



Automated segmentation of porous thermal spray material CT scans with predictive uncertainty estimation

Carianne Martinez^{1,2} · Dan S. Bolintineanu¹ · Aaron Olson¹ · Theron Rodgers¹  · Brendan Donohoe¹ · Kevin M. Potter¹ · Scott A. Roberts¹ · Reeju Pokharel³ · Stephanie Forrest² · Nathan W. Moore¹

Received: 31 October 2022 / Accepted: 14 April 2023 / Published online: 31 May 2023
© The Author(s) 2023

Abstract

Thermal sprayed metal coatings are used in many industrial applications, and characterizing the structure and performance of these materials is vital to understanding their behavior in the field. X-ray computed tomography (CT) enables volumetric, nondestructive imaging of these materials, but precise segmentation of this grayscale image data into discrete material phases is necessary to calculate quantities of interest related to material structure. In this work, we present a methodology to automate the CT segmentation process as well as quantify uncertainty in segmentations via deep learning. Neural networks (NNs) have been shown to excel at segmentation tasks; however, memory constraints, class imbalance, and lack of sufficient training data often prohibit their deployment in high resolution volumetric domains. Our 3D convolutional NN implementation mitigates these challenges and accurately segments full resolution CT scans of thermal sprayed materials with maps of uncertainty that conservatively bound the predicted geometry. These bounds are propagated through calculations of material properties such as porosity that may provide an understanding of anticipated behavior in the field.

Keywords CT analysis · Deep learning · Segmentation · Uncertainty quantification

1 Introduction

In many science and engineering domains, considerable effort is extended toward modeling the performance of materials that will be used in manufactured parts. For parts that require high reliability such as those found in airplanes, turbines, and power plants, engineers must precisely understand the structural properties of the materials used in manufacturing to inform expectations for their performance in the field. X-ray computed tomography (CT) techniques capture images of the complex internal geometry of materials; however, for ingestion into simulations that allow analysts to study and predict the performance of these materials in various environments, CT scans must be processed to identify which type of material or void is represented by each voxel in the volume. For industrial scans, a typical CT scan contains over a billion voxels, prohibiting manual segmentation and necessitating automated tools.

✉ Carianne Martinez
cmarti5@sandia.gov

Dan S. Bolintineanu
dsboli@sandia.gov

Aaron Olson
aolson@sandia.gov

Theron Rodgers
trodger@sandia.gov

Brendan Donohoe
bdonoho@sandia.gov

Kevin M. Potter
kmpotte@sandia.gov

Scott A. Roberts
sarober@sandia.gov

Reeju Pokharel
reeju@lanl.gov

Stephanie Forrest
steph@asu.edu

Nathan W. Moore
nwmooore@sandia.gov

¹ Sandia National Laboratories, Albuquerque, NM, USA

² Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, USA

³ Los Alamos National Laboratory, Los Alamos, NM, USA

Perhaps the simplest applicable segmentation technique is the use of a naive “threshold” for segmenting materials [1] in which a human analyst chooses CT intensity value thresholds above and below which individual voxels are determined to be different materials. These threshold values can be chosen based on either what looks reasonable to the analyst or on outside information such as the expected fraction of each material in the samples. This method requires human judgment and accuracy/reproducibility is contingent on the quality of the judgment or on hard-to-acquire external information. Additionally, this method may lack generality since in many cases no single threshold can be chosen that distinguishes between all examples of two materials. One reason for this inherent ambiguity is that CT reconstructions often have flaws when the local intensity depends upon the structure of surrounding materials which can cast “shadows,” causing artifacts in the scans.

More sophisticated techniques, such as the “random walk” (RW) algorithm [2], are promising, but they often fail to achieve acceptable accuracy for analysis of material properties, and neither the threshold method nor the random walk method provide uncertainty estimates of their segmentation predictions. Therefore, a segmentation method is needed for high resolution industrial CT images which both uses intensity values and accounts for complex material geometric structure to provide precise segmentation and meaningful estimates of segmentation uncertainty. Such a method would lead to more accurate downstream analysis of structural properties, ultimately providing better predictive models of performance and digital twins, and it would enable uncertainty propagation from the segmentation phase to downstream quantities of interest.

In recent years, convolutional neural networks (CNNs) have achieved high segmentation accuracy, particularly in the medical domain [3–5]. However, applying CNNs to high-resolution volumes at the scale typical of industrial applications requires addressing several challenges. CNNs require graphics processing units (GPUs) for efficient training and deployment, but the memory required to ingest an entire CT scan often exceeds the capacity of even the largest commercially available GPUs. Other obstacles include an imbalance in the classes of materials present in the scans, and the lack of sufficiently large labeled training examples to support typical deep learning approaches.

In this paper, we present a methodology to overcome these obstacles. The primary contributions of this work are:

- A NN training pipeline that combines three different labeling approaches (synthetic, thresholding, and random walk) to produce useful segmentations in the absence of expert-labeled ground truth examples.
- An open-source software repository that efficiently manages volumetric CNN training on subvolumes with techniques that overcome class imbalance, together with an inference method that provides segmentation uncertainty estimates.
- A case study demonstrating our proposed methodology that characterizes the structure and predicts the performance-related properties of thermal-sprayed metal coatings with uncertainty.

Previous work [6] illustrated portions of our proposed workflow on data from multiple domains. Here, we provide a detailed description of the entire methodology and report results from a case study that applied the method to thermal sprayed porous materials. Given the emphasis on the digital twin modeling paradigm in recent literature, e.g., [7], the need for practical efficient volumetric segmentation that captures uncertainty in geometry is clear. Our data preparation, labeling, and 3D CNN implementation workflow can be directly applied to other domains that require high confidence in performance of manufactured parts.

2 Background

In this section, we present information about the data used in our case study and discuss related work in segmentation, neural network uncertainty quantification, and synthetic CT reconstruction.

2.1 Thermal-sprayed coatings

Thermal-sprayed metal coatings are used in applications such as airplane wings and turbines [8] and have been considered for the linings of nuclear fusion reactors [9]. A precise understanding of structure informs understanding of structural properties and predictions of performance in environments including these high-mechanical-stress and thermal environments. Characterization of pore morphology is particularly important for several downstream quantities of interest and therefore receives more attention in our study than distinguishing between various metallic phases. In some samples, a second metal was sprayed with the first in approximately equal volume to provide imaging contrast and aid in the characterization process. Ultimately, accurate segmentation of thermal-spray CT scans will be used to understand internal coating structures, help tune the thermal-spray process, and assess performance of the coatings in various environments, including mechanical and thermal environments. Analyses that will benefit from the proposed workflow include:

- Measuring spatial statistical properties in samples to enable model tuning to produce synthetic realizations including the SPPARKS thermal-spray model [10].
- Establishing efficient simulation of 2D mesostructures [11] to serve as a surrogate of 3D analysis for evaluating various properties that depend on mesostructure.
- Enabling effective comparisons of images of plasma facing materials before and after non-destructive hydrodynamic tests, such as those performed at the National Ignition Facility [12].
- Informing shock hydrodynamic simulations in thermal-sprayed structures [13].

2.2 Deep learning segmentation models

Image segmentation is part of many computer vision related tasks, including medical imaging analysis [3], scene recognition and understanding [14], and robotics [15]. Classical image segmentation techniques include thresholding [1], clustering [16], and region growing methods [17]. More sophisticated techniques rely on image processing methods such as wavelet transforms [18] and Fourier transforms [19] to aid in segmentation. In recent years, however, supervised deep learning methods have become a popular approach because they leverage human knowledge through labeled examples and achieve high accuracy; these new methods have set benchmarks across a variety of datasets. Convolutional neural networks have become a standard segmentation tool starting in 2D domains [4, 20], and later extending to 3D [3]. Attention-based methods are also emerging as viable segmentation methods [21].

2.3 Uncertainty quantification for deep learning predictions

Deep learning models are powerful tools which leverage patterns found in data to achieve highly accurate results. However, their predictions do not come with error bars by default. Uncertainty quantification for deep learning predictions is an active and open research field, and several approaches have been proposed, many of which are reviewed in [22]. Ensemble methods [23] have been shown to provide uncertainty estimates by training several NNs independently and pooling their output to calculate a final prediction, while retaining the variance over the model predictions to inform the uncertainty in the model's output. Another proposed approach uses the Bayesian neural network [24], where instead of point estimates of optimal NN weights that produce accurate output, a distribution over each NN weight is learned via variational inference. Finally, the approach we take in this work leverages work by Gal et al. [25], where dropout layers remain active in the NN at inference time to

introduce stochasticity in the model's inference predictions. The resulting Monte Carlo Dropout Network (MCDN) takes several samples of forward passes through the NN for each input example, producing a set of viable segmentations of the input example.

Dropout layers are typically used during training to regularize NNs to prevent overfitting to the training data by dropping the output of a randomly sampled subset of NN weights during training. Gal et al. [25] showed that active dropout layers at inference time approximate a Gaussian process leading to variance in the NN's predicted outputs. In our work, the final segmentation is taken as the mean over all inference runs for the same input, and the standard deviation over the outputs is interpreted as the uncertainty of the model's prediction. The main advantage of the MCDN is that dropout layers are easily integrated into already proven NN models without altering the architecture, increasing the number of NN parameters, or requiring us to train and store multiple models.

In cases where multiple labels are available for training examples, the variance in labels introduces an additional source of uncertainty. Hu et al. [26] studied multiple trusted sources of labels that enable a calibrated ground truth uncertainty target, improving the quality of predictive uncertainty estimates. This method requires a credible set of labels for training. In contrast, the labels we generate as training data are known to be imperfect and would not serve as a well-calibrated uncertainty standard for training. We interpret our estimates as conservative bounds on uncertainty and not well-calibrated to uncertainty in a set of ground truth labels.

2.4 Synthetic CT scans

Several software packages are publicly available to generate synthetic X-rays and CT scans of numerical objects. For this work, we use the ASTRA Toolbox Python library [27] to generate synthetic X-ray CT scans of the simulated microstructures generated from the SPPARKS thermal spray library [10]. The Python API available in ASTRA toolbox allows for GPU-accelerated image processing and reconstruction with several tunable settings such as detector size, number of X-ray projections, and source and detector position relative to the numerical object. The software is able to produce synthetic CT scans similar to those that are within the target segmentation domain for this work.

3 Methods

In this section, we present the details of our methods from dataset curation to NN model development and training, and finally inference and interpretation of our uncertainty characterization. The steps include:

1. Development of a synthetic training dataset.
2. Algorithmic segmentation of real CT scans to serve as a second labeled training dataset.
3. Training a CNN with both synthetic and real data to automatically segment CT scans with characterization of uncertainty in geometric features.
4. Inference of segmentations with uncertainty maps of all real CT scans in our case study.

3.1 Dataset generation and preparation

In order to train a NN to perform segmentation, labeled examples are necessary to accommodate supervised learning. The CT scans that are the target of this investigation lack ground truth labels, and as such we have developed training sets that consist of synthetic examples to augment training as well as algorithmically labeled examples from the real CT domain.

3.1.1 Synthetic training dataset development

In the absence of ground truth segmentations of the CT scans of the real materials with microstructures that are our target domain, we use simulated examples of these microstructures to generate an initial training set for our deep learning model. Synthetic data can provide examples of textural features that are important for NNs to capture to perform accurate segmentation. Using structures generated with the SPPARKS simulation library [10] further described below, we construct a training set with synthetic material examples of varied porosity intended to capture the viable range of porosities present in the real material examples that we intend to segment. The microstructures were not quantitatively tuned to recreate the experimental structures, but contained many relevant features such as multiscale porosity, unmelted particles, and layer-structures. The synthetic structures do not contain some features found in the experimental data that create additional complications such as oxides and diffraction ring artifacts from the CT process. Two-material volumes were generated in a study of stochastic model variation. Twenty-five simulations were performed with identical input parameters but unique random number generator seeds. Model parameters used in the study resulted in a total pore fraction distribution with a mean of 0.059 and standard deviation of 0.0122. The distribution of total pore fractions is shown in Fig. 1.

An additional dataset of 20 simulations (10 single material volumes and 10 two material volumes) was generated to create synthetic volumes with total pore fractions closer to the expected experimental mean of 0.02. The range of total pore fractions in this second set varied between 0.02 and 0.05.

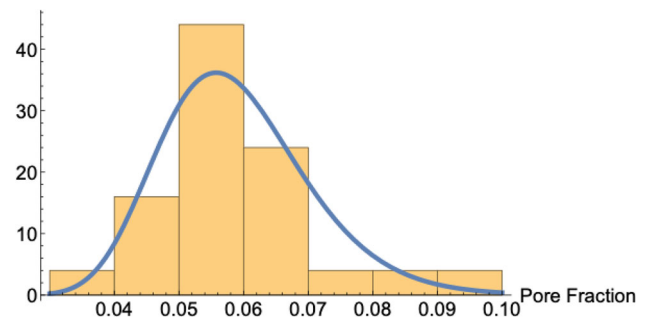


Fig. 1 Distribution of total pore fraction from 25-statistically equivalent microstructure generation simulations

All synthetic data was generated using the Thermal Spray app in the SPPARKS microstructure simulation library [10]. The thermal spray model generates synthetic microstructures at experimentally relevant length-scales using a rules-based model of particle incidence, spreading, and solidification during thermal spray processes. Synthetic microstructures consist of one pore/void phase and an arbitrary number of solid phases. Relative abundances of the solid phases are specified by user parameters, while the abundance of the pore phase emerges from the simulation behavior and cannot be directly specified with an input parameter.

As material phases are represented by integer values, the synthetic volumes are “pre-segmented” and do not require interpretation from continuum intensity values as with experimental CT data. The simulation also generates a unique label for each individual particle/splat; this is not currently used in the segmentation training but would provide ground-truth data for the identification of individual splats, rather than just material phases. Two classes of microstructures were generated for training—one-material volumes representing Ta coatings and two-material volumes representing Ta and Nb coatings, which were chosen experimentally for their excellent miscibility and to provide phase contrast during imaging. The Ta/Nb coatings were created to match the experimental condition of approximately equal number of Ta and Nb particles. The two material volumes were easily transformed to one material volumes by assigning all solid phases to the same integer.

Synthetic CT scans

After establishing the ground truth synthetic material examples, we leveraged an open source python library to develop a synthetic CT generation framework with parameters set to result in images that qualitatively match the real CT domain. The ASTRA toolbox [27] is a library that provides GPU-enabled capabilities to generate simulated X-rays with flexibility to set options such as detector resolution, X-ray source type, reconstruction algorithms, and physical dimensions that define the geometric setup to match the fielded CT

Fig. 2 Slice of simulated one-material example with yellow representing the metal phase and dark purple representing the pore phase (left). Slice of synthetic CT scan of this example generated with ASTRA (right). (Color figure online)

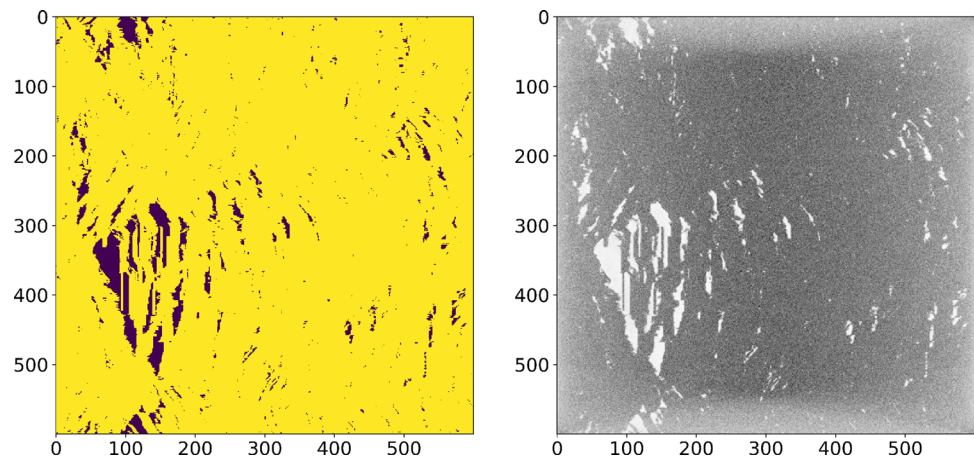
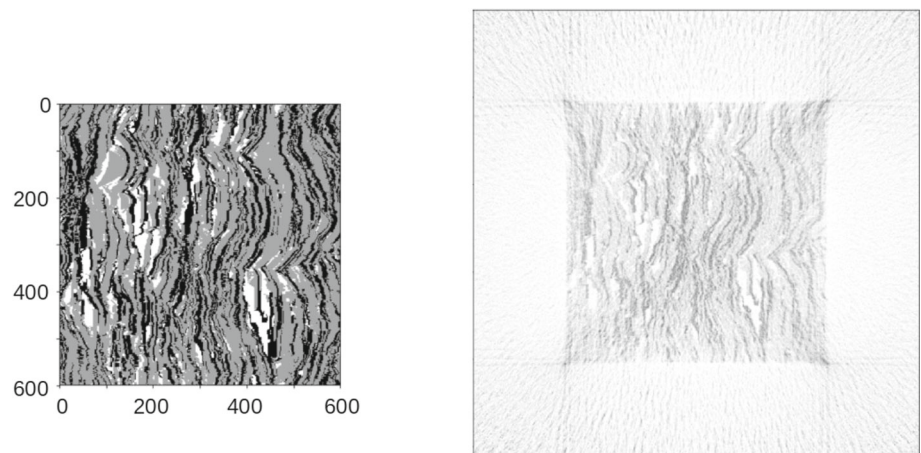


Fig. 3 Slice of synthetic two-material example where distinct metal phases are represented in gray and black, and the pore phase is represented in white (left). Slice of synthetic CT scan generated for this (padded) example before cropping (right). (Color figure online)



machine used to generate scans of the materials that are the subject of this work.

ASTRA takes parameters that define the physical setup of the CT machine to be simulated. We experimented with various parameters to qualitatively match features like resolution, feature scale and noise we observed in the real CT data. Geometric parameters were set to use 1800 cone beam projections with a 1536×1536 pixel detector with spacing of 4 units between adjacent pixel centers over a full 360-degree rotation of the synthetic object. The synthetic object was set to have a 1:3 ratio between the distance to the simulated X-ray source from the object and the distance from the detector to the object. The FDK_CUDA algorithm from the ASTRA library reconstructed the volumetric CT image of the object from the synthesized X-rays, and was executed on the GPU.

To ensure the ground truth material volume would be encapsulated in the imageable area of the simulated CT framework, the volume was padded with zero values along all axes, then the simulated CT scan was generated for each example. Several post-processing steps were taken to transform the synthetic scans to have voxel intensity values similar to the distribution found in the real CT domain. Voxel values were limited to the range $[0,255]$ by first replacing all

negative values within the synthetic CT scan volume with zeros then dividing by the maximum value in the volume, multiplying the result by 255 and rounding to the nearest integer value. Next, the volume was cropped to remove the zero padding needed for the CT simulation. Finally, noise was introduced to the volume via elementwise multiplication with a volume of randomly generated floating point values sampled from a normal distribution with mean 1 and 0.1 standard deviation. All volumetric post-processing leveraged the Numpy python library [28]. The simulated CT framework was implemented on an NVIDIA DGX-2 machine with 32 GB GPUs and approximately 1.5 TB of system memory.

Examples of the simulated CT scans of one-material and two-material simulated objects are shown in Figs. 2 and 3.

3.1.2 Labeled CT data

X-ray computed tomography experiments were carried out at the Advanced Photon Source on several dozen samples, spanning both 1-metal and 2-metal formulations, as well as atmospheric (AS) and cold spray (CS) processing conditions. For details of sample preparation and X-ray characterization, the interested reader is referred to [29]. As discussed

previously, the primary goal of X-ray image segmentation is to separate porous regions from solid metal-filled regions in the CT scans for purposes of quantifying pore morphology. Porous regions (whether gas-filled or vacuum) have very low X-ray attenuation and result in low grayscale intensity/dark regions in reconstructed images; conversely, high density/high attenuation metal regions result in much higher grayscale intensities, and appear as light regions in reconstructed images.

Despite this large difference in grayscale intensity, two key factors result in significant challenges for image segmentation: first, pore space features are occasionally comparable to the X-ray CT imaging resolution ($< 1\mu$), which leads to voxels that contain both pore and metal phases, resulting in intermediate grayscale intensity values. Second, the presence of metal oxides in the samples results in regions that likewise have intermediate intensity values, further complicating the identification of pore and solid phases. In particular, during atmospheric plasma spray processing, which takes place in a high-temperature, oxygen-rich environment, a variety of metal oxides are formed in both one- and two-material samples. These metal oxides have lower X-ray attenuation coefficients in comparison to the pure metals, but also higher attenuation in comparison to porous regions, resulting in grayscale values intermediate between pore and metal. In the following section, we provide additional details along with pertinent example images of these scans to illustrate these challenges in the context of image segmentation.

Several recent advances in computer vision, such as those described in Sect. 2.2 herein, have led to deep learning methods being overwhelmingly adopted as the standard approach to image segmentation in the literature [4]. Specifically, the most successful segmentation models have been trained in a supervised fashion.

In this work, we have no ground truth examples of perfect segmentations of the CT scans that are the target of the project. Our domain of interest is a set of real CT scans of two types of thermal spray materials. The first steps of our approach produce training sets such that the deep learning model can learn with synthetic examples that provide ground truth that is close to our domain of interest as well as two sets of viable but imperfect labels of the real target domain. The remaining steps in the process are training and inference typical of deep learning applications. In this section, we provide a summary of each of these steps. We present detailed descriptions of the methods in Sect. 3.2.1.

In addition to the synthetic training dataset, we used two algorithmic techniques to generate segmentation examples of the real CT scans from our target domain that we used to train the deep learning model. Ideally, to generate ground truth labels for a segmentation training dataset, a human would be tasked with labeling each voxel in a CT scan by the type of material the voxel represents. For our target domain, the

real CT examples comprise on the order of 1 billion voxels, presenting an impractical task for a human to label manually. As such, we have leveraged two algorithmic approaches to generate segmentations that are useful for the deep learning algorithm to learn how to identify features in the real data relevant to a successful segmentation. The use of these flawed segmentations of real CT examples gives the network exposure to the real domain of interest.

CT examples

Figure 4 shows example image slices from a selection of CT scans of one-metal air-sprayed (AS) and cold-sprayed (CS) samples, as well as two-metal AS samples. Images include a significant portion of the full field of view, as well as magnified regions to show detailed features of interest. The darkest regions (intensity $< \sim 20$) correspond to porous regions. Intermediate gray values correspond to metal oxides in 1-material regions, as well as a combination of metal oxides and one of two metals in the two-metal case. In all cases, the brighter gray regions correspond to pure metal. The precise intensity values of each of these phases are not known, and likely overlap significantly as discussed above, which makes image segmentation a challenging task.

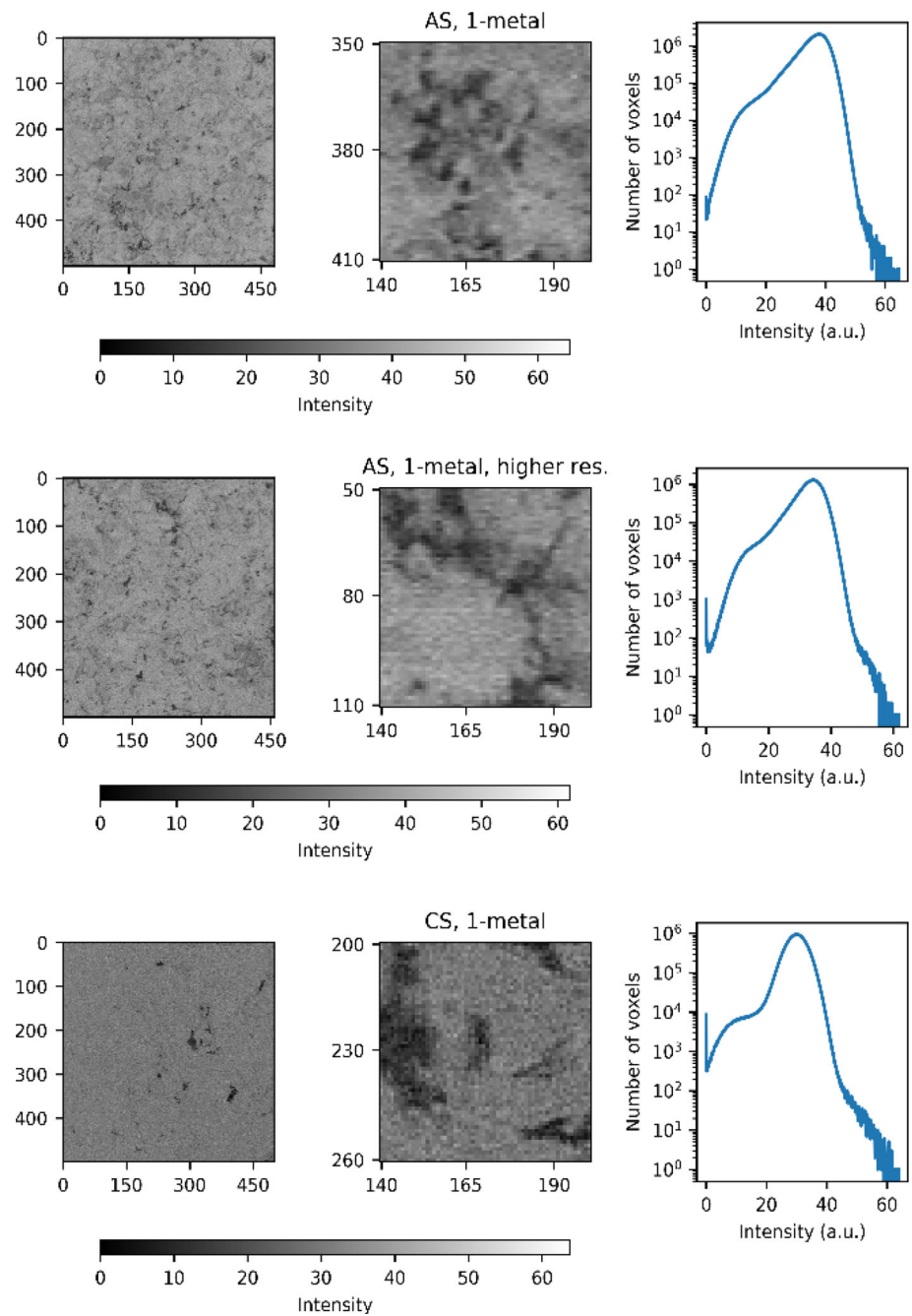
We have initially focused efforts on separating pore space from any solid phase (i.e. oxide or metal), but separation of oxides from metal, as well as separation of different metals in two-metal samples is also of interest for future work. For all examples in Fig. 4, a histogram of all intensity values is also shown. While in some cases there appears to be a separation of the pore space intensity (a trough in the histograms at low intensity values), thresholding images based on the intensity at the bottom of this trough does not produce satisfactory results. Likewise, when separating intermediate and dark gray regions (oxides and/or different metals), the separation is even more ambiguous, with clear overlaps between the two intensity ranges. These considerations ultimately motivated us to pursue more sophisticated segmentation techniques than typical global thresholding techniques that establish a global threshold based on features of the histogram.

Thresholding labels

We denote as “thresholding” a labelling method that selects a global threshold on intensity based on analyst judgment and/or known pore and material volume fractions. An analyst used the following procedure for the thresholding in this work:

1. Proposed a threshold value
2. Segmented the image based on that value
3. Compared the grayscale and proposed segmented images by eye and evaluated how reasonable the segmentation looked

Fig. 4 Example image slices from CT scans of various samples. Left panel in each subfigure: significant portion of the field of view, excluding sample edges. Middle panel: magnified view of selected region of image in left panel (axis labels indicate region). Right panel: Histogram of grayscale intensity values. (Color figure online)



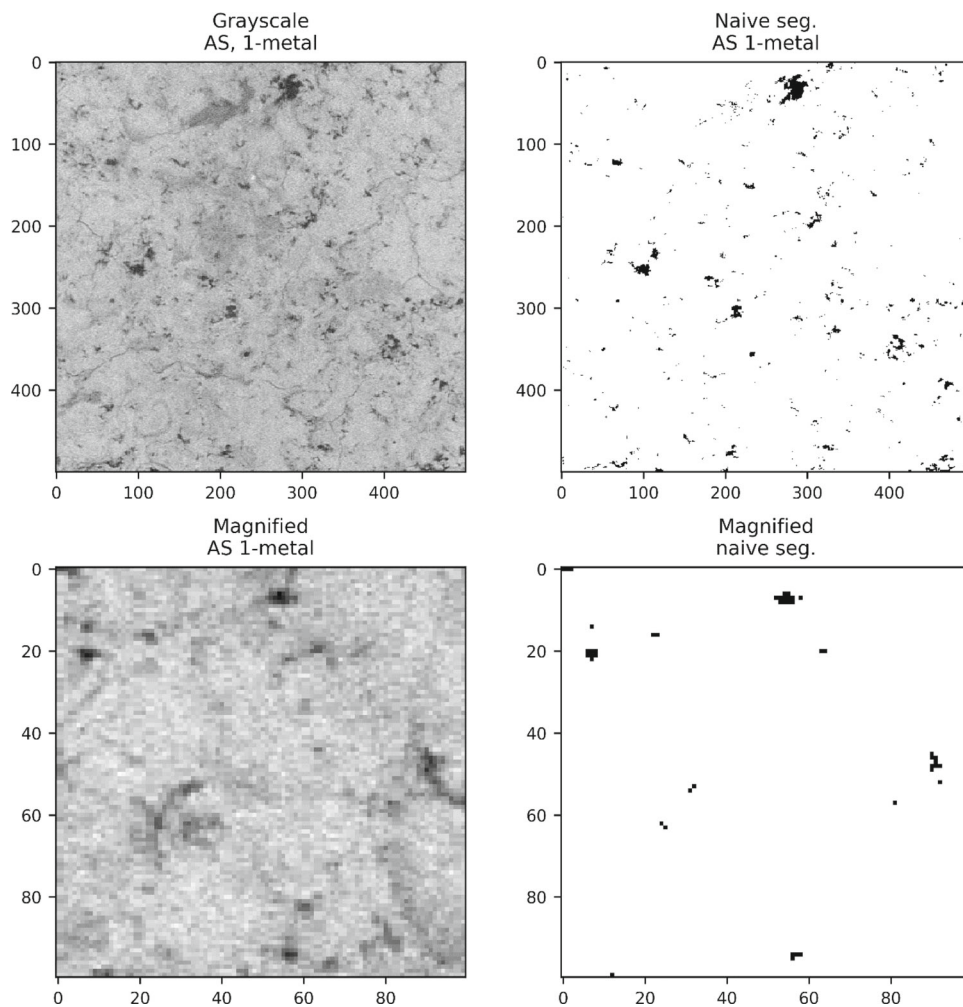
4. Iterated on this process until thresholding values were settled on what appeared to produce a reasonable segmentation

A particular trait used by the analyst in this process was to select thresholding values that tended to minimize the amount of “speckling” in regions that were likely to be all pore or a particular metallic phase. If a threshold value is chosen larger than it probably should be, small clusters of the less-dense material show up like speckles in regions that are likely the higher-density material. Likewise, if a threshold value is cho-

sen smaller than it probably should be, small clusters of the more-dense material show up like speckles in regions that are likely the lower-density material. Resulting volume fractions were then checked to ensure they were within plausible ranges based on subject matter expert estimates.

Another possible application of this method would be to select thresholding values that yielded volume fractions ascertained or supposed based on external information or supposition. Intensity values could then be sorted in ascending order and the value found and selected that corresponds to the volume-fraction percentile of interest (e.g. 6%).

Fig. 5 Examples of grayscale image slices and corresponding results of threshold segmentation for 1-metal samples. (Color figure online)



As this approach is based on a global threshold, it does not consider any local pixel information, which results in erroneously noisy features. This is apparent in Fig. 5, particularly in the magnified view at the bottom. Furthermore, the target volume fractions are not generally known with high accuracy; instead, it would be highly valuable to obtain an estimate of these volume fractions (particularly porosity) based on analysis of the CT scans, rather than imposing them as part of the segmentation process.

Random walk algorithm labels

We have also attempted a more sophisticated segmentation approach that takes advantage of local pixel information, as developed by Grady [2]. This algorithm starts with assigning phase labels to a subset of voxels for which phase labels are known with high certainty, which we refer to as ‘seed’ labels. Conceptually, segmentation then proceeds by placing random walks on all of the remaining, unlabeled voxels, and allowing them to carry out a random walk, where the likelihood of a move to a neighboring voxel is inversely proportional to the difference in grayscale intensities of those neighboring voxels. The probability of a given voxel belong-

Table 1 Percentage of voxels (per example) where thresholded and random walker examples agree

One material	Two material
98.95% (0.16%)	92.24% (1.43%)

ing to each phase is then assigned according to the probability that a random walk beginning on the given voxel first reaches one of the seed label voxels corresponding to that particular phase. Computing this probability amounts to a linear algebra problem, so the random walk simulations do not need to be carried out directly. Additional details are provided by Grady [2]. In all random walk segmentations, we have used the open-source Python implementation in the scikit-image library [30].

Seed labels for random walk segmentation were assigned following an approach similar to the threshold segmentation, but with percentiles chosen very conservatively to ensure high certainty in the assigned voxels. The 1st percentile of intensity was used to assign seed labels for the pore phase,

and the 60th percentile was used to assign labels to the solid phase. All voxels with intermediate intensity values were labeled via the random walk segmentation described above. Figure 6 shows the resulting segmentations for the same slices as the threshold segmentation in Fig. 5. In the case of two-material samples, similarly conservative percentiles were used in assigning seed labels to a small subset of the image. As seen from the images, the resulting segmentations contain much less noise/single-voxel features, which likely represents more meaningful segmentations. More importantly, the algorithm does not require a known porosity as an input, but rather provides an estimate of this important quantity based on the image segmentation. However, there are occasional pore regions that are completely missed by this segmentation, as seen in the magnified view of Fig. 6 (bottom right panel). We attribute this to an inadequate assignment of seed labels, wherein the random walk algorithm requires at least one voxel to be assigned as a seed label for each topologically connected pore region. With the conservative seed labeling strategy used here, this occasionally fails for very small pore regions, where partial voxel effects result in higher relative values of intensity. Nevertheless, the segmentations produced by the random walk algorithm are generally satisfactory and expected to provide good training data for the machine learning algorithms deployed herein. Figure 7 shows an example of a random walk segmentation of a 2-metal sample.

Label comparison

To understand the variance between the thresholded and random walker labels, we have quantified the percentage of voxels where the labels agree over the datasets. Table 1 shows the level of agreement for both the one-material and two-material datasets.

3.2 Deep learning model architecture

In our experiments, we use a modified V-Net [3], implemented in Keras [31] with a TensorFlow [32] backend taking a volumetric input chunks of size (240, 240, 240) and producing a softmax output vector map of size (240, 240, 240, c), where c is the number of output classes within each softmax output vector. A sketch of the architecture is shown in Fig. 8. The V-Net consists primarily of four downsampling blocks and four upsampling blocks, with skip connections between the corresponding downsampling and upsampling blocks. At each block, the input undergoes a series of three-dimensional convolutional operations with unit stride length, same padding, and a fixed number of input channels for all layers in the block, followed by a final operation serving to either downsample or upsample the input. In the case of the downsampling block, this final operation is a convolutional operation using a kernel size of 2 and a stride of 2, reducing

the input size in all dimensions by two and doubling the number of input channels. In the case of the upsampling block, this final layer is a transposed convolutional layer using a kernel size of 2 and a stride of 2, upsampling the input size in all dimensions by two and reducing the number of input channels by a factor of two. For upsampling, the corresponding skip connection is added just prior to the final transposed convolution. Each convolutional and transposed convolutional operation in the upsample blocks is followed by a three-dimensional spatial dropout operation, which deactivates a random fraction of feature maps during each forward pass.

The dropout layers remain active during inference to add stochasticity to the model to perform uncertainty quantification, as in Gal et al. [25]. We use the model to perform several rounds of inference on a single, fixed volumetric input chunk, after which we take the set of output maps and compute the mean and standard deviation maps. The former gives us an improved, reliable output map estimate from which class predictions may be calculated, and the latter gives us an uncertainty map, yielding estimates of the model's uncertainty at different voxel locations within the image.

3.2.1 Neural network training and inference

We separate training of our model into two primary phases. For our first phase of training, we train the model using synthetic examples. For our second phase of training, we train the model on real world CT data, for which ground truth segmentations are not available, instead being estimated using two different approaches: (a) thresholding and (b) the random walk algorithm. This two-phased approach is a form of transfer learning: we begin training on real data only after establishing good initial weights by training on synthetic data with available ground truth.

As is often the case with CT data, the size of our datasets is such that it is impractical to keep an entire scan in GPU memory at once. During training, we randomly select a number of samples from each image of size (240, 240, 240). During inference, we partition each input volume into evenly spaced, overlapping chunks of the same size, with a stride length of 208 in each direction. Overlapped areas use average values between neighboring chunks. We infer on these chunks individually, performing multiple forward passes through the CNN with active dropout, and calculate the mean and standard deviation of each voxel's softmax prediction.

We use the negative log likelihood loss, weighting by the logarithm of the inverse frequency of each class to account for class imbalance within each chunk. This encourages accurate predictions of the minority classes. For optimization, we use Keras' [31] implementation of Adadelta [33] with a learning rate α of 0.001 and a decay rate ρ of 0.95.

Our training protocol for each dataset used both dropout and early stopping for regularization. In each case, we

selected the model with the best validation loss to evaluate our results.

Our implementation is available publicly [34], and we have observed robustness in the preprocessing and training methods such that without modification, the model has been successfully applied to several domains [6].

3.3 Analysis of pore characteristics

In order to analyze pore characteristics via different morphological metrics, segmented three-dimensional images are required. Translating per-voxel uncertainty outputs of the model to error bounds in various pore characteristics is an ongoing area of research, but we adopt a relatively simple approach here to obtain conservative (that is, very loose) bounds on these pore characteristics. For a given three-dimensional (grayscale) image, the CNN segmentation provides ten distinct inferences, corresponding to ten distinct randomly selected dropout configurations. In each inference, a softmax function value is assigned to every voxel for every phase label, where the softmax value is between 0 and 1,

indicating a relative likelihood of a given voxel belonging to each phase. The mean and standard deviation of the values of the softmax output associated with the pore phase class label are then computed at each voxel across these ten inferences. These mean and standard deviation values are denoted as \bar{s} and σ_s . The value of σ_s is associated with a relative level of uncertainty. The most likely segmentation, which we denote as the “base case”, is generated by simply taking the argmax over the softmax outputs for each voxel, i.e. each voxel is assigned to the phase with the highest softmax value. To generate additional segmentations based on the associated uncertainty σ_s , a value of σ_s is first chosen to denote voxels with moderately high segmentation uncertainty. This choice of $\sigma_s > 0.02$ is somewhat arbitrary, but empirically was found to correspond to the value of σ_s that captures the vast majority of voxels that result in different class labels across any of the ten inferences of the CNN (i.e. the majority of voxels that change labels across any of the ten inferences satisfy $\sigma_s > 0.02$). To generate segmentations that represent bounds on the base case, all voxels that satisfy $\sigma_s > 0.02$ are set to the pore phase, resulting in a porosity

Fig. 6 Examples of grayscale image slices and corresponding results of random walk segmentation for 1-metal samples. (Color figure online)

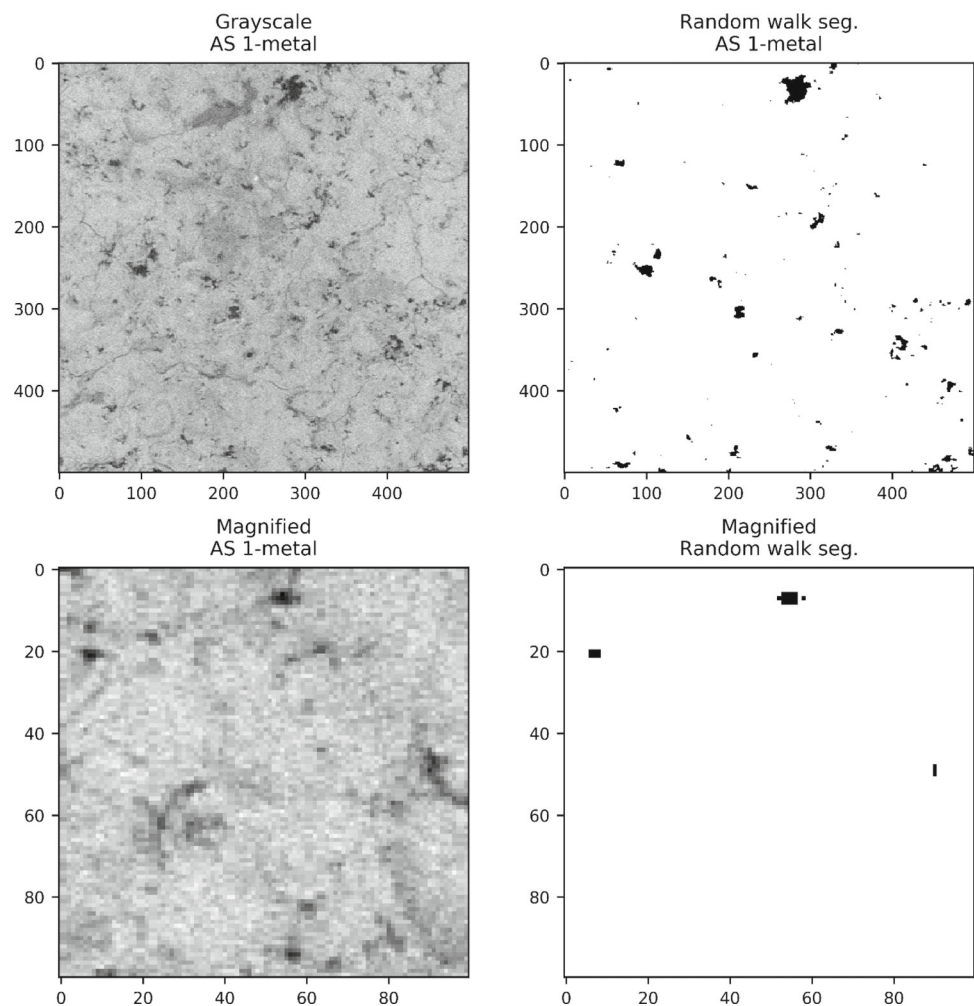
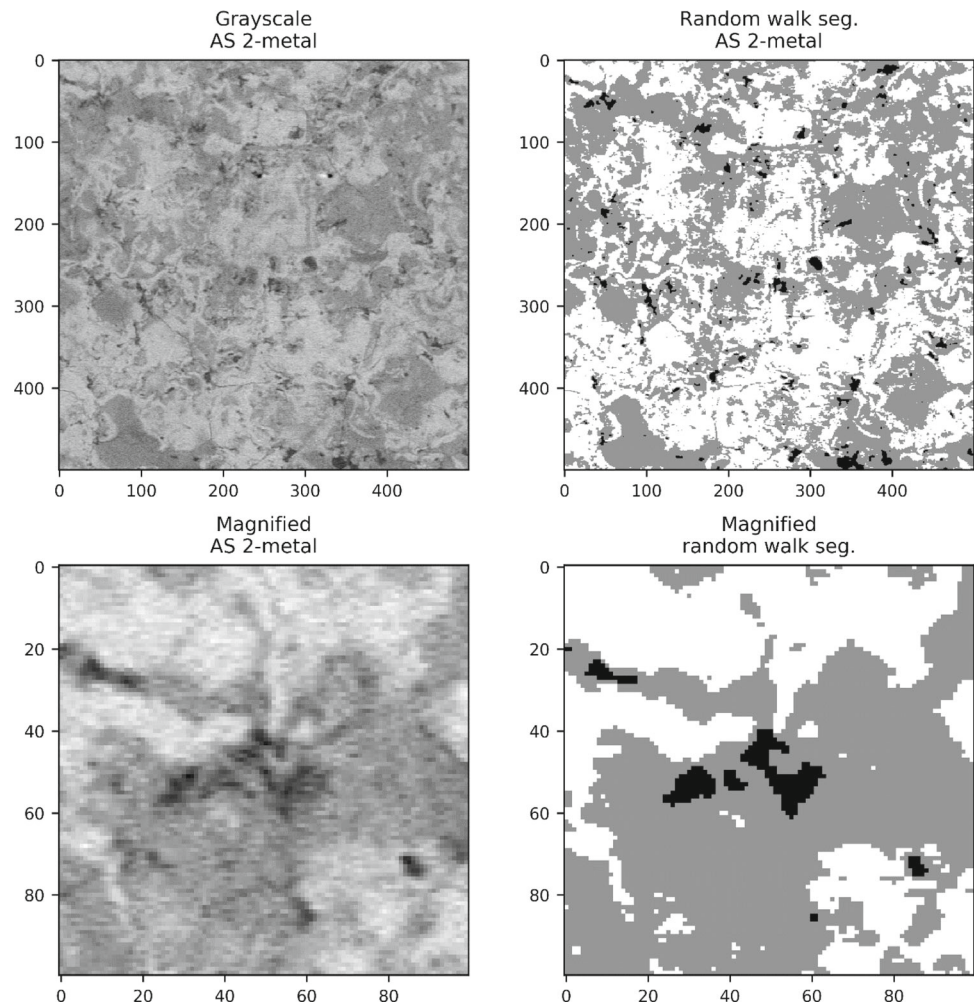


Fig. 7 Examples of grayscale image slices and corresponding results of random walk segmentation for 2-metal samples. (Color figure online)



“upper bound” segmentation; conversely, setting all voxels that satisfy $\sigma_s > 0.02$ to the solid phase produces a porosity “lower bound” segmentation. Figure 9 depicts the quantities $\bar{\sigma}$ and σ_s , as well as the base case and lower and upper bound segmentations for a sample two-dimensional slice of a three-dimensional dataset.

With the low, base and high porosity segmentations as bounds, we compute several metrics that quantify the pore morphology. In the present context, the nature of these metrics and their physical implications are not discussed in detail; the interested reader is referred to other works for this discussion [10]. Here, these metrics are only presented to illustrate quantitative differences arising from image segmentation.

First, we compute the two-point correlation function $S2(r)$, which is a common measure used to describe the spatial distribution of a component within a heterogeneous material. It can be interpreted as the probability of two points that are separated by a line segment of length r randomly placed in a volume both belonging to the pore phase [35, 36]. Given the symmetry of the samples, we plot this metric as a function of $r = z$ defined as the distance in the spray direction, as well

as a radial distance r in the plane perpendicular to the spray direction (denoted as non-spray direction). The value of $S2$ at both $r \rightarrow 0$ and $z \rightarrow 0$ is the porosity, i.e. the total fraction of the volume occupied by pore space. Figure 10 summarizes the results for this metric in both the spray and non-spray directions. Lines for each segmentation bound correspond to the mean value across thirty scans, and error bars correspond to the standard deviation across the same scans. Additional details are provided by Rodgers et al. [11].

Another metric of interest is the pore size distribution (PSD). Since identifying individual pores in these microstructures is difficult and ambiguous, we adopt a more general definition of pore size distribution that relies on identifying the fraction of the volume that can be ‘swept out’ with spheres of a given size, such that the spheres are wholly contained in the pore phase. This approach was originally suggested by Munch and Holzer [37] and provides a general measure of the distribution of the local pore space size that does not rely on arbitrary definitions of individual pores. Since spatial correlation is closely related to pore size, the information in this PSD metric is similar to the two-point

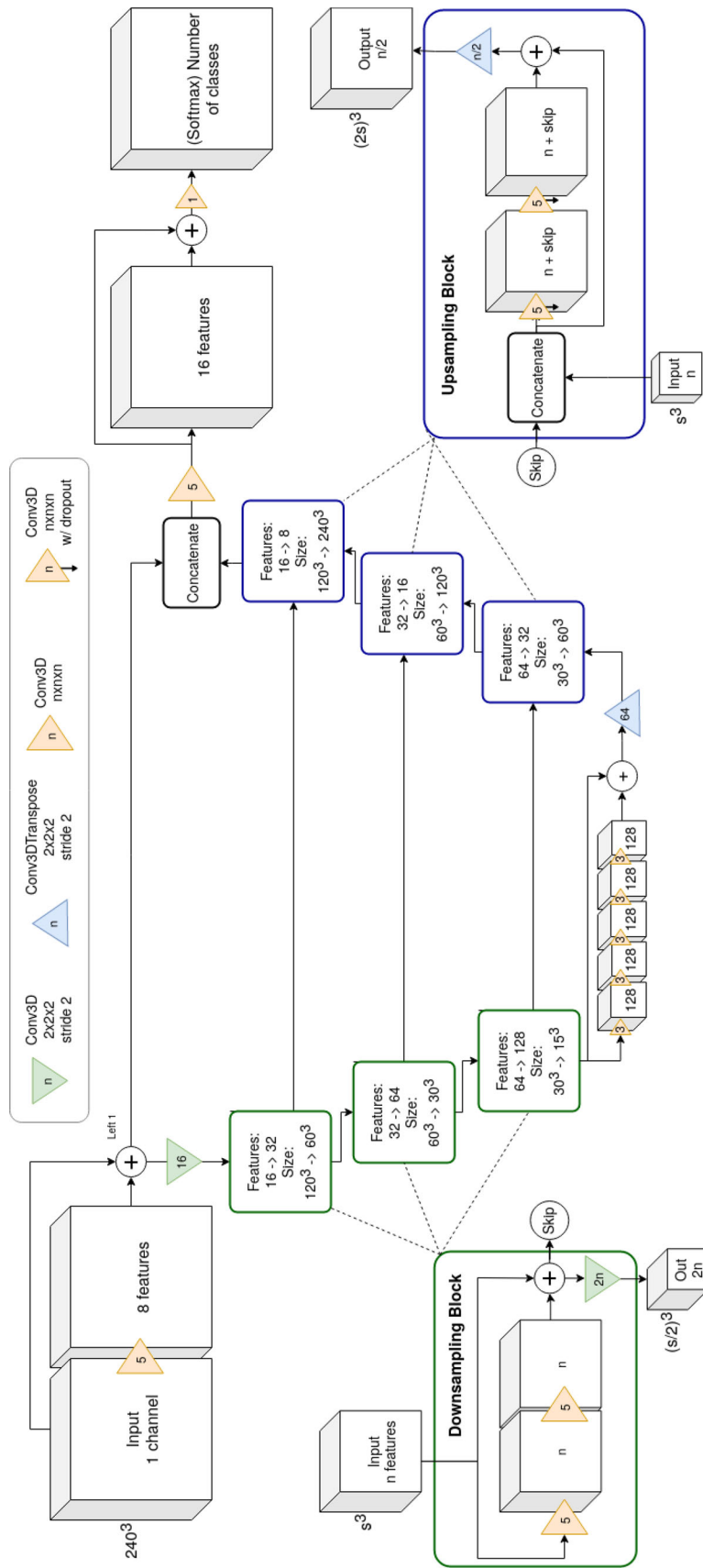


Fig. 8 Sketch of CNN architecture

Fig. 9 Example segmentations of a one material slice, along with low and high bounds and $\bar{\sigma}$ and σ_s

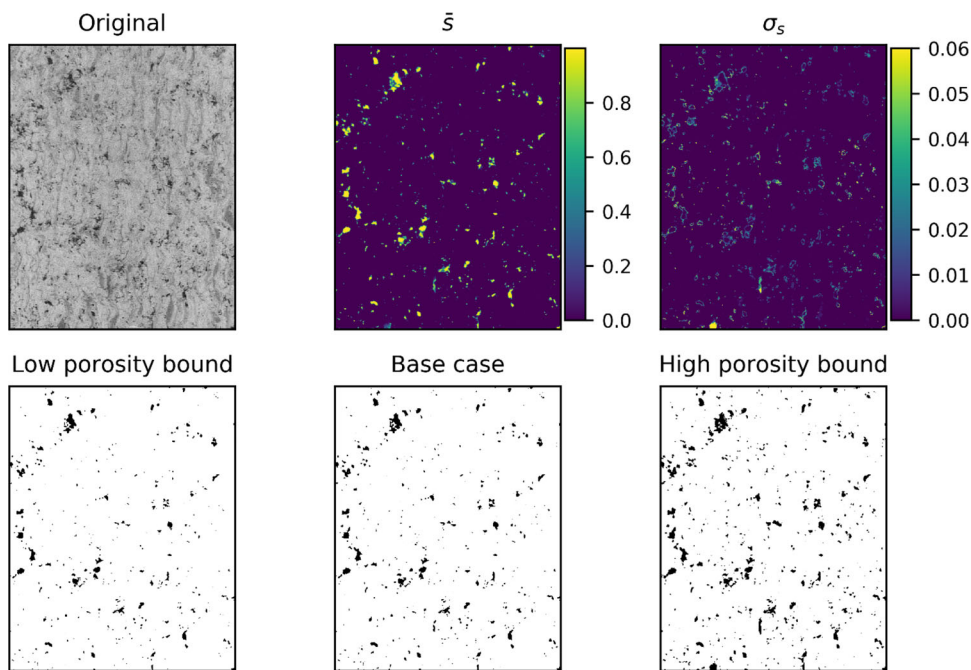
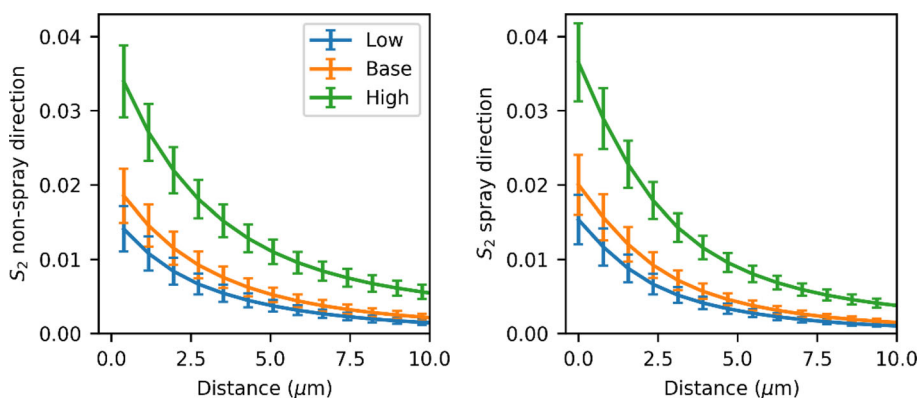


Fig. 10 Summary of pore space two-point correlation function



correlation function S_2 ; however, the pore size distribution has a slightly different focus on local size, as compared to local correlation. The results for the PSD are summarized in Fig. 11, with lines corresponding to mean values across thirty scans, and error bars corresponding to the standard deviation across these scans.

Finally, we quantify pore space connectivity and topology based on a metric suggested by Hilfer et al. [38, 39]. For a given scan, many cubic subvolumes of size L are randomly selected, and the fraction of subvolumes for a given choice of L that contain a percolating pore phase in each direction is computed. In each subvolume, percolation in a given direction is simply determined based on the presence of at least one connected cluster that spans the subvolume in that direction. For purposes of defining connected clusters, a simple connected component labeling algorithm [40] is applied to each

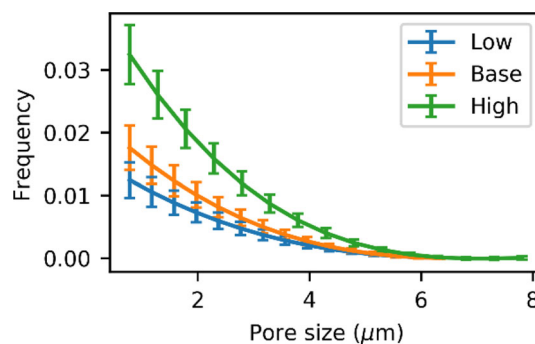


Fig. 11 Summary of pore size distribution

subvolume. Results are summarized in Fig. 12 as probability of percolation in each direction as a function of subsample size.

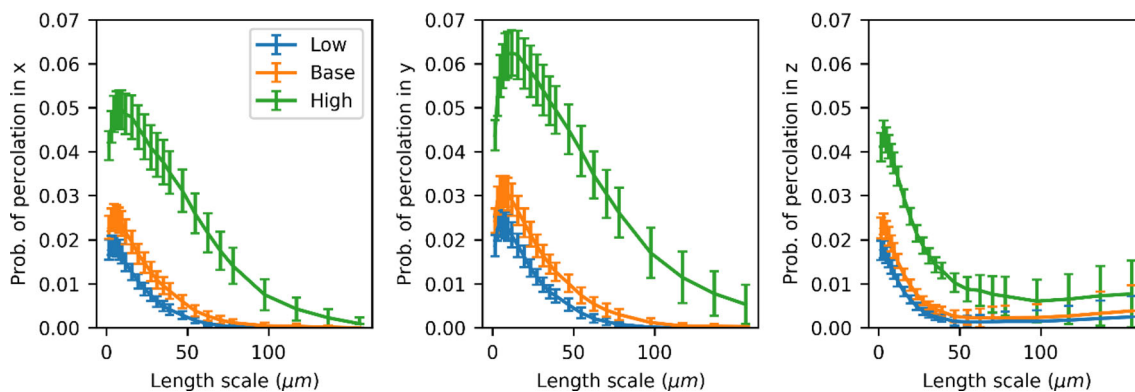


Fig. 12 Probability of percolation as a function of length scales in all three directions. The z direction corresponds to the sample spray direction

4 Results

We present CNN segmentation results from the various training methods using distinct combinations of real and synthetic one- and two-material examples as training data.

4.1 One material thermal spray data

We performed experiments and gathered metrics for models trained on thermal spray scans consisting of two classes per voxel—material and pore. We train NN models on different combinations of synthetic and real examples. For synthetic CT examples, simulated “ground truth” segmentations are generated alongside the corresponding examples. For real CT examples, segmentations are created using a random walk segmentation algorithm. Once our models are trained, we assess the quality of our model’s segmentations across each case and present metrics comparing the NN generated segmentations to the labels used for training.

First, we trained a model solely on real examples and labels generated via random walk segmentation. We used twenty-three scans for training, two scans for validation, and held five scans out for post-training inference and analysis. Figure 13 shows a sample cross section of a held out test example, along with the corresponding segmentations and uncertainty estimates.

Next, we trained a model on synthetic one material CT scans and their respective generated ground truth labels. We used eleven scans for training, two scans for validation, and held nine scans out for post-training inference and analysis. Figure 21 demonstrates model performance for a sample cross section of a test example.

Lastly, we employ a combined method. Initially, we train the model on synthetic examples. We then perform extra training on real data with random walk labels to refine the NN weights with respect to real examples. Figure 14 shows an example of output from our full, combined method. Table 2

Table 2 Mean results from different training methods over the one material test set

Training labels	Accuracy to RW (%)	False positive (%)	False negative (%)
RW	98.70	0.38	28.02
Synthetic	94.88	4.58	22.25
Synthetic + RW	98.80	0.55	20.27

Accuracy is reported with respect to the random walk (RW) labels

Table 3 Mean results from different training methods over the two material test set

Training labels	Accuracy to Threshold (%)	Accuracy to RW (%)
Threshold	98.15	93.08
RW	91.90	95.32
Threshold + RW	97.50	94.50
Synthetic	86.86	87.91
Synthetic + Threshold + RW	92.5	95.00

Accuracy is reported with respect to both the threshold labels and the random walk labels

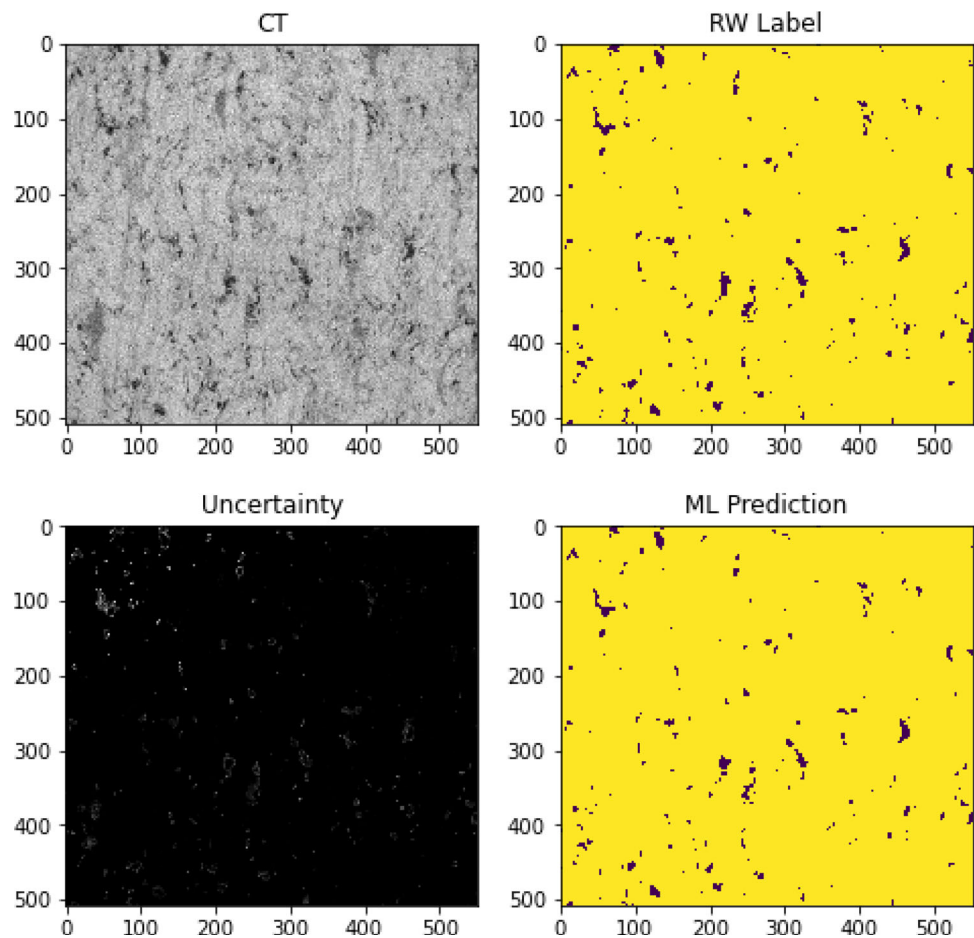
shows a quantitative comparison of the NN predictions to the random walk labels.

4.2 Two material thermal spray data

We trained models on thermal spray scans with two metallic phases in addition to the pores—three classes per voxel in total. We train models on different combinations of the three labeling methods (synthetic, grayscale intensity thresholding, random walk), after which we assess the quality of the generated segmentations in each case.

We trained a model on real CT scan examples and grayscale intensity thresholded labels. We used two scans for

Fig. 13 Sample cross section of held out CT scan (top left), associated random walk label (top right) and our model's predictions (lower right) and corresponding uncertainty map (lower left) for one material model trained on random walk labels



training, one scan for validation, and held one scan out for post-training inference and analysis. Figure 15 shows sample cross sections comparing these results, and Table 4 shows the accuracy of each segmentation method on the held-out test example. We tried additional combinations that achieved similar results (a random walk trained model, a synthetically trained model, and a combined naive / random walk model). Results are included in Appendix A. Finally, we trained a model using a full, combined method. Initially, we train the model on the two material synthetic examples used previously, after which we continue training on real data with both thresholded and random walk labels. Figure 16 shows a test output example.

5 Discussion

In this section, we present a comparison of the results from our CNN model to the corresponding labels, and discuss the impact of uncertainty estimation on analysis of relevant quantities of interest.

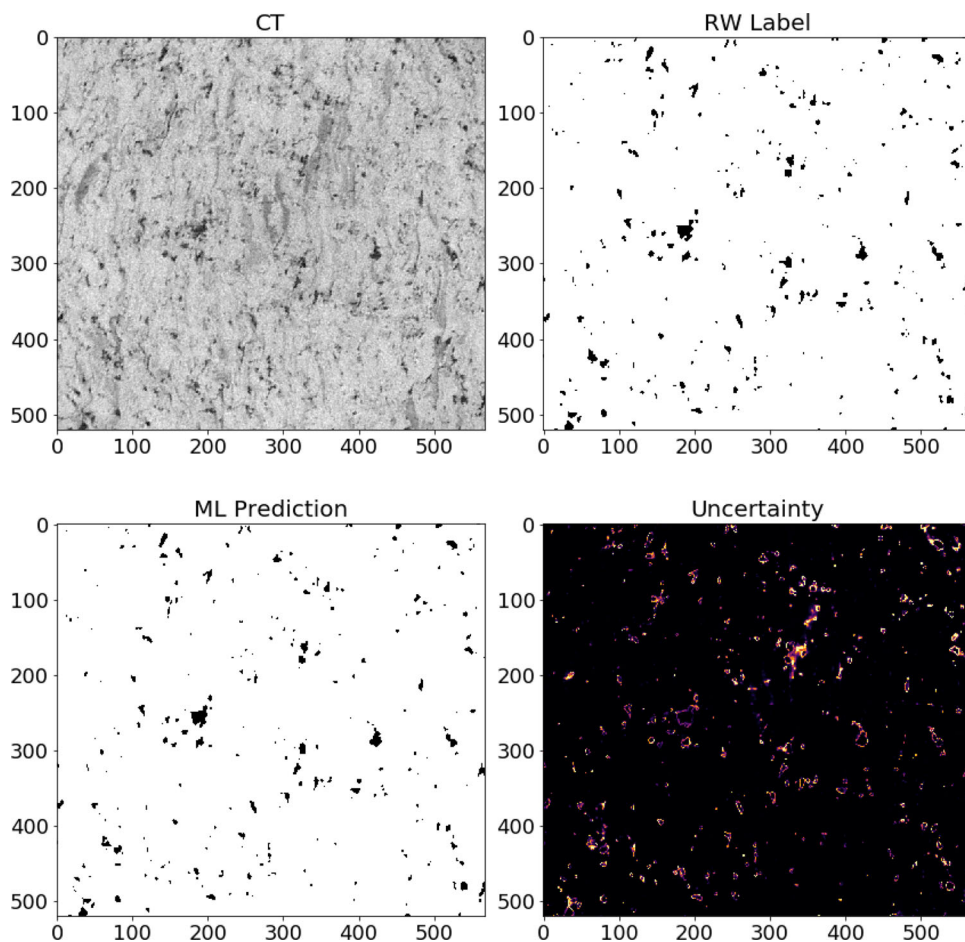
5.1 Comparison of deep learning predictions with generated labels

Herein, we compare the prediction maps of our model to those of the three methods by which the training labels for each model were originally generated.

Across all experiments, we observe that our models produce viable segmentations of the original image, though there are some obvious differences between models. Some of the most prominent differences originate from the models trained on synthetic data alone. In these cases, the model tends to predict a higher proportion of pore voxels in comparison to other approaches. This discrepancy highlights the differences between real and synthetic examples, and points to the importance of training on real segmentations in order to ensure that the resulting outputs are accurate and underlying physical characteristics remain consistent. Notably, although the two-material synthetically trained model tends to over-predict pore voxels, the metallic phase segmentations appear to match the segmentation produced by the random walk algorithm relatively closely.

The two-material model trained on thresholded labels produces pore segmentations that more closely match the

Fig. 14 Sample cross section of held out CT scan (top left), associated random walk label (top right) and our model's predictions (lower left) and corresponding uncertainty map (lower right) from our full method on a one material example



corresponding random walk labels, but for larger pores, these predictions tend to be noticeably spottier in comparison to the random walk labels used for comparison. This illustrates some of the same shortcomings the intensity thresholding algorithm possesses—namely, the lack of local context in the resulting segmentations. In comparison, models such as those trained via the full combined methods whose training sets are comprised of the random walk segmentations tend to avoid these issues and produce segmentations of visually higher quality.

We analyze the complexity of the predictions our model is making and determine whether our model is performing grayscale thresholding or something more sophisticated (as in the random walk segmentation case). We took a two-material example and constructed a histogram of the grayscale values and the corresponding classes for: (a) the thresholded label, (b) the random walk label, and (c) the label produced by our full combined method's two-material model. In Fig. 17, we observe that the class distributions for the pipelined model overlap—providing evidence that our model's decision-making process is more complex than simple grayscale intensity thresholding.

We now estimate each model's performance based on the average accuracies with respect to the random walk labels across each experiment's testing set. Among the one material examples, the full pipeline model outperforms the real data model with respect to the RW labels slightly, with an accuracy of 0.9880 compared to the former's accuracy of 0.9870. Another point of note is the porosity, for which there is far less of a disparity between the random walk labels and model predictions in the pipeline case (0.231 and 0.222, respectively) than in the real data case (0.0281 and 0.226, respectively). Among the two material examples, with an accuracy to the random walk labels of 0.96 across the testing set, followed shortly by the model trained on random walk labels, with an overall accuracy of 0.9532. Qualitatively, visual inspection suggests that the full pipeline method that used both synthetic and real, but imperfect, training data provides the best segmentation that appears to reduce some of the error seen from the random walk.

In order to examine the role that uncertainty plays in our models, we calculate the average uncertainties for voxels in which the predicted classes were either correct or incorrect with respect to the labels using the one material (Fig. 18a) and two-material synthetically trained models (Fig. 18b).

Fig. 15 Sample cross section of held out CT scan (top left), associated random walk label (top right) and our model’s predictions (lower right) and corresponding uncertainty map (lower left) for two material model trained on thresholded labels

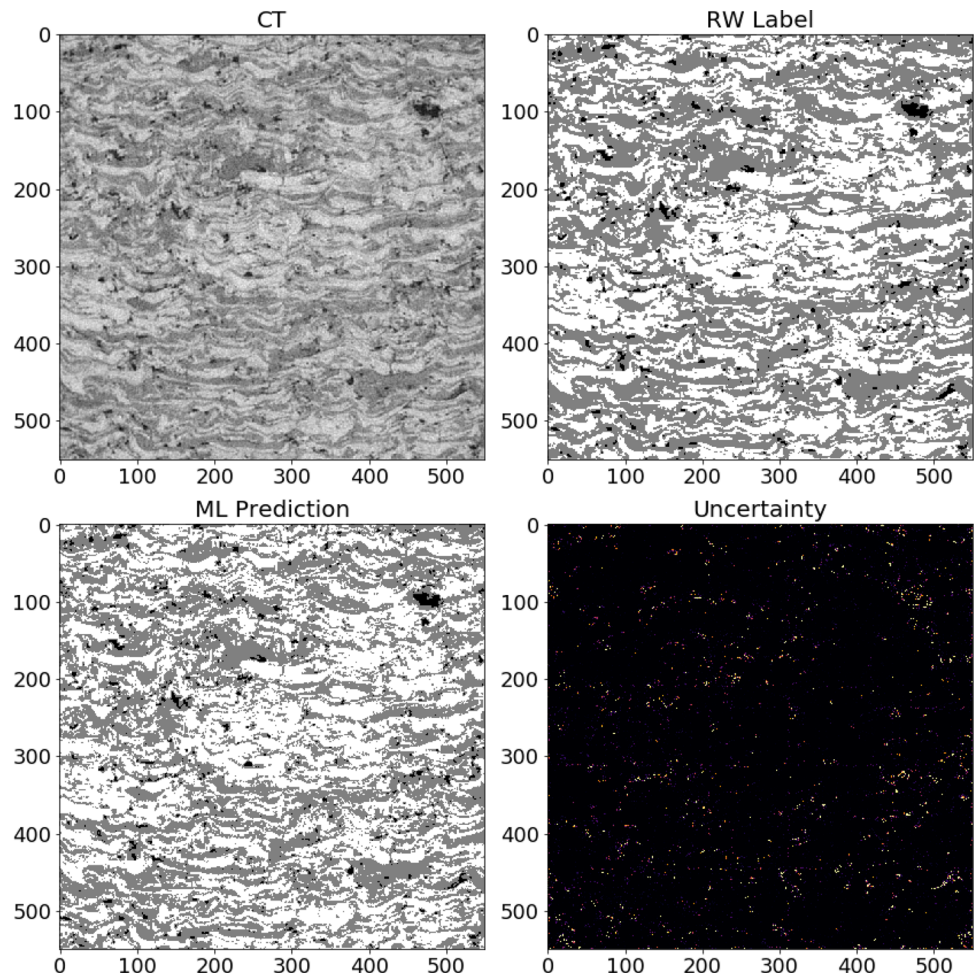


Table 4 Accuracy of threshold, random walker, and convolutional neural network segmentation of synthetic two-material test example

Method	Accuracy (%)
Threshold	85.05
RW	85.98
CNN	91.35

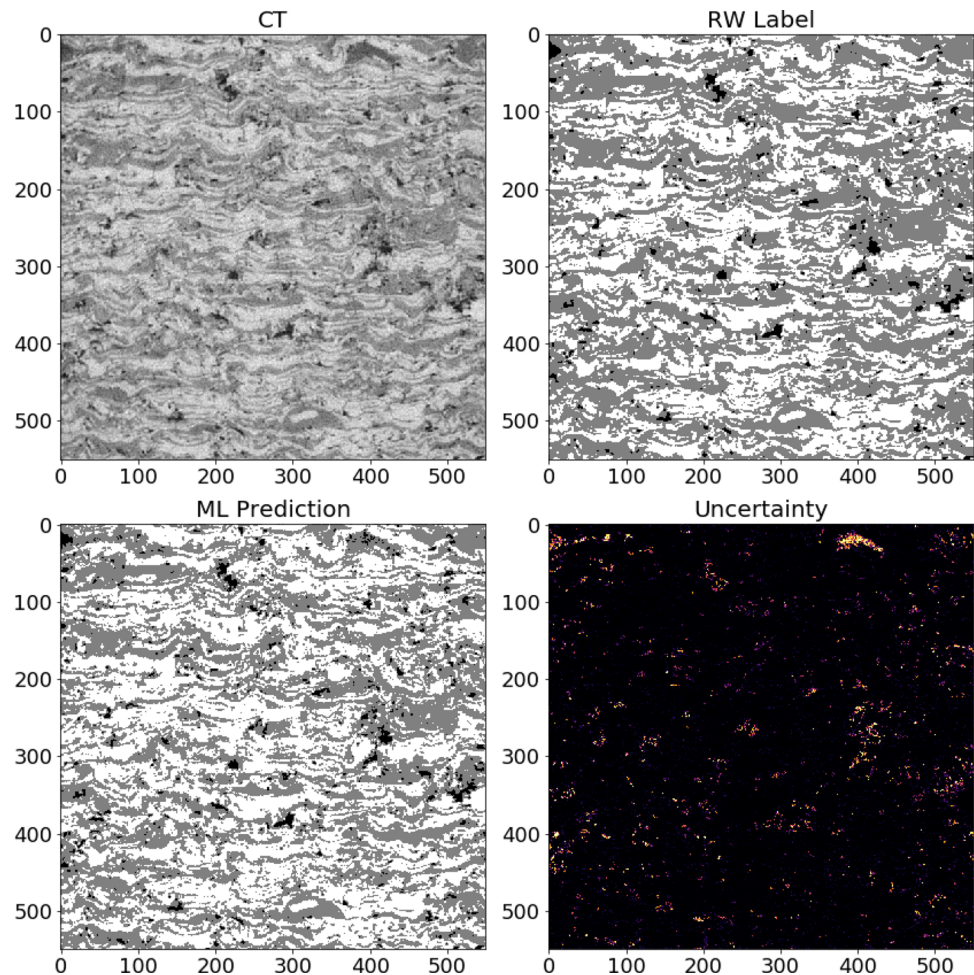
In both cases, we observe that on average, uncertainties tend to be higher in incorrectly predicted regions than correctly predicted regions. We denote a voxel within a volume uncertain if $v_u > \mu_u + \sigma_u$, where v_u is the voxel uncertainty, and μ_u and σ_u are the per-volume uncertainty mean and standard deviation, respectively. We then calculate the proportions of uncertain correct and incorrect voxel predictions for these models.

For both models, we observe that in cases in which the model made a correct prediction, most predictions tend to be certain (though the margin is substantially wider in the one-material case than the two-material case). This observation is consistent with the observed level of agreement between the thresholded and random walker labels quantified in Table 1. The confidence distributions are shown in Fig. 19a

and Fig. 19b. In cases where the model makes an incorrect prediction, the one-material model tends to be confident in several of these predictions, whereas with the two-material model, incorrect predictions are (correctly) less certain. This may be attributed to the presence of the two phases in the two-material case. While both models tend to predict an overabundance of pore voxels within the real examples, the two-material model manages to predict between the two material phases with relatively high accuracy, and the majority of “wrong” cases above might involve discrepancy between these two phases rather than that between phase and pore.

The uncertainty maps generated by the NN are calculated as the standard deviation over the model’s prediction for each voxel from several inference runs each with a different, random subset of neuron activations dropped out of the calculation. Using the method described in Sect. 3.1.2, we identify a threshold value for which voxels with predictions whose standard deviation is greater than the threshold are deemed to have uncertain predictions. We interpret this to mean that those voxels might represent a pore or a material. By generating three versions of the segmentation where

Fig. 16 Sample cross section of held out CT scan (top left), associated random walk label (top right) and our model's predictions (lower left) and corresponding uncertainty map (lower right) from our full training method



(1) all uncertain voxels are labeled as pores (2) all uncertain voxels are labeled as material and (3) all voxels are labeled according to the mean prediction over all inference runs, we produce bounds as well as a nominal prediction for the porosity of the scanned material. This is a conservative bound of the geometry. In none of the NN predictions are all of the ‘uncertain’ voxels all predicted to belong exclusively to one phase—there is no sampled output that realizes either the upper or lower extreme for the pore geometry. Another potential interpretation of the uncertainty is to take the sampled segmentations as a whole and calculate the variance in porosity measured over the realized, sampled segmentations. We currently choose the conservative approach to ensure the true material porosity is captured within the bounds of our model, but we hypothesize that a tighter bound exists, and research toward that end is left for future work.

The metrics discussed in Sect. 3.1.2 describe the spatial distribution, local size, and topology of the pore space. Segmentation clearly plays a significant role and is reflected in differences between these metrics. The differences arising due to segmentation are particularly apparent in the “high porosity” bound segmentation, suggesting that uncertainty

is skewed such that some voxels that are initially assigned as solid ($\bar{x} < 0.5$) have higher uncertainty as compared to voxels that are initially assigned as pore space. This effect is noticeable in all metrics, but particularly pronounced in the percolation-related metrics; this is not surprising, considering that only a few voxels added or removed to the pore phase can facilitate or break connectivity, resulting in highly nonlinear effects on topology-related metrics such as these. As mentioned previously, the segmentation bounds used here likely represent very conservative (i.e. loose) bounds based on the uncertainty. As such, the differences in the metrics across these different segmentations are likely exaggerated, and the present case represents an overall upper bound on uncertainty.

Notably, our pipeline required less training data than is typical for CT segmentation tasks in the medical domain. Unlike some other domains, materials are characterized by many repetitive patterns, which makes this problem tractable with fewer samples than would normally be expected for successful segmentation. Another advantage in our dataset is the size of the CT volumes. In contrast to examples typical in the medical domain literature [3], our data have 2 orders of

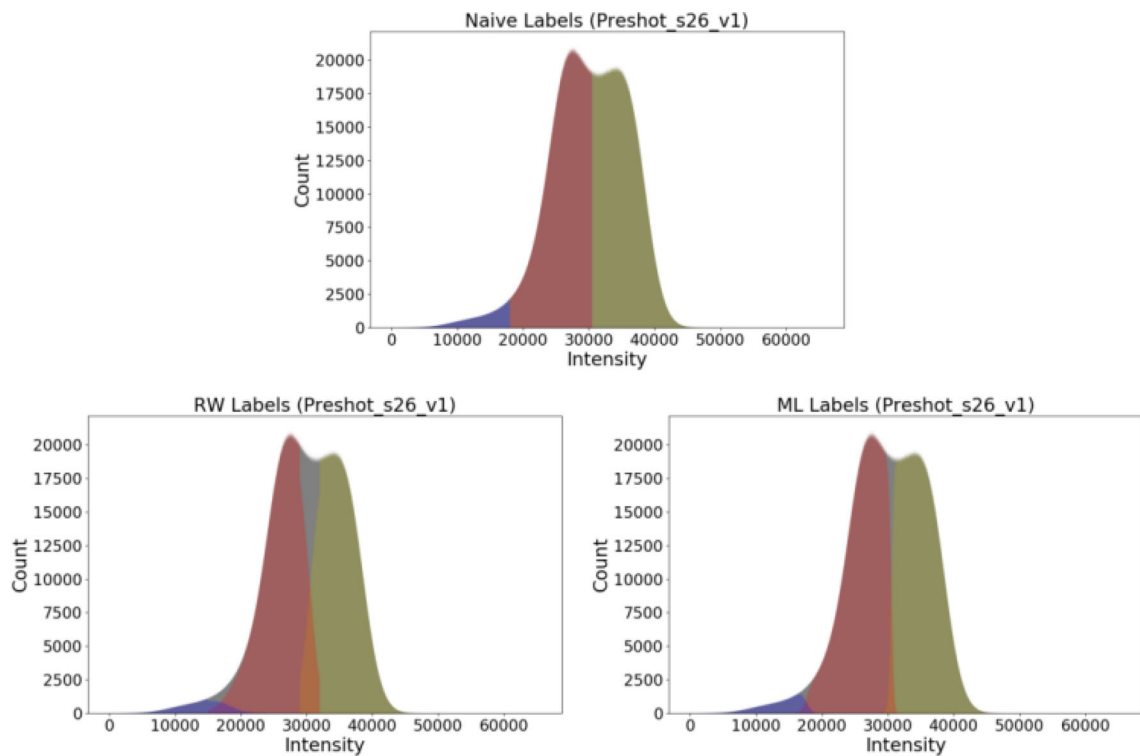
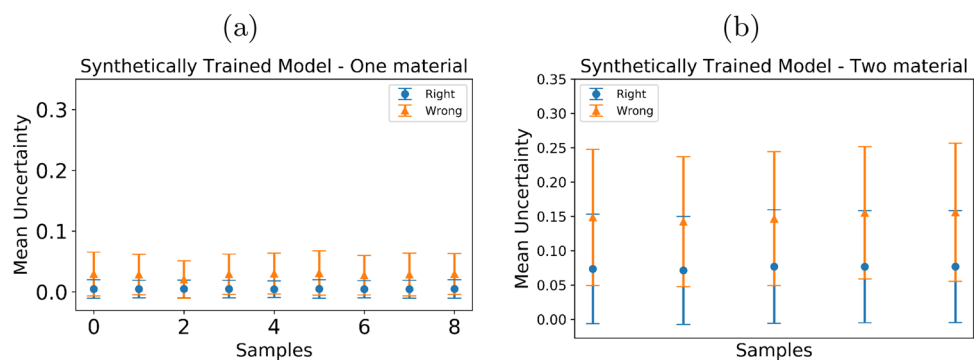


Fig. 17 Grayscale histograms with the number of voxels predicted to belong to each class. Pores are shown in blue, material one is shown in red, and material two is shown in yellow. Threshold labels (top) impose clear cutoffs by voxel intensity, while random walker labels (bottom

left) and CNN labels (bottom right) show overlapping regions of intensity levels for different classes since these algorithms rely on contextual information to assign labels to each voxel in the image. (Color figure online)

Fig. 18 (Left) Average uncertainties across correct and incorrect voxel predictions, using one-material synthetically trained model. (Right) Average uncertainties across correct and incorrect voxel predictions, using two-material synthetically trained model



magnitude more voxels per example, providing significantly more labeled voxels for supervised training.

5.2 Future work

There are several areas related to this work that are ripe for exploration. We found that synthetic training examples improved performance of the deep learning model on the segmentation of real CT scans. While we experimentally determined a set of parameters that generated simulated CT scans that were qualitatively similar to the real target CT domain, there are several potential methods that would serve

to bring the synthetic examples closer to the real data. Deep learning techniques such as domain transfer [41] could be employed to more closely mimic the noise and artifacts associated with scans produced by a particular CT machine, for example. Neural style transfer approaches [42] could also be applied to improve the similarity of the image textures. If we can reliably train the model on only synthetic data, we can appropriately calibrate the model’s uncertainty since we have ground truth by definition with synthetic training data.

While we have conservatively estimated the geometric bounds of the subject materials, a tighter bound could be pursued by gaining a better understanding of the relative

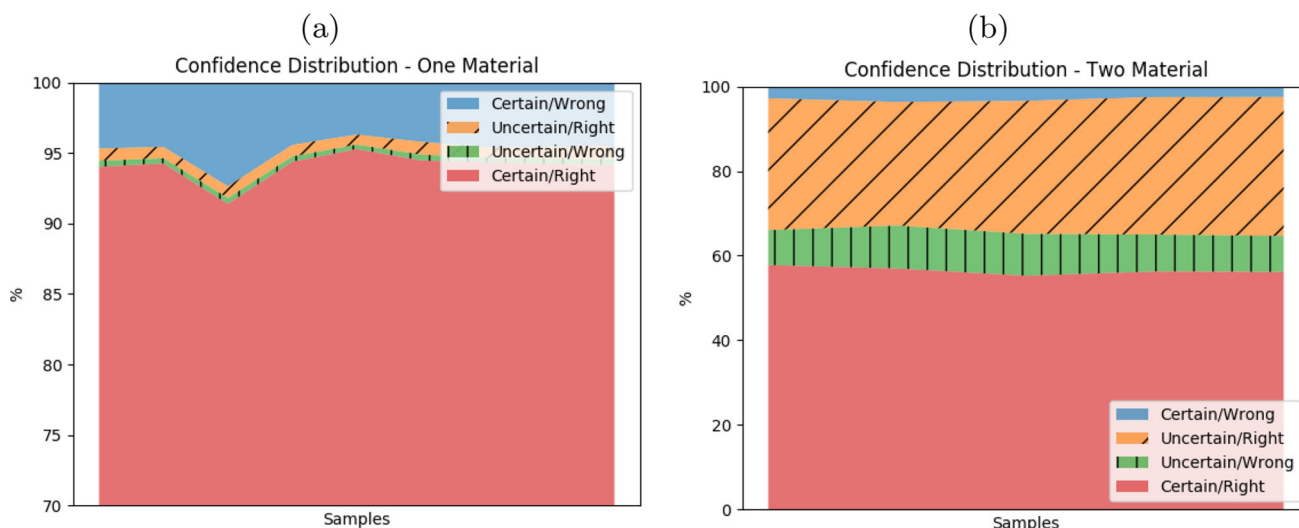


Fig. 19 (Left) Percentages of uncertain correct and incorrect voxel predictions, using one-material synthetic labels to train model zoomed to show small regions of distribution plot. (Right) Percentages of uncertain correct and incorrect voxel predictions, using two-material synthetic labels to train model

importance between where the various predictions for a particular voxel fall on the number line (i.e. close to the decision boundary between classes) and the variance over the predictions for the voxel [43]. Our current approach considers only the variance and a study on the distributions of predicted values could inform a tighter bound. A limitation of this work is that our estimated predictive uncertainty does not distinguish aleatoric from epistemic uncertainty. With multiple algorithmically generated labels, there is additional aleatoric uncertainty to consider in the variance among labels for each training example. The algorithms we used to generate labels do not inherently capture uncertainty, and the development of methods to rigorously quantify this source of uncertainty along with a detailed analysis of the impact of the label variance is left for future work.

Additional uncertainty metrics have been proposed in the literature [44] that can improve the validation of our uncertainty estimation, and potentially the choice of threshold that determines which voxel predictions are designated as ‘uncertain’ in the volumes of interest.

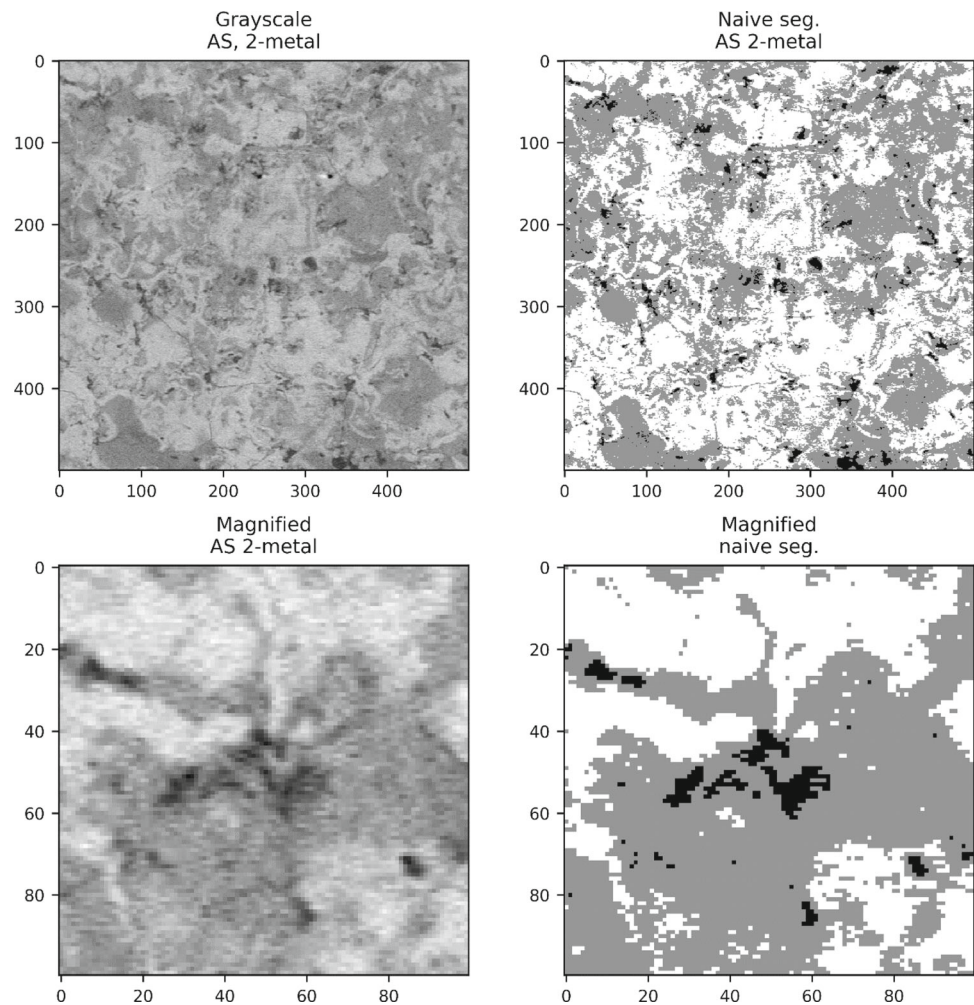
Finally, several approaches in the literature have demonstrated success with segmentation tasks including Transformer-based architectures [21], residual network architectures, [45], and Bayesian NNs [46] as well as optical flow algorithms [47]. Additionally, several novel generative models are emerging such as diffusion-based models [48], with potential to improve segmentation methods. At the time of this study, we limited our scope to use the VNet architecture known to perform well with large material volumes [6], but a broader comparison of results of our pipeline with these

architectures in place of the VNet is an interesting area left for future work.

6 Conclusion

We have developed a method to perform automated, replicable segmentation of CT scans of complex materials while characterizing the uncertainty in geometric predictions. In the absence of expert-generated labels, our workflow produces viable segmentations along with uncertainty maps that can be used downstream to calculate and understand material properties and to bound quantities of interest. Leveraging simulations of target materials as well as synthetic CT software enables pretraining of the NN to recognize features typical of the materials before refining NN weights with examples of real data. Uncertainty maps are generated by performing inference multiple times on the same input example, with stochasticity introduced into the NN with dropout layers that are active during inference. The variance among the inference runs on the same example characterizes the per-voxel uncertainty in the model’s prediction for the classification at each voxel. Bounds on the relative prevalence of each material present in a scanned sample are estimated by shifting all uncertain predictions together to identify the bounds on the amount of a particular material present, enabling estimations of the range of geometry-sensitive quantities in the materials.

Fig. 20 Examples of grayscale image slices and corresponding results of threshold segmentation for 2-metal samples. (Color figure online)



Appendix A: Additional results

In this section, we provide additional examples of our data, labels, and results.

Figure 20 shows examples of thresholding on 2-metal samples.

We examined additional variants as described in Sect. 4. Figure 21 shows output from a model trained only with synthetic one-material data. We also trained a model on real CT scan examples and labels generated via random walk segmentation. We used two scans for training, one scan for validation, and held one scan out for post-training inference and analysis. Figure 22 shows sample output from training a model on random walk labels.

We trained an additional, combined model on real CT scan examples using labels generated via both grayscale intensity

thresholding and random walk segmentation. Our training and validation set consisted of two duplicates of each scan—one with the thresholded label and one with the random walk label. We used two examples for training (four scans total), one example for validation (two scans total), and held one scan out for post-training inference and analysis. Figure 23 shows sample output from this training approach.

Next, as with the one material dataset, we trained a model on synthetic two material CT scans and their respective generated ground truth labels. In this case, we used eleven scans for training, two scans for validation, and held five scans out for post-training inference and analysis. Sample output is shown in Fig. 24.

Fig. 21 Sample cross section of held out CT scan (top left), associated random walk label (top right) and our model's predictions (lower right) and corresponding uncertainty map (lower left) for one material model trained on synthetic data

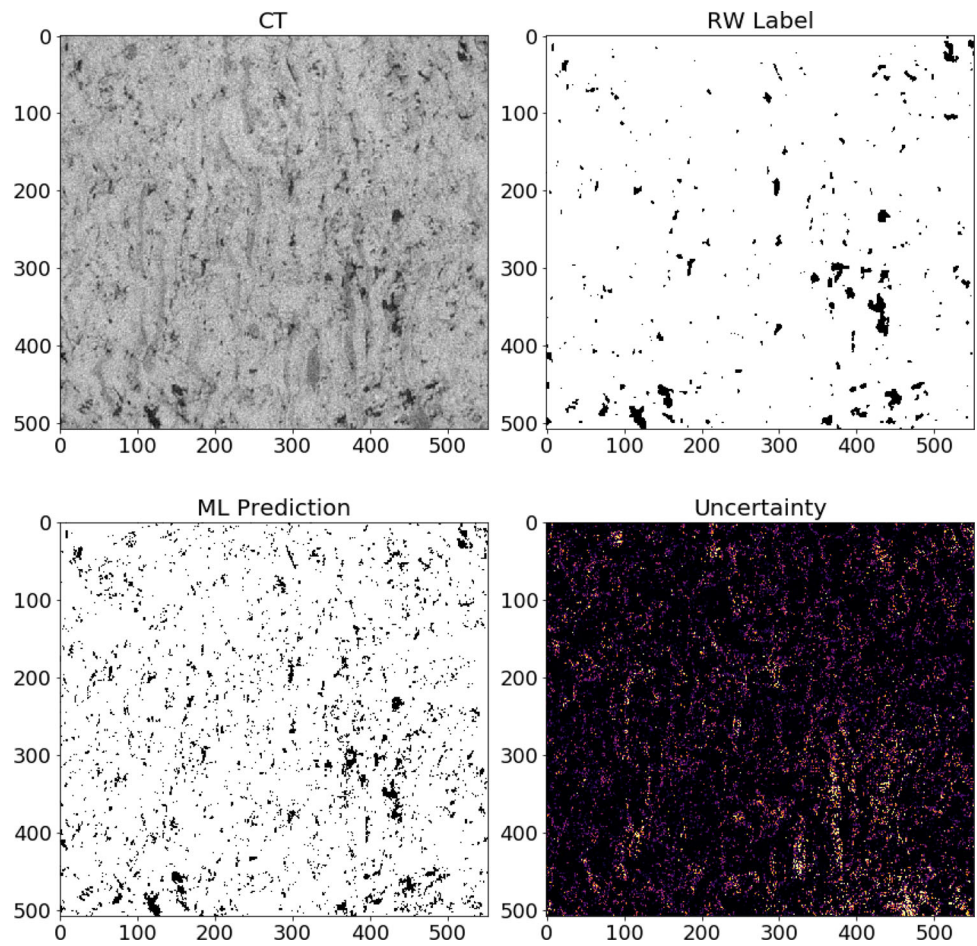


Fig. 22 Sample cross section of held out CT scan (top left), associated random walk label (top right) and our model’s predictions (lower right) and corresponding uncertainty map (lower left) for two material model trained on random walk labels

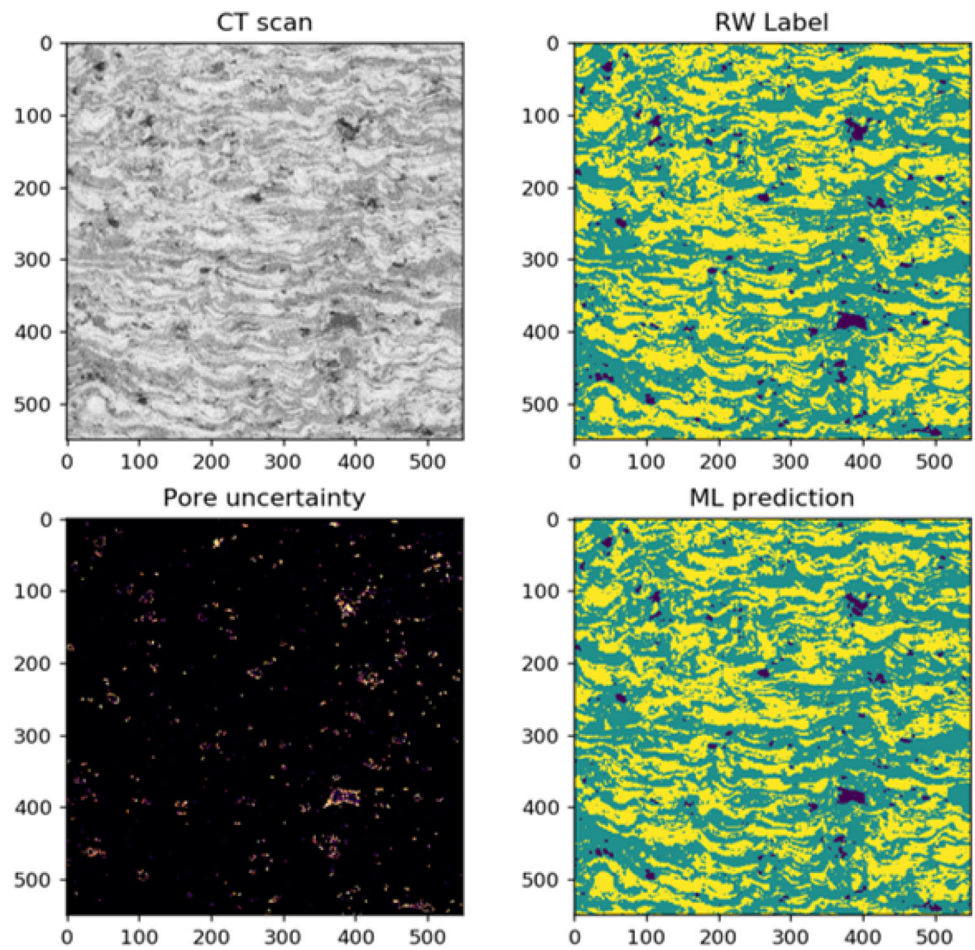


Fig. 23 Sample cross section of held out CT scan (top left), associated random walk label (top right) and our model's predictions (lower right) and corresponding uncertainty map (lower left) for two material model trained on both thresholded and random walk labels

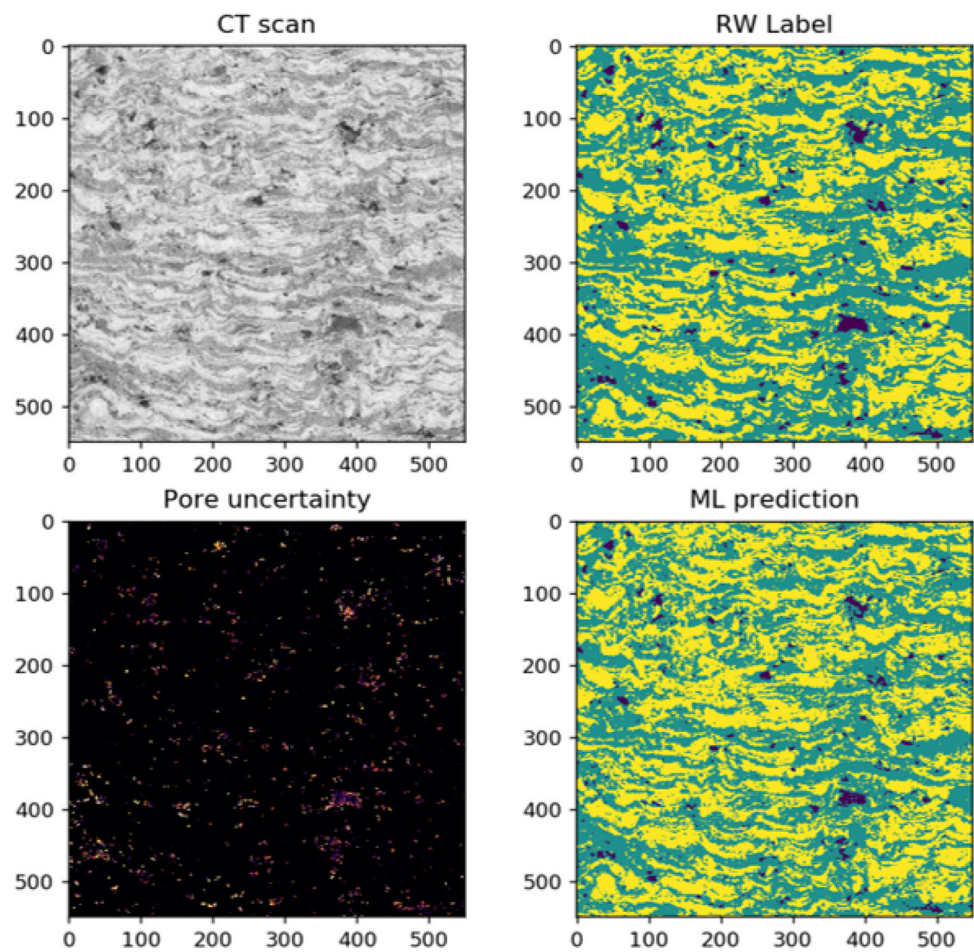
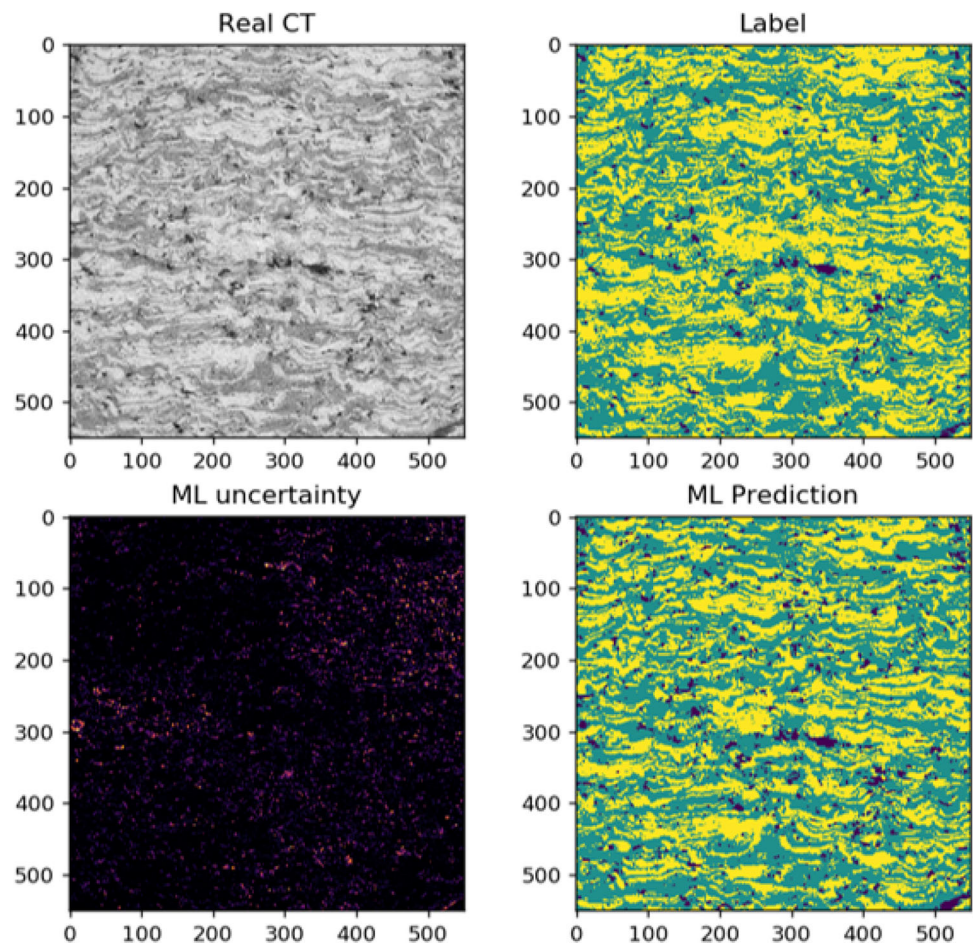


Fig. 24 Sample cross section of real CT scan (top left), associated random walk label of that scan (top right) and our model's predictions and uncertainties for two material model trained on synthetic data



Acknowledgements The authors would like to thank Tyler Ganter for his continual contributions to the codebase and Kat Reinier for supporting the computing infrastructure that enabled this work. We are also grateful to David Stracuzzi and Michael Darling for discussions on uncertainty quantification and potential future directions. Additionally, we would like to thank Don Brown, Björn Clausen, Melvin Borrego, Tim Ickes, Bill Ward, and Andrew Chuang for assistance with CT imaging.

Funding This research used resources of the Advanced Photon Source, a U.S. Department of Energy (DOE) Office of Science User Facility operated for the DOE Office of Science by Argonne National Laboratory under Contract No. DE-AC02-06CH11357. Partial support was provided by the Laboratory Directed Research & Development program at Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. This article has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting

the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan <https://www.energy.gov/downloads/doe-public-access-plan>.

Code availability Monte Carlo Dropout Network (MCDN) 3D CT Segmentation code is available at <https://github.com/sandialabs/mcdn-3d-seg>.

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

References

- Al-Amri SS, Kalyankar NV (2010) Image segmentation by using threshold techniques. CoRR [arXiv:1005.4020](https://arxiv.org/abs/1005.4020)
- Grady L (2006) Random walks for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 28(11):1768–1783. <https://doi.org/10.1109/TPAMI.2006.233>
- Milletari F, Navab N, Ahmadi S-A (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). IEEE, pp 565–571
- Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241
- Hesamian MH, Jia W, He X, Kennedy PJ (2019) Deep learning techniques for medical image segmentation: achievements and challenges. *J Digit Imaging* 32:582–596
- Krygier MC, LaBonte T, Martinez C, Norris C, Sharma K, Collins LN, Mukherjee PP, Roberts SA (2021) Quantifying the unknown impact of segmentation uncertainty on image-based simulations. *Nat Commun* 12(1):1–11
- Jones D, Snider C, Nassehi A, Yon J, Hicks B (2020) Characterising the digital twin: a systematic literature review. *CIRP J Manuf Sci Technol* 29:36–52
- Smith R, Fast R (1994) The future of thermal spray technology. *Weld J (Miami) (United States)* 73(7):43–50
- Hassanein AM (1984) Thermal effects and erosion rates from x-ray energy deposition in ICF reactor first walls. *J Nucl Mater* 123(1–3):1459–1465
- Rodgers TM, Mitchell JA, Olson A, Bolinteanu DS, Vackel A, Moore NW (2021) Fast three-dimensional rules-based simulation of thermal-sprayed microstructures. *Comput Mater Sci* 194:110437
- Rodgers TM, Bolinteanu DS, Moore N (2020) Calibration of thermal spray microstructure simulations to experimental data using Bayesian optimization. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States)
- Moore NW, Bell KS, Hilborn H, Woodworth BN, Mesh M, Bruss DE, Franke BC, Poole PL, Hohlfelder RJ, Zarick T et al (2021) Sample test array and recovery (star) platform at the national ignition facility. *Rev Sci Instrum* 92(5):053539
- Battaile C, Owen S, Moore N (2017) Direct numerical simulations of microstructure effects during high-rate loading of additively manufactured metals. In: APS shock compression of condensed matter meeting abstracts, pp 7–001
- Bassiouny A, El-Saban M (2014) Semantic segmentation as image representation for scene recognition. In: 2014 IEEE international conference on image processing (ICIP). IEEE, pp 981–985
- Mishra A, Aloimonos Y, Fermuller C (2009) Active segmentation for robotics. In: 2009 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 3133–3139
- Coleman GB, Andrews HC (1979) Image segmentation by clustering. *Proc IEEE* 67(5):773–785
- Preetha MMSJ, Suresh LP, Bosco MJ (2012) Image segmentation using seeded region growing. In: 2012 international conference on computing, electronics and electrical technologies (ICCEET). IEEE, pp 576–583
- Gao J, Wang B, Wang Z, Wang Y, Kong F (2020) A wavelet transform-based image segmentation method. *Optik* 208:164123
- Zou M, Wang D (2001) Texture identification and image segmentation via Fourier transform. In: Image extraction, segmentation, and recognition, vol 4550. SPIE, pp 34–39
- Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
- Lahoud J, Cao J, Khan FS, Cholakkal H, Anwer RM, Khan S, Yang M-H (2022) 3d vision with transformers: a survey. *arXiv preprint arXiv:2208.04309*
- Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, Fieguth P, Cao X, Khosravi A, Acharya UR et al (2021) A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Inf Fusion* 76:243–297
- Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv Neural Info Process Syst* 30
- Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural networks. In: International Conference on Machine Learning. PMLR, pp 1613–1622
- Gal Y, Ghahramani Z (2016) Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International conference on machine learning, pp 1050–1059
- Hu S, Worrall D, Knekt S, Veeling B, Huisman H, Welling M (2019) Supervised uncertainty quantification for segmentation with multiple annotations. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd international conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22. Springer, pp 137–145
- Van Aarle W, Palenstijn WJ, De Beenhouwer J, Altantzis T, Bals S, Batenburg KJ, Sijbers J (2015) The ASTRA toolbox: a platform for advanced algorithm development in electron tomography. *Ultramicroscopy* 157:35–47
- Van Der Walt S, Colbert SC, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng* 13(2):22–30
- Bolinteanu DS, Olson AJ, Collis HR, Pokharel R, Brown DW, Chuang AC, Vackel A, Ivanoff T, Madison JD, Moore NW Quantifying pore morphology in thermal sprayed tantalum using x-ray micro-computed tomography. Under review
- van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T (2014) The scikit-image contributors: scikit-image: image processing in Python. *PeerJ* 2:453. <https://doi.org/10.7717/peerj.453>
- Chollet F et al (2015) Keras. <https://keras.io>
- Dillon JV, Langmore I, Tran D, Brevdo E, Vasudevan S, Moore D, Patton B, Alemi A, Hoffman M, Saurous RA (2017) TensorFlow distributions. *arXiv preprint arXiv:1711.10604*
- Zeiler MD (2012) ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*
- Monte Carlo Dropout Network (MCDN) 3D CT Segmentation. <https://github.com/sandialabs/mcdn-3d-seg>
- Torquato S, Haslach H Jr (2002) Random heterogeneous materials: microstructure and macroscopic properties. *Appl Mech Rev* 55(4):62–63
- Jiao Y, Stillinger F, Torquato S (2007) Modeling heterogeneous materials via two-point correlation functions: basic principles. *Phys Rev E* 76(3):031110
- Münch B, Holzer L (2008) Contradicting geometrical concepts in pore size analysis attained with electron microscopy and mercury intrusion. *J Am Ceram Soc* 91(12):4059–4067
- Hilfer R (1992) Local-porosity theory for flow in porous media. *Phys Rev B* 45(13):7115
- Biswal B, Manwart C, Hilfer R (1998) Three-dimensional local porosity analysis of porous media. *Physica A* 255(3–4):221–241
- Samet H, Tamminen M (1988) Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Trans Pattern Anal Mach Intell* 10(4):579–586
- Hoffman J, Tzeng E, Park T, Zhu J-Y, Isola P, Saenko K, Efros AA, Darrell T (2017) CyCADA: cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*

42. Gatys LA, Ecker AS, Bethge M, Hertzmann A, Shechtman E (2017) Controlling perceptual factors in neural style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3985–3993
43. Darling MC (2019) Using uncertainty to interpret supervised machine learning predictions. https://digitalrepository.unm.edu/ece_etds/485/
44. Kolodny MA, Wiegmann DM, Pham T (2010) Ground/air multi-sensor interoperability, integration, and networking for persistent ISR IX. SPIE
45. Chen H, Dou Q, Yu L, Heng P-A (2016) VoxResNet: deep voxel-wise residual networks for volumetric brain segmentation. arXiv preprint [arXiv:1608.05895](https://arxiv.org/abs/1608.05895)
46. LaBonte T, Martinez C, Roberts SA (2019) We know where we don't know: 3d Bayesian CNNs for credible geometric uncertainty. arXiv preprint [arXiv:1910.10793](https://arxiv.org/abs/1910.10793)
47. Huang T-C, Zhang G, Guerrero T, Starkschall G, Lin K-P, Forster K (2006) Semi-automated CT segmentation using optic flow and Fourier interpolation techniques. *Comput Methods Progr Biomed* 84(2–3):124–134
48. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. *Adv Neural Inf Process Syst* 33:6840–6851

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