ip}$ values for the crack-front points in the instantiation. Fig. 7 is used to determine when adding additional training/validation data is worth the expense. The increasing model performance with increasing data is much more significant when the amount of training/validation data is smaller, so plateaus in scatter and average R^2 are targeted. The

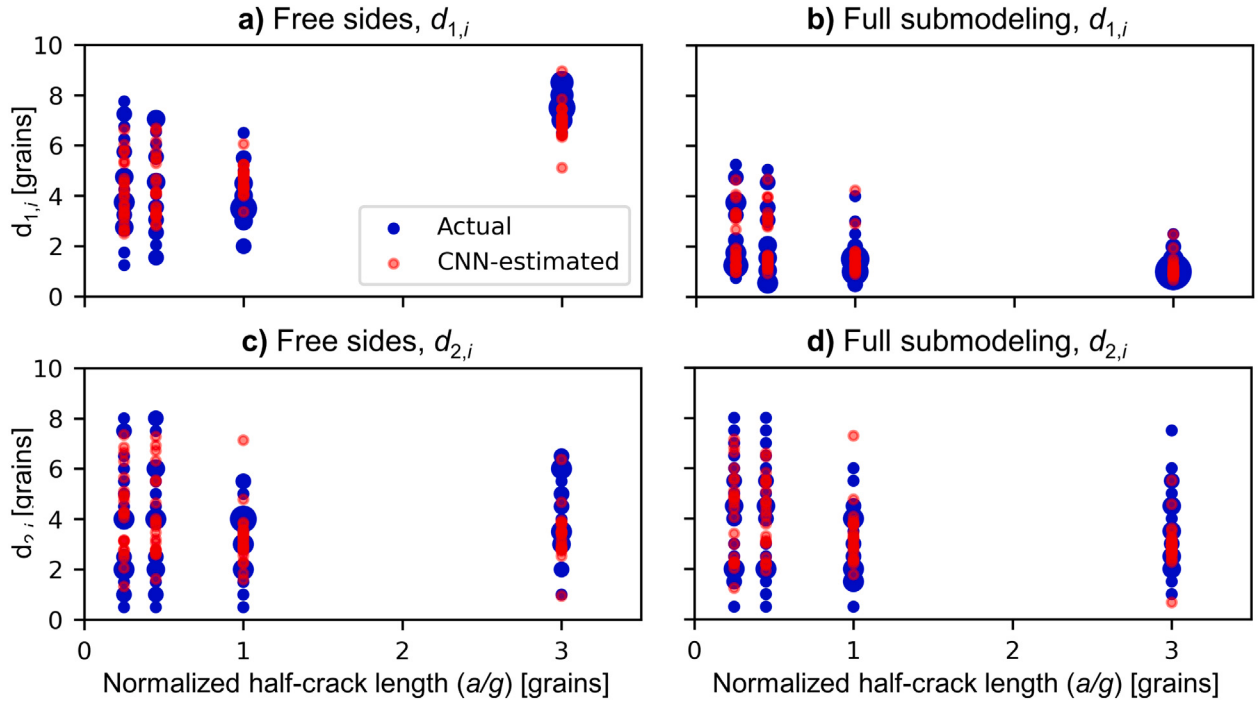


Fig. 8. Scatter plots of actual and CNN-estimated $RVE_{MSC,i}$ parameter values. Estimates are made using the ensemble method described in Section 2.3.3. Results are shown for four different CNN estimates of a specific $RVE_{MSC,i}$ parameter under a specific boundary condition: (a) $d_{1,i}$ with free sides, (b) $d_{1,i}$ with full submodeling, (c) $d_{2,i}$ with free sides, and (d) $d_{2,i}$ with full submodeling. Relative sizes of points reflect number of times a given value is repeated in the data set.

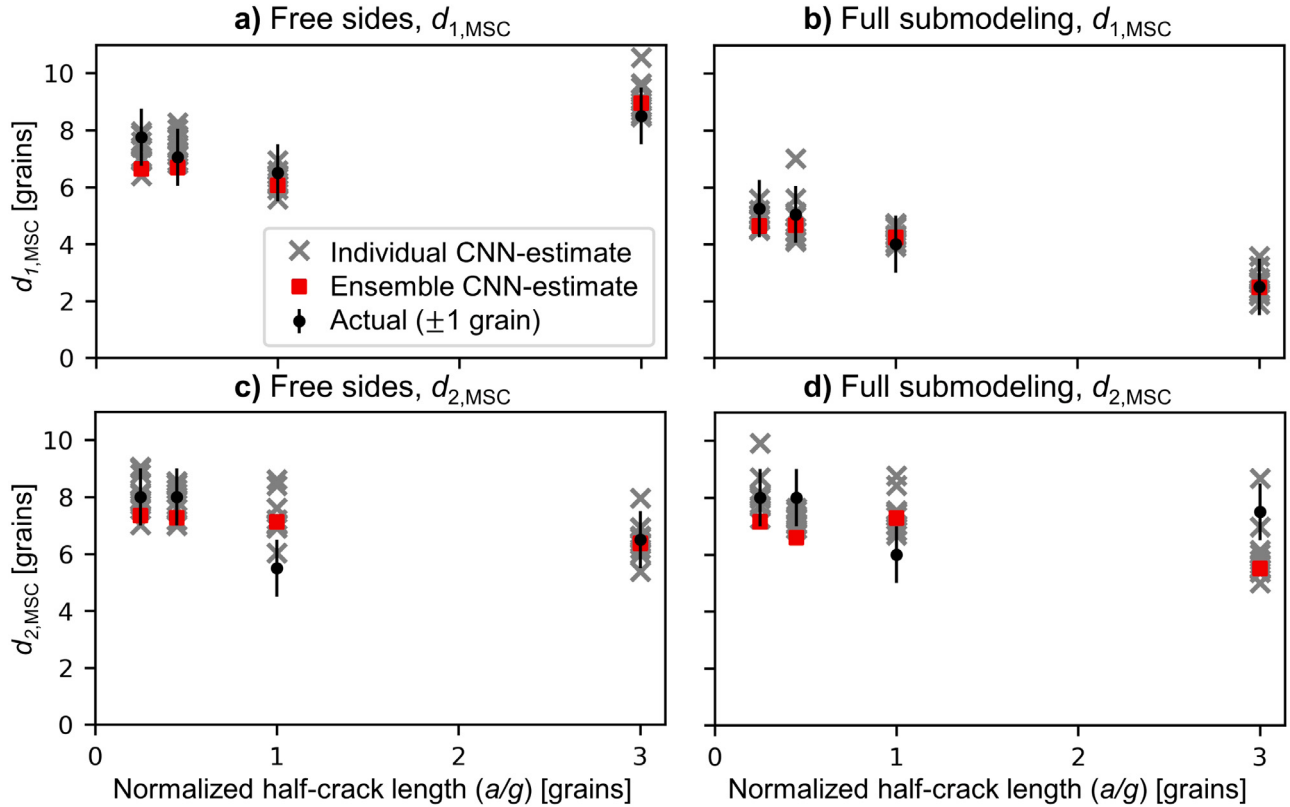


Fig. 9. Actual and CNN-derived estimates of RVE_{MSC} parameter values. Actual values are shown with ± 1 grain error bars. The individual RVE_{MSC} parameter estimates are made during ten individual CNN model runs. The ten individual estimates are used to calculate the ensemble model estimates. Both $d_{1,ip}$ (a,b) and $d_{2,ip}$ (c,d) values are compared for free-sides (a,c) and full-submodeling (b,d) boundary conditions.

optimal training/validation data amount is selected as 50 CFP groups, or ~8000 crack-front data points. This optimal training/validation data

amount is used in training the CNN models for studies discussed in the following sections.

Table 4

Performance metrics for the microstructure screening study in which actual and CNN-estimated $RVE_{MSC,i}$ values are placed into bins of large, medium, or small volume requirements. Metrics are given for the four types of CNN models corresponding to each combination of boundary condition and RVE_{MSC} parameter type.

Model type	Test set	Large/Medium/Small	Large/Not large		
		Accuracy [%]	Accuracy [%]	Sensitivity [%]	Precision [%]
Free sides, d_1	1	50.00	91.67	66.67	100.00
	2	66.67	83.33	80.00	80.00
	3	58.33	83.33	100.00	66.67
Free sides, d_2	1	83.33	91.67	66.67	100.00
	2	41.67	58.33	20.00	50.00
	3	75.00	75.00	50.00	66.67
Full submodeling, d_1	1	66.67	83.33	50.00	100.00
	2	66.67	75.00	60.00	75.00
	3	83.33	91.67	75.00	100.00
Full submodeling, d_2	1	50.00	66.67	50.00	50.00
	2	75.00	83.33	50.00	100.00
	3	75.00	91.67	100.00	80.00

4.2. Improvement of RVE_{MSC} predictions via ensemble CNN

Table 3 demonstrates the improvements in $RVE_{MSC,i}$ estimates and $RVE_{MSC,ip}$ predictions from applying an ensemble prediction approach. Seven of the eight (87.5 %) average R^2 values from ensemble cross-validation predictions (or estimates) are greater than the corresponding individual cross-validation R^2 values. Decreases in average RMSE values accompany the seven increases in average R^2 values.

The improvement of the ensemble cross-validation predictions (or CNN-based estimates) over the individual cross-validation predictions (or CNN-based estimates) comes from harnessing the stochasticity associated with CNN training to stabilize CNN predictions. During the CNN training for ensemble cross-validation, as described in Section 2.3.3, there are two primary sources of stochasticity. First, the data used for training and validation are randomly shuffled during each training run. Second, the CNN model weights are randomly initialized using the Glorot uniform initializer [67] at the beginning of each training run. The resulting stochasticity leads to slightly different CNN predictions for a given test set each time the CNN is trained.

The stabilization in CNN-based RVE_{MSC} parameter estimates that occurs by averaging slightly different $RVE_{MSC,ip}$ parameter predictions is shown in Fig. 9. Made according to the steps outlined in Section 2.3.3, the ensemble RVE_{MSC} parameter value estimates (shown as red squares) represent the results of averaging ten sets of individual $RVE_{MSC,ip}$ parameter value predictions. Looking at the ten individual model RVE_{MSC} parameter value estimates (shown as gray crosses), there is a difference of more than one grain separating the maximum and minimum individual $d_{1,MSC}$ or $d_{2,MSC}$ values for each crack size, which can translate to a difference of hundreds of grains in RVE_{MSC} . Additionally, some of the individual CNN-based estimates are unusually high or low relative to other CNN-based estimates. Using the ensemble model, each of the ten predictions is considered in selecting a prediction favored by the model over multiple training runs, while diminishing the effect of unusually high or low individual model predictions.

4.3. Performance of CNN in predicting RVE_{MSC}

Having established the superior performance of the ensemble CNN model approach to an individual CNN, we evaluate the regression performance of the ensemble CNN models. Ensemble cross-validation R^2 values averaged over five modified five-fold cross-validations (Section 2.3.3) in Table 3 vary from 0.67 to 0.82 for $RVE_{MSC,ip}$ parameters and from 0.52 to 0.80 for $RVE_{MSC,i}$ parameters, depending on boundary condition and RVE_{MSC} parameter. The best CNN performance is associated with the full-submodeling boundary condition and $d_{1,ip}$ (or $d_{1,i}$). In contrast, the worst CNN performance is associated with the free-sides boundary condition and $d_{2,ip}$ (or $d_{2,i}$). The variations may be explained, in part, through the trends in the $RVE_{MSC,ip}$ (or $RVE_{MSC,i}$) parameter values being predicted (or estimated). Fig. 8 demonstrates some key

trends in actual $RVE_{MSC,i}$ parameter values that may impact CNN-based estimates of $RVE_{MSC,i}$. First, the ranges of actual $d_{1,i}$ values (Figs. 8a,b) are smaller than the ranges of $d_{2,i}$ values (Figs. 8c,d), especially at larger crack sizes. Second, the $d_{1,i}$ values for the full-submodeling boundary condition (Fig. 8b) converge to values below two grains as crack size increases, while the $d_{1,i}$ values for the free-sides boundary condition (Fig. 8a) increase to values greater than six grains as the crack size increases beyond a/g of unity. The different ranges of $RVE_{MSC,i}$ parameter values impact the similarity of CNN target values for different microstructural instantiations and crack sizes. Increased similarity of target values leads to increased similarity between CNN target data in the training and test sets. Given that CNNs learn to make predictions based on the training data, test data that are the most similar to the training data (i.e., $d_{1,i}$ under the full-submodeling boundary condition) should be best predicted by the CNN. In other words, having more microstructural instantiations with similar $RVE_{MSC,i}$ parameter values might improve CNN performance as test data targets better reflect targets present in the training set.

Recalling that the CNN-estimated $RVE_{MSC,i}$ parameter values are used in determining RVE_{MSC} , we now evaluate the final CNN-based RVE_{MSC} estimates. Fig. 9 shows the RVE_{MSC} parameter estimates corresponding to the $RVE_{MSC,i}$ parameter estimates shown in Fig. 8. Of the 16 RVE_{MSC} parameter estimates shown in Fig. 9, 11 ensemble CNN-based estimates fall within one grain of the corresponding actual RVE_{MSC} parameter value. Optimally, RVE_{MSC} captures just enough microstructure to ensure that an MSC behaves as if it were embedded in a full-scale part, while minimizing the expense associated with the study of the MSC. Considering that a difference of just one grain in $d_{1,MSC}$ or $d_{2,MSC}$ translates to a difference of hundreds of grains in RVE_{MSC} , the CNN-based estimates of $d_{1,MSC}$ and $d_{2,MSC}$ cannot be relied upon to find RVE_{MSC} sizes that truly optimize MSC studies. With this in mind, completely replacing all FE simulations with CNN predictions when determining RVE_{MSC} cannot be justified.

4.4. Performance of CNN in identifying large-volume-requirement instantiations

Although CNN predictions cannot directly replace FE simulations, the ability of the CNN to identify microstructural instantiations with large volume requirements is now evaluated. For all except one test set, the large/medium/small binning accuracy is greater than or equal to 50%, as shown in Table 4. For each test set, the large/not large binning accuracy is both greater than 50% and higher than the large/medium /small binning accuracy. The binning accuracies demonstrate that CNNs generally estimate $RVE_{MSC,i}$ values within the same range as the actual $RVE_{MSC,i}$ values from FE simulations. Large labeling sensitivity, or the percent of actual large-volume-requirement instantiations correctly labeled as large-volume-requirement instantiations, is greater than or equal to 50% for all except one test set and equal

to 100% for two test sets. Thus, instantiations with $RVE_{MSC,i}$ values close to RVE_{MSC} are more often identified than not identified during the CNN-based screening for large-volume-requirement instantiations. Large-volume binning precision, or the percent of CNN-estimated large-volume-requirement instantiations that are actually large-volume-requirement instantiations, is greater than or equal to 50% for all test sets and equal to 100% for five test sets. Thus, most of the instantiations identified during the CNN-based screening for large-volume-requirement instantiations are the large-volume-requirement instantiations of interest.

4.5. Benefits of incorporating CNNs into the determination of RVE_{MSC}

Incorporating CNNs into the determination of RVE_{MSC} provides several improvements compared to an FE-only based framework [7]. First, the rapid predictions of the CNN allow for many more microstructural instantiations to be considered. In the FE-based framework, the number of instantiations considered was limited by computational cost. Second, the selection of critical instantiations (i.e., those that likely contribute to a conservative estimate of RVE_{MSC}) to simulate with FE will be more informed. In the FE-based framework, instantiations were pseudo-randomly generated with no indication *a priori* of whether or not $RVE_{MSC,i}$ would be near RVE_{MSC} . The blind selection process led to many $RVE_{MSC,i}$ values that were much smaller than RVE_{MSC} and did not directly contribute to the conservative estimation of RVE_{MSC} . With CNNs, microstructural instantiations that are likely to have $RVE_{MSC,i}$ values close to RVE_{MSC} can be identified and simulated, making FE simulations more strategic in determining RVE_{MSC} . As screened instantiations are simulated, the new simulation results serve to improve the estimation of RVE_{MSC} and improve the performance of the CNN by increasing the amount of available training data (Fig. 7).

Thus, the results from this study suggest that FE simulations in tandem with CNN models can serve to rapidly, yet with reasonable accuracy, make conservative estimates of the minimum volume requirements for heterogeneous microstructures containing MSCs as a function of a/g and boundary condition.

4.6. Computational cost comparison

The computational cost of training CNNs (assuming sufficient training data are available) and making $RVE_{MSC,i}$ estimates based on CNN predictions is significantly lower than the cost of performing the FE simulations described in Section 2.1. The ten different CNNs needed for making ensemble predictions can be trained in less than 100 CPU hours. Once CNNs are trained, a CNN-based $d_{1,i}$ or $d_{2,i}$ estimate for one microstructural instantiation can be made in less than 1 s. On the other hand, using FE simulations to determine $d_{1,i}$ or $d_{2,i}$ for one microstructural instantiation required at least 40 CPU hours. Even accounting for CNN training time, hundreds of $d_{1,i}$ or $d_{2,i}$ values can be estimated via an ensemble CNN approach in the same amount of time that two $d_{1,i}$ or $d_{2,i}$ values can be determined via FE simulations.

5. Limitations

The CNNs trained during this work are only tested on making CNN-based estimates of RVE_{MSC} for idealized microstructures with linear-elastic behavior. The idealized microstructures consist of cube-shaped grains whose orientations are implicitly specified via different elastic moduli. The surface cracks included in the microstructures are planar, semi-circular, and vary in size from $a/g = 0.25$ to $a/g = 3$. The transferability of the trained CNNs from this work to conditions varying from those in the training data, such as non-idealized microstructures, elastic-plastic behavior, internal cracks, non-semi-circular cracks, or cracks with $a/g > 3$, is not explored here. However, new CNNs could be trained to estimate RVE_{MSC} (via $RVE_{MSC,ip}$ predictions) for different conditions by providing relevant training data to CNN models.

6. Conclusions

In computational materials science, computational expense is often a limiting factor when investigating material behavior via numerical simulation. Here, machine learning is explored as a potential tool for reducing the computational expense required for investigations of material behavior. Convolutional neural networks (CNNs) are implemented to expedite the determination of representative volume elements for microstructurally small cracks (RVE_{MSC}). By definition, RVE_{MSC} is the minimum volume of microstructure required around a microstructurally small crack (MSC) to achieve convergence of crack-front parameters with respect to volume size [7].

In a previous study [7], RVE_{MSC} was determined using a computationally expensive framework, which relied on three-dimensional finite-element (FE) simulations of idealized heterogeneous microstructures containing MSCs. The FE simulations considered 88 combinations of microstructure and crack size under two different types of boundary conditions (full submodeling and free sides). The previous FE simulations provide data for the CNNs in this work.

In this work, CNNs are trained to predict $RVE_{MSC,ip}$ parameters $d_{1,ip}$ and $d_{2,ip}$ at 14168 crack-front points, where $RVE_{MSC,ip}$ is the volume at which the crack-front parameter at a given crack-front point p in a given microstructure i converges with respect to volume size. Inputs to the CNNs comprise microstructural and geometrical information local to the given crack-front point. Estimates of $RVE_{MSC,i}$ parameters ($d_{1,i}$ and $d_{2,i}$) are made from $RVE_{MSC,ip}$ parameter predictions, where $RVE_{MSC,i}$ is the volume at which crack-front parameters in a given microstructure i converge with respect to volume size. The $RVE_{MSC,i}$ parameter estimates are subsequently used to make conservative estimates of RVE_{MSC} parameters ($d_{1,MSC}$ and $d_{2,MSC}$).

Several studies are performed to evaluate the effectiveness of using CNNs to expedite the determination of RVE_{MSC} . First, the amount of training data provided to the CNNs is systematically increased to determine the optimal amount of data to use in CNN training. Second, individual CNN model predictions are compared to predictions from an ensemble CNN approach that incorporates predictions from multiple CNNs. Third, CNNs are evaluated on two different tasks: estimating $RVE_{MSC,i}$ values from CNN predictions and screening microstructural instantiations to identify those requiring large volumes. From the studies, the following conclusions are drawn:

- The improvement in CNN model performance with the addition of more training data diminishes as the total amount of training data increases. Considering the computational cost associated with obtaining more training data, an optimal amount of training data for this problem is selected by identifying the point where model performance plateaus (50 cracked microstructural instantiations, corresponding to ~8000 discrete data points).
- Using an ensemble CNN approach improves $RVE_{MSC,ip}$ predictions and $RVE_{MSC,i}$ estimates over individual CNN predictions and estimates, respectively. Taking advantage of the stochasticity associated with CNN training, noisy individual CNN predictions are stabilized in ensemble predictions by averaging individual CNN predictions (ten, in this case).
- The performance of CNN models varies when estimating different $RVE_{MSC,i}$ parameters under different boundary conditions. The best performance is achieved for the full submodeling, $d_{1,i}$ CNN-based estimates (average ensemble model R^2 of 0.80). In contrast, the worst performance is achieved for free sides, $d_{2,i}$ CNN-based estimates (average ensemble model R^2 of 0.52). The difference in performance indicates that features of the $RVE_{MSC,i}$ data, such as trends across crack size and data scatter, significantly impact CNN performance.
- Despite the previous conclusion, CNN predictions are inadequate as direct substitutes for FE simulations in determining RVE_{MSC} . Estimates of RVE_{MSC} parameter values based on CNN predictions

have errors greater than one grain. The differences in RVE_{MSC} parameter values translate to differences of hundreds of grains between the actual and estimated RVE_{MSC} sizes.

- Overall, CNNs are effective as a tool for identifying microstructural instantiations with large $RVE_{MSC,i}$ parameter values. Accuracies ranging from 58.3 % to 91.7 % are achieved in identifying instantiations with large $RVE_{MSC,i}$ parameters (top 33% of $RVE_{MSC,i}$ parameter value range) from other instantiations.
- Using CNN predictions to estimate $RVE_{MSC,i}$ parameters provides massive time savings over the previous FE-based approach. In the time required to train CNN models and make CNN-based estimates of $RVE_{MSC,i}$ for hundreds of cracked microstructural instantiations, $RVE_{MSC,i}$ can only be determined for two cracked microstructural instantiations with the FE-based approach.

While not capable of completely replacing FE simulations, CNN predictions streamline an inefficient, simulation-based framework that previously relied on brute-force FE simulations of a limited number of cracked microstructural instantiations. Through rapid CNN predictions, a large number of cracked microstructural instantiations can be evaluated, and microstructures likely to contribute to the conservative estimate of RVE_{MSC} can be identified and confirmed via FE simulation.

As demonstrated through the application of CNNs to the determination of RVE_{MSC} , future studies of material behavior involving computationally expensive simulations of microstructure can benefit from the incorporation of machine learning. For example, CNNs can be used to rapidly predict material response with reduced numerical simulation. Thus, CNNs allow for rapid yet thorough searches of a materials-design space. When machine learning predictions of material response are not sufficiently accurate to fully replace physics-based simulations, rapid CNN predictions can be used to inform the selection of microstructural instantiations to investigate via high-fidelity simulation. With this CNN-based screening approach, both CNN efficiency and simulation accuracy can be harnessed in future materials investigations.

CRedit authorship contribution statement

Karen J. DeMille: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft, Visualization, Funding acquisition. **Ashley D. Spear:** Conceptualization, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Acknowledgments

KJD was supported by the Department of Defense (DoD), USA through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program. This material is based upon work supported by the National Science Foundation, USA under Grant No. CMMI-1752400. The support and resources from the Center for High Performance Computing at the University of Utah are gratefully acknowledged. The authors gratefully acknowledge Aowabin Rahman and Vignesh Babu Rao for their valuable discussions and support.

References

- [1] S. Suresh, *Fatigue of Materials*, Cambridge University Press, 1998, <http://dx.doi.org/10.1017/CBO9780511806575>.
- [2] D. McDowell, Damage mechanics and metal fatigue: a discriminating perspective, *Int. J. Damage Mech.* 8 (4) (1999) 376–403, <http://dx.doi.org/10.1177/105678959900800406>.
- [3] W. Ludwig, J.-Y. Buffière, S. Savelli, P. Cloetens, Study of the interaction of a short fatigue crack with grain boundaries in a cast Al alloy using X-ray microtomography, *Acta Mater.* 51 (3) (2003) 585–598, [http://dx.doi.org/10.1016/S1359-6454\(02\)00320-8](http://dx.doi.org/10.1016/S1359-6454(02)00320-8).
- [4] G.M. Castelluccio, W.D. Musinski, D.L. McDowell, Recent developments in assessing microstructure-sensitive early stage fatigue of polycrystals, *Curr. Opin. Solid State Mater. Sci.* 18 (4) (2014) 180–187, <http://dx.doi.org/10.1016/j.cossms.2014.03.001>.
- [5] A. Ayyar, N. Chawla, Microstructure-based modeling of the influence of particle spatial distribution and fracture on crack growth in particle-reinforced composites, *Acta Mater.* 55 (18) (2007) 6064–6073, <http://dx.doi.org/10.1016/j.actamat.2007.06.044>.
- [6] A.R. Najafi, A.R. Arshi, M. Eslami, S. Fariborz, M.H. Moeinzadeh, Micromechanics fracture in osteonal cortical bone: A study of the interactions between microcrack propagation, microstructure and the material properties, *J. Biomech.* 40 (12) (2007) 2788–2795, <http://dx.doi.org/10.1016/j.jbiomech.2007.01.017>.
- [7] K.J. DeMille, A.D. Spear, Determination of representative volume elements for small cracks in heterogeneous, linear-elastic domains, *Eng. Fract. Mech.* 220 (2019) 106643, <http://dx.doi.org/10.1016/j.engfractmech.2019.106643>.
- [8] W.J. Drugan, J.R. Willis, A micromechanics-based nonlocal constitutive equation and estimates of representative volume element size for elastic composites, *J. Mech. Phys. Solids* 44 (4) (1996) 497–524, [http://dx.doi.org/10.1016/0022-5096\(96\)00007-5](http://dx.doi.org/10.1016/0022-5096(96)00007-5).
- [9] M. Ostoja-Starzewski, Material spatial randomness: From statistical to representative volume element, *Probab. Eng. Mech.* 21 (2) (2006) 112–132, <http://dx.doi.org/10.1016/j.probabeng.2005.07.007>.
- [10] D. Trias, J. Costa, A. Turon, J.E. Hurtado, Determination of the critical size of a statistical representative volume element (SRVE) for carbon reinforced polymers, *Acta Mater.* 54 (13) (2006) 3471–3484, <http://dx.doi.org/10.1016/j.actamat.2006.03.042>.
- [11] S. Swaminathan, S. Ghosh, N.J. Pagano, Statistically equivalent representative volume elements for unidirectional composite microstructures: Part I-Without damage, *J. Compos. Mater.* 40 (7) (2006) 583–604, <http://dx.doi.org/10.1177/0021998305055273>.
- [12] M. Nygård, Number of grains necessary to homogenize elastic materials with cubic symmetry, *Mech. Mater.* 35 (11) (2003) 1049–1057, [http://dx.doi.org/10.1016/S0167-6636\(02\)00325-3](http://dx.doi.org/10.1016/S0167-6636(02)00325-3).
- [13] Z.-Y. Ren, Q.-S. Zheng, Effects of grain sizes, shapes, and distribution on minimum sizes of representative volume elements of cubic polycrystals, *Mech. Mater.* 36 (12) (2004) 1217–1229, <http://dx.doi.org/10.1016/j.mechmat.2003.11.002>.
- [14] M. Kovač, L. Cizelj, Modeling elasto-plastic behavior of polycrystalline grain structure of steels at mesoscopic level, *Nucl. Eng. Des.* 235 (17–19) (2005) 1939–1950, <http://dx.doi.org/10.1016/j.nucengdes.2005.05.009>.
- [15] S.I. Ranganathan, M. Ostoja-Starzewski, Scaling function, anisotropy and the size of RVE in elastic random polycrystals, *J. Mech. Phys. Solids* 56 (9) (2008) 2773–2791, <http://dx.doi.org/10.1016/j.jmps.2008.05.001>.
- [16] K.S. Stopka, M. Yaghoobi, J.E. Allison, D.L. McDowell, Simulated effects of sample size and grain neighborhood on the modeling of extreme value fatigue response, *Acta Mater.* (2022) 117524, <http://dx.doi.org/10.1016/j.actamat.2021.117524>.
- [17] Z. Shan, A.M. Gokhale, Representative volume element for non-uniform microstructure, *Comput. Mater. Sci.* 24 (3) (2002) 361–379, [http://dx.doi.org/10.1016/S0927-0256\(01\)00257-9](http://dx.doi.org/10.1016/S0927-0256(01)00257-9).
- [18] T. Ozturk, C. Stein, R. Pokharel, C. Hefferan, H. Tucker, S. Jha, R. John, R.A. Lebensohn, P. Kenesei, R.M. Suter, et al., Simulation domain size requirements for elastic response of 3D polycrystalline materials, *Modelling Simulation Mater. Sci. Eng.* 24 (1) (2015) 015006, <http://dx.doi.org/10.1088/0965-0393/24/1/015006>.
- [19] I. Simonovski, L. Cizelj, Representative volume element size of a polycrystalline aggregate with embedded short crack, in: *Proceedings of the International Conference Nuclear Energy for New Europe*, 2007, pp. 0906–1.
- [20] D.M. Dimiduk, E.A. Holm, S.R. Niezgoda, Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering, *Integr. Mater. Manuf. Innov.* 7 (3) (2018) 157–172, <http://dx.doi.org/10.1007/s40192-018-0117-8>.
- [21] U. Ali, W. Muhammad, A. Brahme, O. Skiba, K. Inal, Application of artificial neural networks in micromechanics for polycrystalline metals, *Int. J. Plast.* 120 (2019) 205–219, <http://dx.doi.org/10.1016/j.iplas.2019.05.001>.
- [22] C. Herriott, A.D. Spear, Predicting microstructure-dependent mechanical properties in additively manufactured metals with machine-and deep-learning methods, *Comput. Mater. Sci.* 175 (2020) 109599, <http://dx.doi.org/10.1016/j.commatsci.2020.109599>.

- [23] A. Beniwal, R. Dadhich, A. Alankar, Deep learning based predictive modeling for structure-property linkages, *Materialia* 8 (2019) 100435, <http://dx.doi.org/10.1016/j.mta.2019.100435>.
- [24] L. Liang, M. Liu, C. Martin, W. Sun, A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis, *J. R. Soc. Interface* 15 (138) (2018) 20170844, <http://dx.doi.org/10.1098/rsif.2017.0844>.
- [25] W. Muhammad, A.P. Brahme, O. Ibragimova, J. Kang, K. Inal, A machine learning framework to predict local strain distribution and the evolution of plastic anisotropy & fracture in additively manufactured alloys, *Int. J. Plast.* 136 (2021) 102867, <http://dx.doi.org/10.1016/j.ijplas.2020.102867>.
- [26] A. Mangal, E.A. Holm, Applied machine learning to predict stress hotspots I: Face centered cubic materials, *Int. J. Plast.* 111 (2018) 122–134, <http://dx.doi.org/10.1016/j.ijplas.2018.07.013>.
- [27] A. Mangal, E.A. Holm, A comparative study of feature selection methods for stress hotspot classification in materials, *Integr. Mater. Manuf. Innov.* 7 (3) (2018) 87–95, <http://dx.doi.org/10.1007/s40192-018-0109-8>.
- [28] A. Mangal, E.A. Holm, Applied machine learning to predict stress hotspots II: Hexagonal close packed materials, *Int. J. Plast.* 114 (2019) 1–14, <http://dx.doi.org/10.1016/j.ijplas.2018.08.003>.
- [29] M.V. Pathan, S.A. Ponnusami, J. Pathan, R. Pitongsawat, B. Erice, N. Petrinic, V.L. Tagarielli, Predictions of the mechanical properties of unidirectional fibre composites by supervised machine learning, *Sci. Rep.* 9 (2019) 13964, <http://dx.doi.org/10.1038/s41598-019-50144-w>.
- [30] A. Rovinelli, M.D. Sangid, H. Proudhon, Y. Guilhem, R.A. Lebensohn, W. Ludwig, Predicting the 3D fatigue crack growth rate of small cracks using multimodal data via Bayesian networks: In-situ experiments and crystal plasticity simulations, *J. Mech. Phys. Solids* 115 (2018) 208–229, <http://dx.doi.org/10.1016/j.jmps.2018.03.007>.
- [31] A. Rovinelli, M.D. Sangid, H. Proudhon, W. Ludwig, Using machine learning and a data-driven approach to identify the small fatigue crack driving force in polycrystalline materials, *npj Comput. Mater.* 4 (2018) 35, <http://dx.doi.org/10.1038/s41524-018-0094-7>.
- [32] A. Pandey, R. Pokharel, Machine learning based surrogate modeling approach for mapping crystal deformation in three dimensions, *Scr. Mater.* 193 (2021) 1–5, <http://dx.doi.org/10.1016/j.scriptamat.2020.10.028>.
- [33] R. Cang, Y. Xu, S. Chen, Y. Liu, Y. Jiao, M. Yi Ren, Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design, *J. Mech. Des.* 139 (7) (2017) 071404, <http://dx.doi.org/10.1115/1.4036649>.
- [34] Z. Yang, Y.C. Yabansu, D. Jha, W.-k. Liao, A.N. Choudhary, S.R. Kalidindi, A. Agrawal, Establishing structure-property localization linkages for elastic deformation of three-dimensional high contrast composites using deep learning approaches, *Acta Mater.* 166 (2019) 335–345, <http://dx.doi.org/10.1016/j.actamat.2018.12.045>.
- [35] A. Agrawal, A. Choudhary, Deep materials informatics: Applications of deep learning in materials science, *MRS Commun.* 9 (2019) 779–792, <http://dx.doi.org/10.1557/mrc.2019.73>.
- [36] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (2015) 436–444, <http://dx.doi.org/10.1038/nature14539>.
- [37] E.A. Holm, R. Cohn, N. Gao, A.R. Kitahara, T.P. Matson, B. Lei, S.R. Yarasi, Overview: Computer vision and machine learning for microstructural characterization and analysis, *Metall. Mater. Trans. A* (2020) 5985–5999, <http://dx.doi.org/10.1007/s11661-020-06008-4>.
- [38] N. Lubbers, T. Lookman, K. Bhar, Inferring low-dimensional microstructure representations using convolutional neural networks, *Phys. Rev. E* 96 (2017) 052111, <http://dx.doi.org/10.1103/PhysRevE.96.052111>.
- [39] A. Cecen, H. Dai, Y.C. Yabansu, S.R. Kalidindi, L. Song, Material structure-property linkages using three-dimensional convolutional neural networks, *Acta Mater.* 146 (2018) 76–84, <http://dx.doi.org/10.1016/j.actamat.2017.11.053>.
- [40] Z. Yang, Y.C. Yabansu, R. Al-Bahrani, W.-k. Liao, A.N. Choudhary, S.R. Kalidindi, A. Agrawal, Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets, *Comput. Mater. Sci.* 151 (2018) 278–287, <http://dx.doi.org/10.1016/j.commatsci.2018.05.014>.
- [41] X. Li, Z. Liu, S. Cui, C. Luo, C. Li, Z. Zhuang, Predicting the effective mechanical property of heterogeneous materials by image based modeling and deep learning, *Comput. Methods Appl. Mech. Engrg.* 347 (2019) 735–753, <http://dx.doi.org/10.1016/j.cma.2019.01.005>.
- [42] D.W. Abueidda, M. Almasri, R. Ammourah, U. Ravaoli, I.M. Jasiuk, N.A. Sobh, Prediction and optimization of mechanical properties of composites using convolutional neural networks, *Compos. Struct.* 227 (2019) 111264, <http://dx.doi.org/10.1016/j.compstruct.2019.111264>.
- [43] S. Ye, B. Li, Q. Li, H.-P. Zhao, X.-Q. Feng, Deep neural network method for predicting the mechanical properties of composites, *Appl. Phys. Lett.* 115 (2019) 161901, <http://dx.doi.org/10.1063/1.5124529>.
- [44] A. Frankel, K. Tachida, R. Jones, Prediction of the evolution of the stress field of polycrystals undergoing elastic-plastic deformation with a hybrid neural network model, *Mach. Learn.: Sci. Technol.* 1 (3) (2020) 035005, <http://dx.doi.org/10.1088/2632-2153/ab9299>.
- [45] B.P. Croom, M. Berkson, R.K. Mueller, M. Presley, S. Storck, Deep learning prediction of stress fields in additively manufactured metals with intricate defect networks, 2021, arXiv preprint [arXiv:2105.10564](https://arxiv.org/abs/2105.10564).
- [46] K. Pierson, A. Rahman, A.D. Spear, Predicting microstructure-sensitive fatigue-crack path in 3D using a machine learning framework, *JOM* 71 (2019) 2680–2694, <http://dx.doi.org/10.1007/s11837-019-03572-y>.
- [47] C. Kantzos, J. Lao, A. Rollett, Design of an interpretable Convolutional Neural Network for stress concentration prediction in rough surfaces, *Mater. Charact.* 158 (2019) 109961, <http://dx.doi.org/10.1016/j.matchar.2019.109961>.
- [48] *Abaqus Analysis User's Guide* (6.14), Dassault Systèmes, 2014.
- [49] P.J. Withers, J. Bennett, Y.-C. Hung, M. Preuss, Crack opening displacements during fatigue crack growth in Ti–SiC fibre metal matrix composites by X-ray tomography, *Mater. Sci. Technol.* 22 (9) (2006) 1052–1058, <http://dx.doi.org/10.1179/174328406X114108>.
- [50] H. Toda, S. Yamamoto, M. Kobayashi, K. Uesugi, H. Zhang, Direct measurement procedure for three-dimensional local crack driving force using synchrotron X-ray microtomography, *Acta Mater.* 56 (20) (2008) 6027–6039, <http://dx.doi.org/10.1016/j.actamat.2008.08.022>.
- [51] N. Limodin, J. Réthoré, J.-Y. Buffière, A. Gravouil, F. Hild, S. Roux, Crack closure and stress intensity factor measurements in nodular graphite cast iron using three-dimensional correlation of laboratory X-ray microtomography images, *Acta Mater.* 57 (14) (2009) 4090–4101, <http://dx.doi.org/10.1016/j.actamat.2009.05.005>.
- [52] J. Williams, K. Yazzie, E. Padilla, N. Chawla, X. Xiao, F. De Carlo, Understanding fatigue crack growth in aluminum alloys by in situ X-ray synchrotron tomography, *Int. J. Fatigue* 57 (2013) 79–85, <http://dx.doi.org/10.1016/j.ijfatigue.2012.06.009>.
- [53] A. Rovinelli, R.A. Lebensohn, M.D. Sangid, Influence of microstructure variability on short crack behavior through postulated micromechanical short crack driving force metrics, *Eng. Fract. Mech.* 138 (2015) 265–288, <http://dx.doi.org/10.1016/j.engfracmech.2015.03.001>.
- [54] G.M. Castelluccio, D.L. McDowell, Microstructure and mesh sensitivities of mesoscale surrogate driving force measures for transgranular fatigue cracks in polycrystals, *Mater. Sci. Eng. A* 639 (2015) 626–639, <http://dx.doi.org/10.1016/j.msea.2015.05.048>.
- [55] H. Bao, A.J. McEvily, On plane stress–plane strain interactions in fatigue crack growth, *Int. J. Fatigue* 20 (6) (1998) 441–448, [http://dx.doi.org/10.1016/S0142-1123\(98\)00013-9](http://dx.doi.org/10.1016/S0142-1123(98)00013-9).
- [56] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- [57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* 15 (56) (2014) 1929–1958, URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- [58] F. Chollet, et al., Keras, 2015, <https://keras.io>.
- [59] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org. URL <https://www.tensorflow.org/>.
- [60] M. Pumperla, Hyperas: Keras + Hyperopt: A very simple wrapper for convenient hyperparameter optimization, 2020, <https://maxpumperla.com/hyperas/>.
- [61] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017, arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [62] G. Vanwinckelen, H. Blockeel, On estimating model accuracy with repeated cross-validation, in: *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning*, 2012, pp. 39–44.
- [63] Y. Ren, L. Zhang, P.N. Suganthan, Ensemble classification and regression-recent developments, applications and future directions, *IEEE Comput. Intell. Mag.* 11 (1) (2016) 41–53, <http://dx.doi.org/10.1109/MCI.2015.2471235>.
- [64] D. Cireşan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3642–3649, <http://dx.doi.org/10.1109/CVPR.2012.6248110>.
- [65] S.-Y. Kim, Effects of sample size on robustness and prediction accuracy of a prognostic gene signature, *BMC Bioinformatics* 10 (2009) 147, <http://dx.doi.org/10.1186/1471-2105-10-147>.
- [66] J. Cho, K. Lee, E. Shin, G. Choy, S. Do, How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? 2016, arXiv preprint [arXiv:1511.06348](https://arxiv.org/abs/1511.06348).
- [67] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y.W. Teh, M. Titterton (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, in: *Proceedings of Machine Learning Research*, 9, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256, <https://proceedings.mlr.press/v9/glorot10a.html>.