



# Deep learning approaches to semantic segmentation of fatigue cracking within cyclically loaded nickel superalloy

David B. Menasche<sup>a,\*</sup>, Paul A. Shade<sup>b</sup>, S. Safriet<sup>b</sup>, Peter Kenesei<sup>c</sup>, Jun-Sang Park<sup>c</sup>, William D. Musinski<sup>b</sup>

<sup>a</sup> Hamiltonian Group LLC, Washington DC 20007, United States

<sup>b</sup> Air Force Research Laboratory, Materials and Manufacturing Directorate, Wright-Patterson AFB, OH 45433, United States

<sup>c</sup> Advanced Photon Source, Argonne National Laboratory, Argonne, IL 60439, United States



## ARTICLE INFO

### Keywords:

Synchrotron  
Microcomputed tomography  
Deep learning  
Artificial intelligence  
Image segmentation  
Computer vision  
Fatigue crack  
Superalloy

## ABSTRACT

Improvements to synchrotron-based micro-computed tomography scanning capabilities have gifted researchers the ability to characterize 4D material thermomechanical responses more thoroughly than ever before. These advancements, however, have brought about new challenges in analyzing the resulting deluge of data. We report on a nickel-based superalloy specimen imaged 26 times in-situ during cyclic loading at Argonne National Laboratory Advanced Photon Source beamline 1ID, in order to monitor crack growth within the microstructure. Several deep learning approaches which utilize convolutional neural networks are implemented to segment crack features from reconstructed tomography scans. U-Net architecture implementations are found to be especially effective, achieving  $IoU = 0.995 \pm 0.004$  and Matthews correlation coefficient scores of  $\phi = 0.826 \pm 0.085$ . These advancements broaden possibilities for scientists seeking to automate segmentation analyses of similar large datasets.

## 1. Introduction

Recent advances in synchrotron-based non-destructive characterization methods have made possible novel four-dimensional experiments that interrogate macroscopic material volumes during microstructural and/or micromechanical state evolution [1]. By imaging the same sample multiple times during a thermo-mechanical loading process, snapshots of the 3D structure and its evolution are obtained [2]. These snapshots provide materials scientists and engineers with data used to instantiate material models and evaluate their efficacy in predicting material performance and properties.

Micro-computed tomography ( $\mu$ CT) [3] is a method used to probe the material density of specimens, including the location, sizes, and distribution of any internal defects or precipitates, secondary phases, or porosity [4]. When  $\mu$ CT is performed at synchrotron x-ray sources, scan times for macroscopic volumes are rapid. This facilitates both the collection of numerous volumetric measurements of a specimen during the course of an experiment [5] and the imaging of time-resolved phenomena [6]. Harnessing these advancements has led to significant increases in measurement throughput and corresponding increases in

reconstructed data volume [7].

These  $\mu$ CT experiments produce 2D cross-sectional images, which correspond to the specimen's 3D material density when stacked. Performing image analysis operations on each cross-section is a straightforward way to extract volumetric information. These operations may require multiple iterations of artifact removal, filtration, denoising, segmentation, or similar procedures before desired data are extracted. These operations are often implemented using convolutional filters, the effects of which depend on the parameters of the convolutional kernel. Researchers typically optimize these parameters based on the problem context, but full examination of the parameter space can be intractable given complex processing pipelines. More generally, the ideal rules for processing and segmenting images are often complex, multi-conditional, and context-based, which makes explicit logical programming intractable to implement in a general work-flow.

Artificial intelligence and machine learning have emerged as advantageous frameworks for computing in exactly these conditions, a rise aided by accessible GPU programming tools for model building and deployment. By leveraging large data pools, these applications utilize optimization procedures to develop empirical models which encode

\* Corresponding author.

E-mail address: [david.menasche@hamiltoniangroup.com](mailto:david.menasche@hamiltoniangroup.com) (D.B. Menasche).

complex rules implicitly [8]. Applying these approaches to semantic segmentation (pixel-by-pixel-classification) tasks has been of particular interest to those developing computer vision applications for classification and image analysis [9,10], including within the medical imaging community for lesion detection [11–13]. Other applications automate the detection of morphological features of interest like protein fibril crossovers in electron microscopy images [14].

Deep learning approaches [15], specifically those that utilize convolutional neural networks (CNNs) or fully convolutional networks (FCNs) [16], have demonstrated best-in-class efficacy in semantic segmentation (pixel-by-pixel-classification) [17] and image classification [18,19] tasks. Engineered initially for segmentation of volumetric biomedical information, the U-Net [17] introduced a U-shaped encoder-decoder convolutional architecture capable of identifying collections of low-level features of an image and tracking them spatially. Subsequently, these advances been applied to computer vision applications like automating the segmentation of cracking in concrete [20,21], an important indicator for structural health monitoring. The framework has also been extended to achieve semantic segmentation of 3D volumes with the 3D U-Net [22] and the V-Net [23] architectures.

Networks such as these that substitute convolutional layers in lieu of densely connected hidden layers for feature extraction and processing are called fully convolutional networks (FCNs). Adapted by Long et al. from the AlexNet [18], VGG [24], and GoogLeNet [25] architectures, FCNs utilize the translational invariance of the convolution operation to identify features of interest anywhere on an image [16]. Trained end-to-end, these models predict image phases on a pixel-by-pixel basis, obviating the need for bounding box detection or similar iterative approaches. These algorithms and similar derivatives have been implemented by groups using  $\mu$ CT to study the morphology of complex multiphase Al-Si alloys [26,27] and to classify phases within steel microstructure [28].

Here, we apply a range of deep learning models featuring FCNs and their variations to the semantic segmentation of images depicting fatigue cracks within a cyclically loaded microstructure. Specifically, we compare 2D image segmentations generated from

1. the U-Net;
2. a fully convolutional (FC) U-Net model which probes the choice of pooling scheme used by the original U-Net architecture;
3. the FCN-8s network, a FCN developed by Long et al. from the VGG16 model;
4. a ‘FCN-4s’ model, an extension of the FCN-8s which bridges the U-Net and the FCN-8s.
5. A three-channel U-Net model is also considered as an alternative to the wholly 3D models of [22,23], to provide limited 3D context while also accommodating the unbalanced aspect ratio of the region of interest.

## 2. Material and methods

### 2.1. Experimental data collection and reconstruction

Preparation of the experimental specimen took place at the Air Force Research Laboratory campus in Wright Patterson-AFB, Ohio.

The material used was obtained from a sub-scale low solvus high refractory (LSHR) nickel-base superalloy billet that was isothermally forged and subsequently exposed to supersolvus and aging heat treatments. The resulting microstructure had an average grain diameter of  $\sim 85 \mu\text{m}$ . The specimen was extracted from the sub-scale billet forging via electrical discharge machining (EDM) in the shape of an hourglass specimen containing a 1.5 mm long uniform gauge region with a cross-sectional area of  $1 \text{ mm} \times 1 \text{ mm}$ . No subsequent polishing was applied to the surface. More details of the processing and characterization of the LSHR microstructure used in this study can be found in Musinski et al [29].

The chemical composition of the LSHR used in this study is shown in Table 1.

To promote crack localization, a notch with nominal dimensions of  $80 \mu\text{m}$  wide by  $40 \mu\text{m}$  deep by  $8 \mu\text{m}$  high was milled by focused-ion-beam (FIB) into the center of the specimen. A pure-gold fiducial marker [32] was affixed to the sample approximately  $100 \mu\text{m}$  above this notch to aid in sample alignment and dataset registration. As this marker is much denser than the LSHR specimen, the marker’s x-ray absorption signature is easily identifiable in detector radiographs (as in Fig. 1a) and the reconstructed  $\mu$ CT maps. In Fig. 1a, the dark cuboidal region superimposed over the rectangular specimen highlights the x-ray absorption signature of the gold marker relative to the rectangular nickel sample.

The experiment was performed at the 1-ID beamline of Argonne National Laboratory’s Advanced Photon Source (ANL-APS). The RAMS1 load frame insert previously reported in [33] was used to hold, load, and rotate the sample. The device supports both tensile and compressive loading, as well as full 360-degree range of rotation and compatibility with *in-situ* x-ray measurement modalities. During the experiment,  $\mu$ CT scans were repeatedly performed in between periods of mechanical testing to observe the evolution of a growing fatigue crack; material crystallographic and strain information were also queried through concurrent application of near- and far-field High Energy Diffraction Microscopy techniques [34,35]. Analysis of these additional data are not within the scope of this paper but will be the topic of future work.

To perform  $\mu$ CT, x-ray projections are collected on an area detector while the specimen is rotated on a fixed axis. The material density is computationally recovered from the set of projections, yielding a ‘stack’ of cross-sections with each pixel element corresponding to a relative measure of material density for that pixel. The  $\sim 2 \text{ k} \times 2 \text{ k}$  CCD area detectors available at synchrotron sources facilitate simultaneous collection of  $\sim 1 \text{ mm}^3$  volumes at effective pixel pitches between  $0.5$  and  $1.5 \mu\text{m}$ .

In this case, a parallel-beam  $\mu$ CT geometry was used, in which a  $71.676 \text{ keV}$  monochromatic x-ray beam illuminated a subvolume of the sample while it was rotated about its loading axis. The beam energy was calibrated by optimizing x-ray absorption through a Rhenium foil at the k-shell edge. The notched region was aligned to the incident beam before each scan. Radiographs were captured every  $0.2$  degrees over a 360-degree interval on an optical LuAG scintillator coupled to a  $(2048 \text{ pixel})^2$  Retiga CCD detector placed  $70 \text{ mm}$  from the sample rotation axis. The effective pixel pitch of the CCD-scintillator system is  $1.48 \mu\text{m}$ , yielding a reconstructed isotropic spatial resolution of  $\sim 3 \mu\text{m}$ .

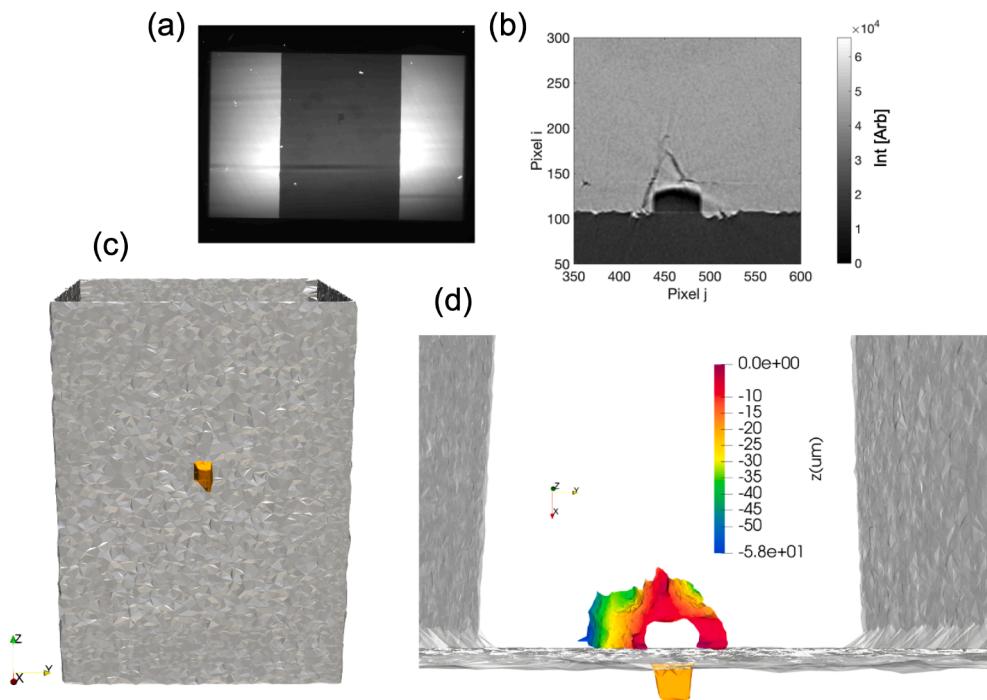
The sample was imaged a total of 26 times: once before any load was applied, once after loading the sample to  $1000 \text{ N}$ , and then 24 times throughout the cyclic process according to the schedule shown in Fig. 2b. The cyclic loading was conducted in uniaxial tension at a rate of  $2 \text{ Hz}$ , where each loading cycle consisted of a minimum load of  $50 \text{ N}$  and a maximum load of  $1000 \text{ N}$ ; at the end of a loading segment the load was reduced from the maximum and held in force control at  $750 \text{ N}$  during x-ray data collection to prevent material creep.

Raw  $\mu$ CT data were reduced at ANL-APS on in-house computing resources immediately after each  $\mu$ CT scan using the GridRec algorithm [36–38]. This procedure produced a stack of approximately 900 cross-sectional image files (unsigned 16-bit \*.tiff), for each of the 26 scans, which report the computed relative material density of the specimen within each pixel. The normal of each cross-sectional image points parallel to the specimen loading direction, the z-direction in the schematics in Fig. 1c and 1d. An enlarged region belonging to one “slice” is reproduced in Fig. 1b. Pictured is a cross-section of material that intersects the stress-localization notch. Fatigue cracks are detectable in the reconstructed  $\mu$ CT density field as regions of comparatively lower material density, which is the signature we desire to segment using the tools discussed above. Fig. 1d illustrates the ground truth annotation of the crack after 9500 cycles, localized within the specimen in relation to both the FIB notch and the gold fiducial cube.

**Table 1**

Chemical composition (weight-percent) of the LSHR nickel-base superalloy material [30,31].

Material	Co	Cr	Al	Ti	Mo	W	Nb	Ta	C	B	Zr	Ni
LSHR	20.4	12.3	3.5	3.5	2.7	4.3	1.5	1.5	0.045	0.027	0.05	balance



## 2.2. Training and validation dataset generation

### 2.2.1. Raw image processing

Following collection and reconstruction of the  $\mu$ CT maps, the maps representing the sample after 9500 cycles were chosen for initial model training, as the crack features are most developed in this state. The final crack surface length,  $2c$ , at this instance is approximately 250  $\mu\text{m}$ , and the crack depth,  $a$ , is found to be  $\sim 135 \mu\text{m}$ . In order to annotate the dataset for training, a GUI-based program was written in MATLAB which utilized Laplacian-of-Gaussian (LoG) edge detection to identify candidate data points within each image. This edge detection generated an ansatz segmentation which was overlaid over each material density map, and then individual pixels were manually annotated to be “crack-associated” or not.

A 128x128 pixel region in each image—corresponding to an approximately 190x190 micron<sup>2</sup> region in sample space—was selected as the region of interest (ROI). The size of the ROI is limited by the class imbalance between crack-associated and non-crack associated pixels, discussed below.

The ROI was approximately centered on the crack feature and only the 46 cross-sectional slices containing crack-associated points in the ROI were retained. The ROI within each slice was scaled to unsigned 8-bit integer precision and written to png image format. The choice to downsample image precision was motivated by practicality in light of the extensive use of remote computer resources. Given that the comparatively small ( $N = 46$ ) number of annotated images is too few to fully train a given model, we make heavy use of data augmentation as in [17], whereby affine distortions are applied to each image to generate additional training data from the existing annotated images. These procedures are described in detail in Section 2.4.1.

**Fig. 1.** This figure illustrates the data collected. Micro-computed tomography was performed on the specimen by illuminating it with the x-ray beam and collecting a series of radiographs while it was rotated about the  $z$ -axis. These projections are computationally recovered to generate a series of cross-sections representing the material density, as in (b). This cross-section including this region illustrates the FIB notch milled to induce stress localization and subsequent material cracking, shown as connected low intensity regions within the bulk in (b). A representation of the sample surface, with an attached gold fiducial marker is shown in (c), created by processing raw data collected from images like (b). The gold fiducial is visible through x-ray contrast in (a). (d) locates the crack within a cutaway view down the loading axis.

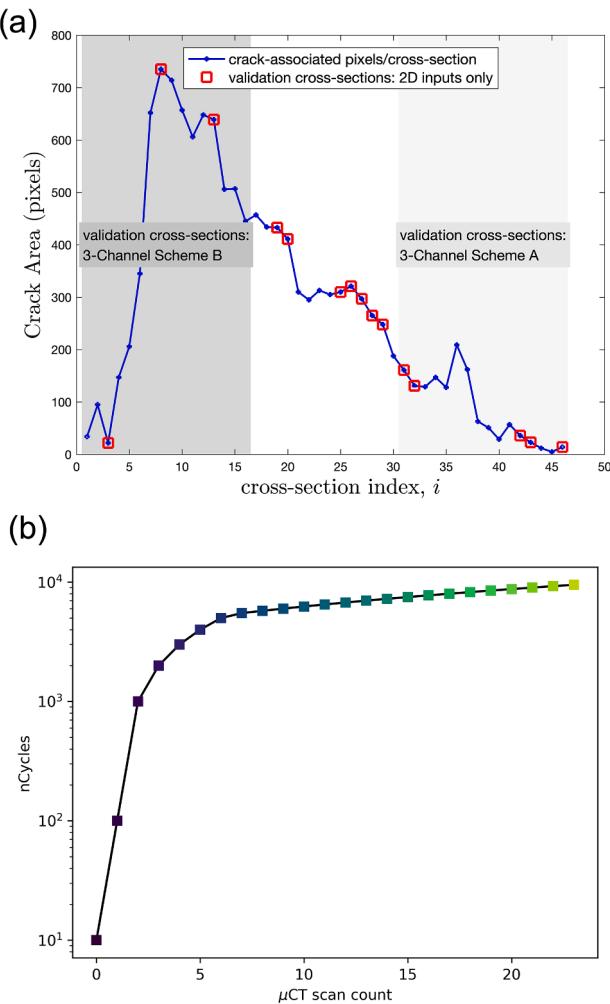
### 2.2.2. Image partitioning

The strategy for partitioning of the images between training and validation data sets was different for models using 2D-inputs versus those models using three-channel input modes. For the 2D-input models described below, we randomly select 2/3 of the images to act as seed images for the augmented data model training. The slices corresponding to these indices are highlighted by red boxes in Fig. 2a.

During each training iteration, one of these images is selected and a random affine distortion is applied; the resulting distorted image and its corresponding distorted annotation are used to train the model. These training images are generated on-the-fly at runtime. The remaining 1/3 of the original image set forms the validation set. Fig. 2a also depicts the number of crack associated pixels as a function of the cross-sectional index (slice) of the tomographic scan. Notably, most of the crack’s surface area is imaged by slices with approximate indices  $i \in [7, 30]$ .

Partitioning of the three-channel input training and validation sets was obtained by first separating the slices into two contiguous groups. The group containing the first 2/3 of the 46 reconstructed slices with indices  $i_t \in \{1, 2, \dots, 30\}$  is initially designated the training set, under a training scheme we name “A.” The data block containing the remainder of the slices is designated the validation set, corresponding to images with indices  $i_v \in \{31, 32, \dots, 46\}$ . These validation images are marked in the lightly shaded region (Fig. 2a). To measure effectiveness of the 2D-input U-Net vis-à-vis the three-channel input U-Net, we also train and validate the 2D-input U-Net model under training scheme A.

Subsequent cross-validation of the three-channel U-Net model uses a distinct, ‘flipped’ training scheme which we name “B.” Under this scheme, the first contiguous 1/3 of the cross-sectional images,  $i_t \in \{1, 2, \dots, 15\}$ , is designated the validation set, while the second contiguous 2/3 of the images,  $i_v \in \{16, 17, \dots, 46\}$ , serve as seeds for the augmented training set.



**Fig. 2.** In (a), the number of crack-associated pixels per cross section after 9500 fatigue cycles, as a function of slice cross-section index. Superimposed are markers and color-blocks which illustrate the validation datasets for each network. Cross-sections used to validate the 2D-input networks are indicated with red boxes; images forming validation sets A and B are marked by inclusion within the light and dark shaded regions, respectively. In (b), the  $\mu$ CT measurement schedule as a function of the number of fatigue cycles. Colored markers correspond to the color coding in Fig. 7 and indicate a  $\mu$ CT measurement was performed after that number of cycles.

Data augmentation under each three-channel model training scheme is performed by randomly choosing a slice from the training set to be the ‘center’ slice, then constructing a 128x128x3 tensor (image stack) from that slice and those adjacent. Affine distortions are applied jointly to the stack, which subsequently becomes the model input. The image mask corresponds to the transformed annotations belonging to the ‘center’ slice.

#### 2.2.3. Class imbalance

Within this dataset, a class imbalance [39] exists between crack-associated and non-crack-associated classes. In total, there are 12,902 crack-associated pixels out of 753,664 total pixels in the ROI, a relative abundance of 1.7%. Physically, this can be interpreted to mean the crack interface is sparse in each image. This class imbalance limits the potential size of the training ROI in this case because the crack is sparse and localized; increasing the size of the ROI will have the overall effect of decreasing this relative abundance of crack associated points (given fixed crack area), thereby increasing the probability that the model training finds the false cost minimum associated with predicting zero crack points.

To emphasize consideration of these sparse crack-associated pixels by the model, we consider the effect of employing a weight  $w_i$  to the  $i^{\text{th}}$  pixel within the binary cross-entropy loss,  $L$ , given by

$$L = \sum_{ij} w_i \lambda_{ij} \log p_j$$

where  $\lambda_{ij} = 1$  if pixel  $i$  is a member of class  $j$ , and 0 otherwise, and  $p_j$  is the predicted probability to be associated with class  $j$ . We test the effect of this rescaling during training of the 2D-input U-Net network in particular. Weights of  $w_1 = 29$  (crack-associated pixels) and  $w_0 = 0.51$  (other pixels) are assigned during training under scheme  $w_i$ . These weights were computed utilizing scikit-learn’s implementation of the balancing heuristic developed in [40], where weight  $w_i$  is given by the expression  $w_i = \frac{N}{2n_i}$ , where  $N$  is the total number of pixels in the volume and  $n_i$  is the number of ground truth observations belonging to class  $i$ , such that  $n_0 + n_1 = N$ .

Weights are all unity (an unweighted condition) under training schemes A and B and unless otherwise noted.

#### 2.3. Model descriptions

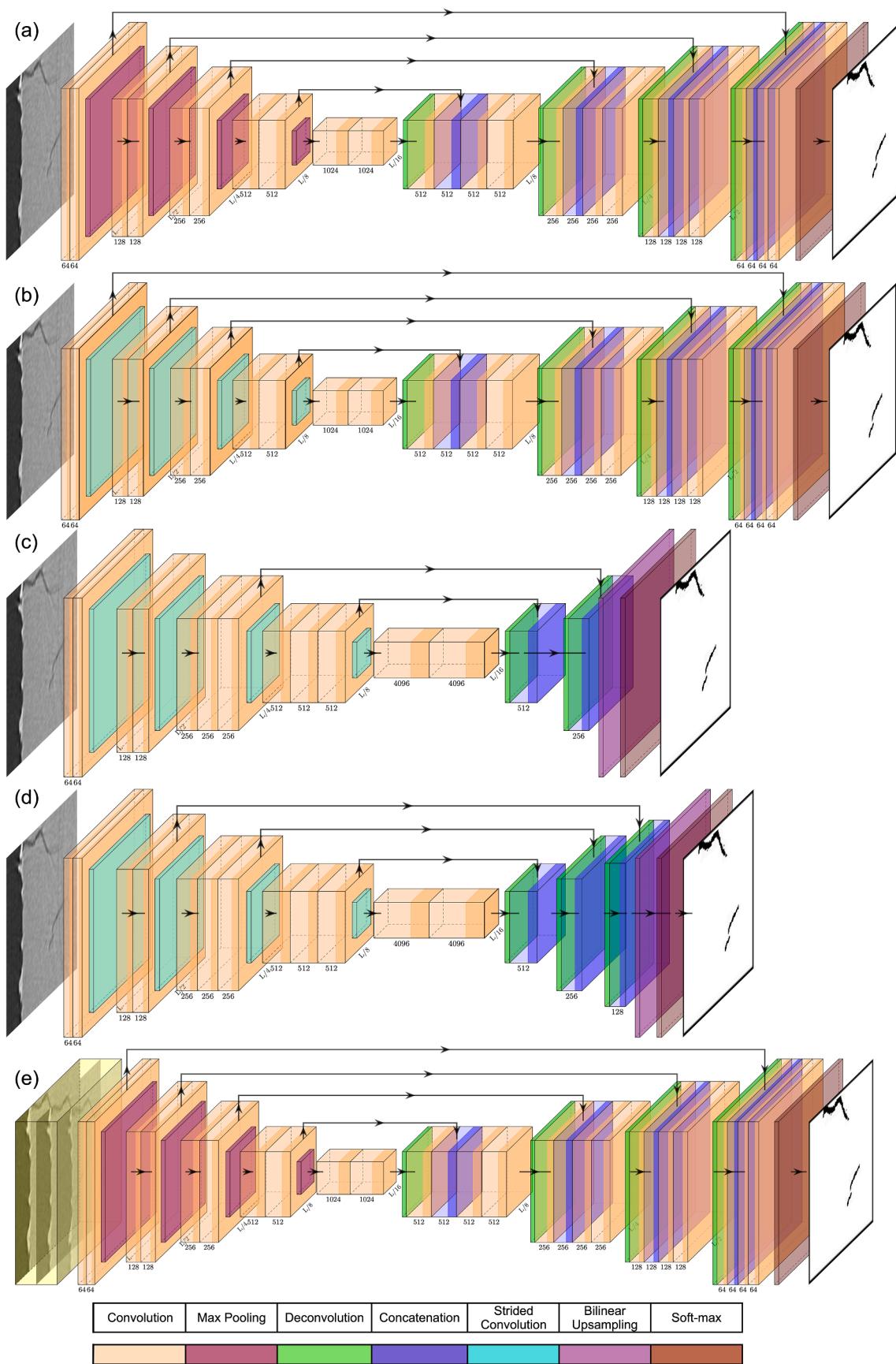
We select five models to compare for evaluation, each depicted schematically in Fig. 3:

- U-Net, (Fig. 3a), the architecture developed in [17] with no changes;
- FC U-Net, (Fig. 3b): a fully convolutional U-Net which substitutes strided convolutions for the pooling operations of the U-Net;
- FCN-8s, (Fig. 3c): a FCN with only two skip connections, but a comparatively longer encoding branch;
- FCN-4s (Fig. 3d): a FCN with three skip connections, and a deep encoding branch; and
- Three-channel U-Net, (Fig. 3e): a generalization of the U-Net that accepts small stacks of three contiguous cross-sections in order to provide limited 3D context to the model.

The U-Net (Fig. 3a) is arranged to utilize successive series of unstrided convolutional layers. In each series or “block,” several convolutional layers increase the number of feature channels of the image, after which a max-pooling operation is applied, thereby reducing the spatial resolution of the image. Successive blocks repeat this operation, until the network reaches the decoding branch which reverses the process through upsampling de-convolution. After each decoding series, model layer outputs are combined (via skip-connections) with feature channel output from the properly sized encoding layers, producing spatially located activations. A final 1x1 convolution rectifies these activations with a sigmoid nonlinearity interpreted as a probability that a given pixel belongs to a class of interest. Choosing and applying a decision boundary to this final layer produces pixel-by-pixel segmentation. For more detailed explanations of the U-Net architecture, we refer readers to Ronneberger *et al* [17].

We also probe the effect of substituting strided convolutions for the max pooling layers performing downsampling within the U-Net, as suggested by [41]. This modification increases the generality of the model, by allowing it to learn the upsampling kernel, at the cost of modestly increasing the number of model parameters. These changes are reflected in Fig. 3b.

The FCN-(4,8,16,32)s series of architectures is adapted from a VGG16 network, where spatially coarse convolutional layers with large numbers of filter channels are substituted for densely connected hidden layers [16]. The model naming convention indicates the upsampling required for the model prediction to match the resolution of the initial input image, e.g. the penultimate activation of the FCN-32s must be upsampled by a factor of 32 to agree with the input dimensions. This upsampling is achieved by a strided convolutional layer as discussed above. The FCN-16s appends to the FCN-32s by adding a 2x upsampling



(caption on next page)

**Fig. 3.** Block schematics of the network architectures evaluated. (a) **The U-Net** utilizes successive series of encoding blocks which embed spatially organized signals into convolutional feature channels. Higher level structures are detected and combined with copies of spatially resolved activations to produce spatially resolved segmentations. (b) **The FC U-Net** substitutes strided convolutions in place of Max Pooling operations. (c) **The FCN-8s** has a comparatively longer encoding structure than the U-Net, with an additional convolutional layer in encoding blocks three and four, but only two skip connections. The spatially narrowest layer also has four times as many feature channels as compared to the U-Net constructs. Also notable, there are no convolutional layers to mix and process activations following the deconvolutional layers. (d) **The FCN-4s** adds another block of deconvolution and concatenation to the FCN-8s network and includes a total of 3 skip connections before a final 4x upsampling filter. **The Three-Channel U-Net** has the same overall architecture as the U-Net, but takes a three-channel tensor as input, highlighted in yellow, with each of the three feature channels coded to one of three adjacent cross-sections of density field.

deconvolutional layer together with a ‘skip connection,’ an appropriately sized activation from earlier in the model pipeline, before a final 16x upsampling layer. Long et al. note improved prediction on the PASCAL Visual Object Classes (VOC) Challenge [9] when another round of deconvolution and activation concatenation is appended to FCN-16s and refer to the resultant network as FCN-8s [16]. We test the FCN-8s (Fig. 3c) against the U-Net architectures and extend it one step further, adding yet another block of deconvolution and prior activation concatenation. (A total of three skip connections.) We refer to this network as FCN-4s (Fig. 3d) and compare it as well.

Rather than utilize a fully 3D U-Net framework as in Ciçek *et al.*, we make minor modifications to the U-Net in order to introduce limited 3D context to the model, following the example of [26]. Depicted in Fig. 3e, the three-channel U-Net model utilizes a three-channel 128x128x3 input, shown in yellow, corresponding to a tomographic slice of interest and the slices directly above and below that slice. The mapping of each image channel to a small but contiguous region in 3D space provides quasi-3D context to the model at the cost of requiring the data partitioning scheme described above in order to keep the training and validation data pools completely separate. The remainder of this three-channel U-Net, including 2D pooling and upsampling layers, is identical in form to the 2D-input U-Net, save that its penultimate layer performs an additional 1x1 convolution for channel mixing.

#### 2.4. Model implementation and training parameters

Models were implemented using a Python3-based software stack, built on the computing resource Mustang located at the Air Force Research Laboratory’s DoD Supercomputing Resource Center (AFRL DSRC). The DSRC hardware resource Mustang is an HPE SGI 8600 machine, clocked at 4.87 PFLOPS. Mustang has 24 nodes with NVIDIA Tesla P100 GPUs, on which all model training, validation, and scoring was conducted. Each network was trained and scored on one GPU node at a time, but networks were trained concurrently by utilizing multiple nodes. Typical model training times were 24–36 GPU-hours at the training parameterizations used.

An anaconda environment was created through which dependencies were managed [42]. In order to expedite the training of the CNN models, the Keras API [43] was used to wrap and call the TensorFlow computational backend [44]. Additional computations were aided by the use of scikit-learn for model scoring and numpy for the management of input data and predictions [45,46].

##### 2.4.1. Data augmentation scheme

For model training, reconstructed tomography slices were divided into training and validation sets as described in section 2.2. For each training iteration, an image is selected at random from the training set. That image is then acted upon by an affine transformation,  $T$ . This transform maps the image to a distorted image, by applying compositions of image reflection, translation, scaling, shearing, and rotation and then sizing the image to 128x128 pixels.

Given dummy image coordinates  $(x,y)$ , the effective form of the matrix giving rise to these distortions is given by

$$T = \begin{pmatrix} (-1)^{f_y} \eta_x \cos \theta & -\sin \phi - \theta & \delta x \\ \sin \theta & (-1)^{f_x} \eta_y \cos \theta + \phi & \delta y \\ 0 & 0 & 1 \end{pmatrix}$$

where  $f_x$  and  $f_y$  are inversion indicators (such that  $f_y = 1$ , if the image is to be inverted over the  $y$ -axis,  $f_y = 0$ , otherwise),  $\eta_x$  and  $\eta_y$  scale the original image coordinates,  $\theta$  is an angle of rotation,  $\phi$  is a shear angle, and  $\delta x$  and  $\delta y$  translate the transformed image. During data augmentation, an angular range of  $\theta \in [-90^\circ, 90^\circ]$  was utilized, and horizontal and vertical translations of up to 13 pixels (10% of image size) size were permitted. Zoom parameters were chosen  $\eta_x, \eta_y \in [0.5, 1.5]$ , and  $\phi \in [-45^\circ, 45^\circ]$ . These specific values were chosen to optimize variation in the augmented training data with the requirement that the distorted and translated data contain a sufficient number of crack-associated points to allow for convergent model training. Random tuples of parameters are generated to produce a collection of transformations which are applied to applied to the seed images. In practice, a composition of sequential operations produces the form of the  $T$  above.

This transformation is then left-multiplied to the model inputs such that both the ground truth image and mask are distorted by the same  $T$ , for each training observation. For training iterations where the batch size is not one, distinct  $T$ s are applied to each observation.

For the U-Net, the FC U-Net, and the FCN-8/4s networks, the model input tensor is of size 128x128x1, *i.e.* a single channel mapped to the material density values in each pixel. The predictions returned by these models are classification labels. For these classification labels, zero or one represents the crack-association status of a pixel with indices  $ijk$ . This model, therefore, maps a 2D tomographic slice of material density proxies to a corresponding 2D slice of crack-associated or non-crack-associated pixel labels.

The three-channel U-Net model is trained similarly, save that the model input tensors are of size 128x128x3, corresponding to a stack of three connected slices. The model maps this mini-3D spatial field of material density proxy values to a 2D image of labels corresponding to the classified phases of the voxels in the ‘middle’ slice. (Each  $ijk$  set of indices of the input tensor represents a point in 3-space, so we refer to each set as a voxel, not a pixel.) Thus, the density signals in the out-of-plane dimension with respect to each tomographic slice are able to influence the classification of an in-plane voxel.

##### 2.4.2. Hyperparameters

The model weights are initialized using He initialization [47], and input feature channels are scaled and normalized prior to model training. The loss function selected is binary cross-entropy. Trial training runs were conducted for each model, at learning rates  $\eta \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$  and with batch sizes  $b \in \{1, 4\}$  in order to probe model convergence properties. Learning rates of  $\eta < 10^{-5}$  resulted in slow model convergence, and  $\eta \geq 10^{-3}$  caused loss divergence for all models. The learning rate,  $\eta = 10^{-5}$ , was selected across all models. Model weights were updated during training by gradient backpropagation using the Adam optimizer implemented in the Keras API, with the constant learning rate described above.

Each network was trained with a total of 1000 inputs per epoch, for 100 epochs, for a total of  $10^5$  image observations per model training. (2D inputs are single images; three-channel inputs are mini-stacks of images.) Batch sizes  $b \in \{1, 4\}$  were considered for the 2D-input models, such that when  $b = 4$ , the number of iterations was 250 and the total number of observations remained fixed. At  $b = 1$ , the total number of iterations per epoch was 1000. After each batch, model weights were

updated. Each model was saved after each epoch and the model version that minimized validation loss over the 100 epochs was chosen for comparison. The epoch at which validation loss was at a minimum is given for each model in Table 2.

### 3. Results

#### 3.1. Analytic metrics

Following training, models are scored by generating predictions from the images in the validation set and then comparing these predictions to the annotated ground truth images. Metrics including accuracy, Intersection-over-Union (IoU), and  $\phi$  coefficient [48] are then computed between the two images. The  $\phi$  coefficient is a holistic figure of merit which characterizes the confusion matrix in a two-class system according to

$$\phi = \frac{TP^*TN - FP^*FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Here, TP, TN, FP, FN refer to true positive, true negative, false positive, and false negative identifications. The  $\phi$  coefficient ranges from  $[-1, 1]$ , with values at unity representing only true positive and true negative predictions. Results for each slice are averaged across the validation set and are tabulated in Table 2.

Fig. 4a and 4b plot two of these figures of merit,  $\phi$  and IoU values, for each of the models considered across every cross-section, regardless of whether the original image belonged to the training or the validation set. (Note, the tabulations in Table 2 only include contributions from the validation set.) Images included in the validation set are denoted with solid squares in these figures.

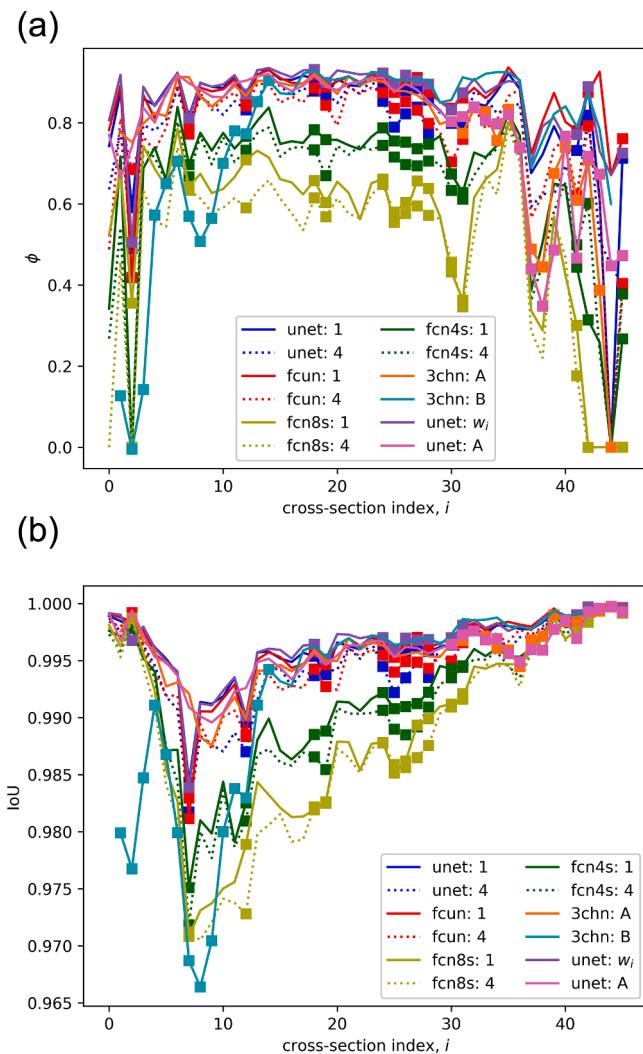
#### 3.2. Visual inspection

Segmentation quality is most intuitively assessed by inspection. In Fig. 5a, we show an illustration of model performance, as tested on the raw image corresponding to  $i_v = 12$ . The grayscale image representing the reconstructed µCT cross-section is reproduced in the first row on the

**Table 2**

Results from model validation. Quantities listed are averaged over the collection of cross-sections comprising the validation sets. Quantities in parentheses indicate the standard deviations of these quantities at the last digit of precision listed. For example,  $0.996(4) = 0.996 \pm 0.004$ .

Network	Batch Size	Accuracy	IoU	$\phi$	nEpochs	nPars
U-Net	1	0.996(4)	0.995 (4)	$0.826 \pm 0.085$	93	37.0 M
U-Net	4	0.994(4)	0.995 (5)	$0.757 \pm 0.136$	94	37.0 M
FC U-Net	1	0.996(4)	0.996 (5)	$0.829 \pm 0.120$	96	40.1 M
FC U-Net	4	0.995(4)	0.995 (4)	$0.768 \pm 0.120$	99	40.1 M
FCN-8 s	1	0.989(8)	0.989 (8)	$0.476 \pm 0.224$	100	196 M
FCN-8 s	4	0.988(8)	0.988 (8)	$0.425 \pm 0.244$	92	196 M
FCN-4 s	1	0.992(6)	0.992 (6)	$0.607 \pm 0.231$	62	197 M
FCN-4 s	4	0.991(8)	0.988 (8)	$0.425 \pm 0.245$	92	197 M
3CH U-Net A	1	0.998(1)	0.998 (1)	$0.630 \pm 0.224$	53	37.0 M
3CH U-Net B	1	0.981(8)	0.981 (8)	$0.560 \pm 0.271$	87	37.0 M
U-Net A	1	0.997(1)	0.997 (1)	$0.652 \pm 0.169$	75	37.0 M
U-Net $w_i$	1	0.996(4)	0.996 (4)	$0.843 \pm 0.106$	99	37.0 M



**Fig. 4.** These plots illustrate model performance as a function of tomographic cross-section index,  $i$ . In (a), the model prediction Matthews correlation coefficients ( $\phi$ ) plotted for all slices, both training and validation. Validation images are denoted in both plots using solid squares. Dashed lines indicate models trained with batch size  $b = 4$ . In (b), line plots illustrate model IoU scores.

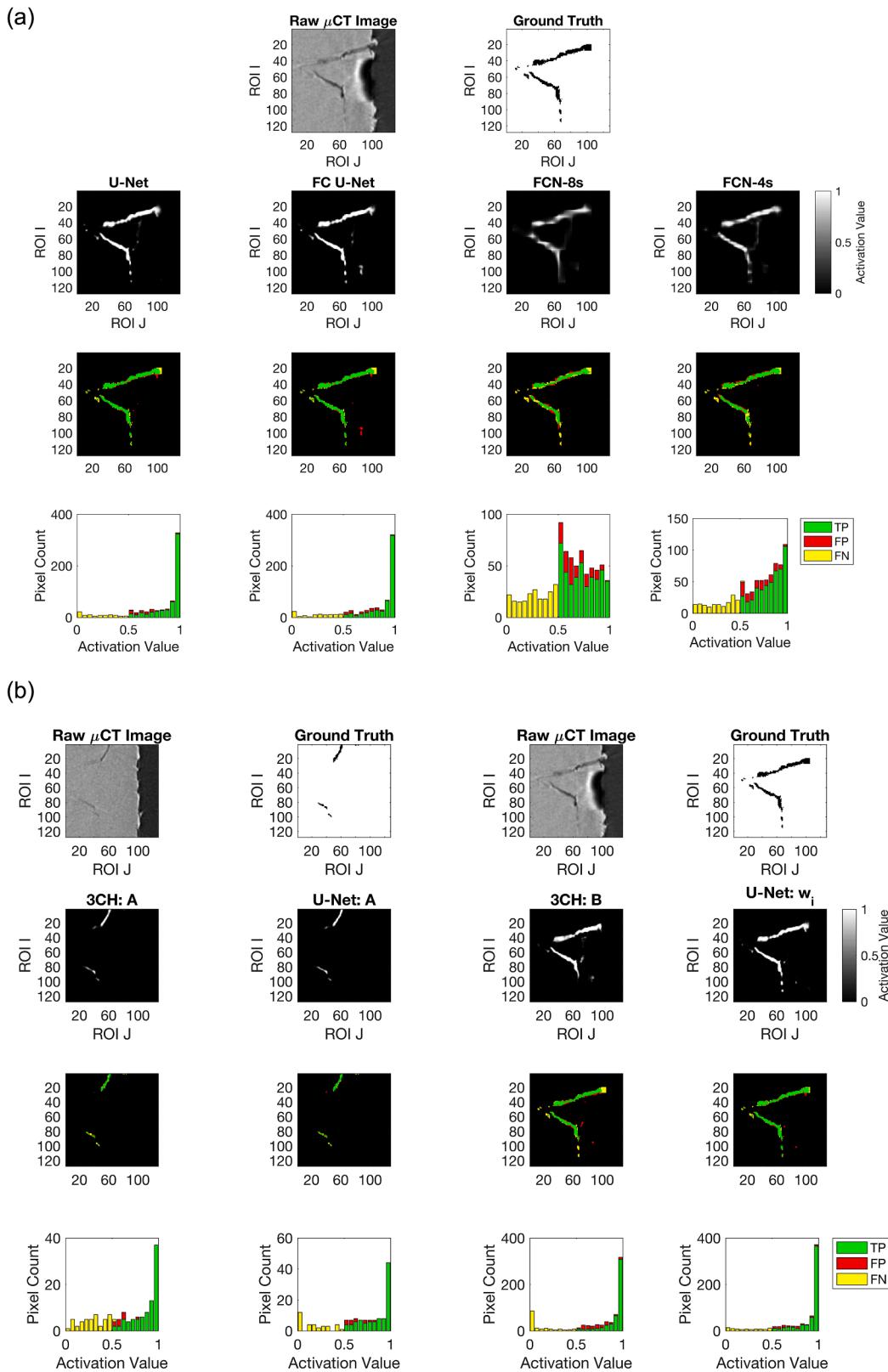
left. At right in the first row is the ground truth annotation, the model target.

The second row illustrates the model output activation, which in this context is the probability the given pixel should be segmented as part of the crack. A perfect segmentation would appear as the inverse coloration of the ground truth, with activation values  $\alpha_{ij} > 1/2$  only within pixels marked black on the ground truth panel. In row three, we give a colorized taxonomy for segmented pixels: true positives identifications are marked in green, false positives in red, and false negatives in yellow. In row four, we histogram the model activations shown in row two and shade each bar using the same color code as in row three.

Thus, in rows three and four, green regions represent pixels that the model has segmented correctly. Type one and type two errors are colored red and yellow respectively.

Fig. 5b shows example results for the three-channel U-Net model under training schemes A (column 1) and B (column 3), as well as the single channel U-Net model under training schemes A (column 2) and  $w_i$  (column 4).

Because cross-section  $i_v = 12$  is not in the validation set under training scheme A, we show cross-section  $i_v = 31$  instead. Note, the relative scaling of the activation histograms in row four reflects the

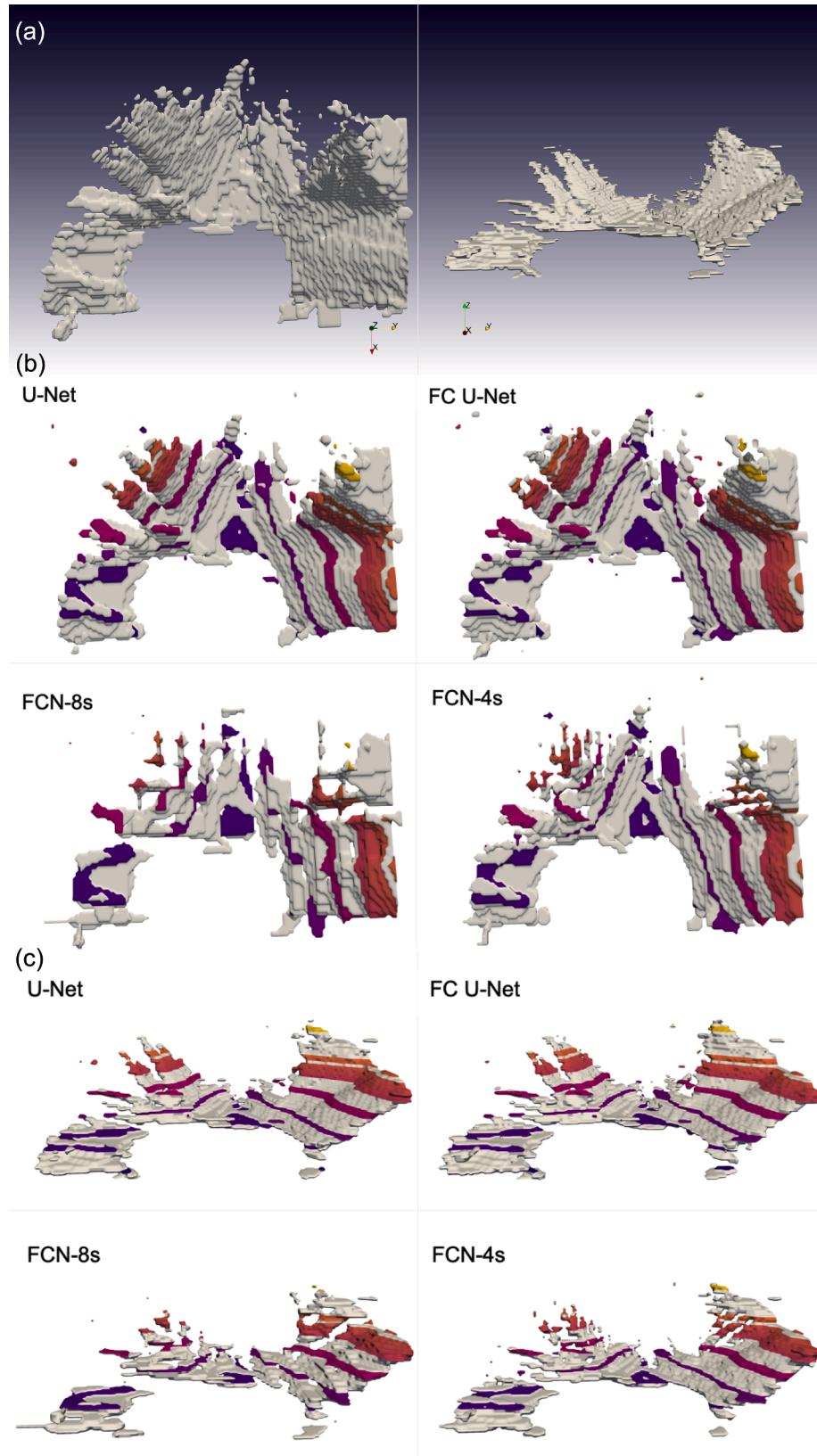


**Fig. 5.** Examples of model segmentations and activations for the networks tested. The top rows illustrate examples of grayscale  $\mu$ CT reconstructions and the corresponding ground truth annotations which highlight the crack features. The ground truths were produced by hand, by picking pixels which correspond to the internal crack features. The second row in each (a) and (b) shows the sigmoid activations from the output layer of each model listed. This can be interpreted to be the model's predicted probability that each pixel should be segmented as part of the crack. In each row three, we highlight true positive identifications in green, with false negative and false positive errors denoted in yellow and red, respectively. In the fourth rows, colored histograms illustrate the distribution of model activations and show the model's classification of the corresponding pixel.

comparative sparsity of crack-associated points in this cross-section.

We also examine model performance in 3D, rendering first the ground-truth crack surface (Fig. 6a) for comparison. We also render the model segmentations (Fig. 6b), and color layers from the validation set

according to a diverging colorscale which encodes the index of the cross-section. Layers from within the training set are rendered in gray. Fig. 6c illustrates the same segmentations pictured in Fig. 6b, but from a perpendicular view.



**Fig. 6.** (a) 3D renderings of the ground truth annotations used to train and score the model, shown from two perpendicular views (b) Top-down segmentations produced by various deep learning models. Gray slices are segmentations made of slices which were augmented to form the model training sets. Segmentations depicted in these gray slices, however, are produced from un-augmented versions of these training images. Colored layers represent the predicted segmentations for slices forming the validation datasets. (c) Segmentations produced by each model, as in (b), but from a perpendicular point-of-view.

## 4. Discussion

Generally, performance metrics and visual inspection both indicate a high degree of model performance for this application. Decreases in IoU metrics appear strongly correlated to the distribution of the crack-area, i.e. lower scores where there are more positive class pixels to predict. While this is true,  $\phi$  scores are relatively consistent in these areas, except at points near the periphery where some models fail to correctly segment small crack features. By visual inspection, the morphologies of crack features in 3D are fairly well represented. In all cases evaluated, models trained with batch size  $b = 4$  underperform models trained with batch size  $b = 1$ . Given that many models achieved performance maxima close to the end of the allowed training time, model performance could potentially be improved by further training.

### 4.1. U-Net and FC U-Net

The U-Net is a competitive model. Its predictions track crack features at the micron scale. In Fig. 5a, row two, fine features at the crack periphery are well captured, but both of these models give false positive predictions within the material bulk as well as at the crack interface. These additional predictions capture intensity variations in the input image which may be caused by artifactual phase contrast present in the reconstructed  $\mu$ CT image. This phase contrast is visible at material-air interfaces, particularly at the FIB notch, visible as a characteristic dark-light fringe in the raw  $\mu$ CT image in Fig. 5a. The large peak at one in the histogram of activation scores (Fig. 5a, row 4) for the U-Net and FC-U-Net indicates most true positives are identified with high probability.

Focusing on the more encompassing metric of  $\phi$  computed, the U-Net and the FC U-Net rank highest among the architectures considered. While the results in Table 2 might seem to initially imply the FC U-Net out-performs the unmodified version in a general sense, we note that this out-performance occurs where there are few crack-associated points to predict, particularly for validation image indices  $i_v > 40$ . (Refer to  $\phi$  values on the right-hand side of Fig. 4a) As this is the periphery of the crack feature, the characteristic length-scale of the crack features present is smaller. Improved performance here may reflect a potential benefit of learning the upsampling kernel and an associated improvement in identifying small features more effectively.

Future research could consider the effects of varying the upsampling filter kernel size as a parameter with which to further improve the FC U-Net. Differences in training times between the two models were marginal, as were differences in memory requirements due to model size. Smooth model convergence properties were observed for both models, allowing for the application of an early stopping criterion to prevent overfitting and reduce training times, if desired.

### 4.2. FCN-8s and FCN-4s

#### 4.2.1. FCN-8s

Examination of the segmentations from this model show poor performance when compared to the U-Net architectures and to the FCN-4s. Notably, discretization in the model activation fails to resolve fine spatial features which are resolved by the other architectures. This discretization is due to the fewer number of skip connections in this model, a change which deprives the decoding branch of the network with finer-scale spatial activations [16]. The decoding branch is more rapidly upsampled, which also serves to enlarge unrefined spatial activations. These are visible in the intensity patterns and stair-stepping within the two-dimensional model activation (Fig. 5a, row two) and also three-dimensionally in the pixelated prediction shown in Fig. 6b. Thus, this model suffers from a greater number of false-positive and false-negative predictions.

Despite this, classified pixels preserve large scale spatial structure. With over five times more parameters than the U-Net class of models,

gradient back-propagation is comparatively more expensive and model training is more time consuming. These additional parameters are not well served for the task of two-phase semantic segmentation, but perhaps would be more useful for cases requiring greater semantic power, e.g. multiple phases.<sup>1</sup>

#### 4.2.2. FCN-4s

The FCN-4s achieves better performance than the FCN-8s on the metrics tested here. Examination of the activations in Fig. 5a show that the fine features are more visible and that the intensity of the fringing and aliasing is reduced. The distribution of activation values skews toward unity, indicating more predictions are made further from the decision boundary (i.e. the model is more confident in the determination). The additional skip connection improves the resolution of the prediction, thereby improving the quality of the prediction. For both the FCN-4s and the FCN-8s, the large number of feature channels in the spatially coarsest convolutional layers (4096 versus 1024) do not improve model performance over the U-Net models. As these layers provide space to encode combination of low-level features, the FCN-(4,8)s models may be better suited to computer vision classification tasks requiring differentiation between contexts or integration of various levels of abstraction, e.g. predicting a dog's breed from an image.

### 4.3. Three-channel U-Net

While the three-channel U-Net performed better than the FCN-(4,8)s models, the additional out-of-plane context from adjacent tomographic cross-sections did not improve performance over the 2D-input U-Net models. Examining Fig. 5b, we note that under scheme B model predictions included a large number of false negatives identified with near-zero activation, far from the decision boundary at  $\alpha_{ij} = 1/2$ . These regions are 'missed,' and show up as contiguous yellow regions in row three of Fig. 5b.

Results are better when the network is trained using scheme A, but still does not realize benefit above the 2D-input U-Net model. From Fig. 2a, we infer the number of crack-associated points present in the training set influences the model performance. From this perspective, we might expect the results under scheme A to be representative of the three-channel model efficacy.

Due to the requirement that training and validation datasets both be physically contiguous along the sample loading direction, the validation sets used in each scheme each contain points at the crack periphery. Here, features are sparser, and signals are less dense on each image. If additional representative training data at the crack center were available, out-of-plane context could be more predictive.

### 4.4. Effects of model weighting

Under scheme  $w_i$ , the U-Net model enjoys a modest improvement from an analytical perspective. That said, the weights change the loss function and thus the task, but the underlying training/validation data sets do not contain any new information. This information could be reincorporated by training the models considered with full image precision. Further investigation should also focus on whether weighting the loss function in this way allows for model training from larger image regions, despite the physical sparsity of the crack.

### 4.5. Performance at smaller crack sizes

As the segmentation tools may be used to study crack growth and

<sup>1</sup> We note this network's lineage from the VGG-16 network and point out that performance on the ILSVRC and PASCAL VOC [10,41] metric tasks requires sensitivity to a large number of object classes and therefore require deep and narrow layers to identify high-level feature abstractions.

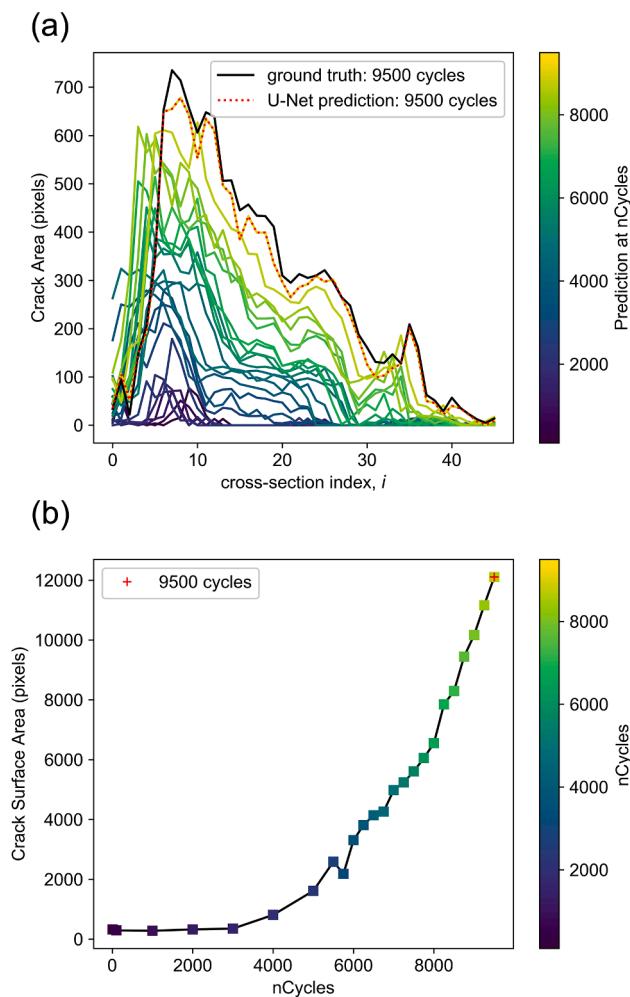
propagation, it is important to consider whether these approaches properly capture crack features at smaller crack sizes. To this end, the U-Net model was used to segment potential crack features in the ROI from the 25 tomography scans not yet evaluated. In Fig. 7a, we illustrate the number of crack-segmented pixels per cross-section as a function of the number of load cycles. The slice index is plotted on the horizontal axis and the function height shows the integrated crack area on that slice. Each distribution is colorized by the number of fatigue load steps conducted prior to that tomography scan. We see that the distributions generally grow taller and wider as material cracks grow and spread as the load steps progress. Distributions evolve smoothly as the same features are found repeatedly in independent scans.

In Fig. 7b, we show the integral of each of these curves, total crack area, plotted against the number of fatigue cycles. A proper segmentation should preserve monotonicity of this quantity, as it is generally nonphysical for increasing load to shrink a fatigue crack. We see in Fig. 7b, that this monotonicity is mostly preserved, with one exception. Subsequent investigation shows that phenomenon is explained by the

crack feature shifting slightly out of the fixed ROI, as additional vertical registration between tomographic states was not conducted outside of the experimental alignment. This could be corrected in future work by centering the notched regions in each tomographic scan prior to comparison of the distributions in Fig. 7a.

We also visually confirm that the U-Net captures crack development and growth by plotting the model segmentation of cross section  $i_v = 12$  at earlier load cycles. The top row of Fig. 8 illustrates the raw grayscale tomographic reconstructions from scans after 2000, 5500, 6500, and 8000 fatigue cycles. The crack is not present after 2000 cycles, but becomes visible after 5500 cycles as a diagonal, dark, linear, and ultimately triangular form at the center of the image. The second row of the Figure illustrates the 2D-input U-Net segmentations of the raw images in row one. The model predicts no crack-pixels after 2000 cycles, while the dark regions in row two, columns 2–4, correspond to pixels that the model predicts are part of the growing crack shown in row one.

From these examinations, we advance that these models are strong tools for performing segmentation operations in these applications.



**Fig. 7.** These distributions illustrate the 2D-input U-Net model's segmentation of the tomography from earlier load states, which were not considered as part of the model development. In (a), the segmented crack area versus cross-section slice index, colored by the number of fatigue cycles seen. The distributions evolve smoothly to the ultimate ground truth distribution at 9500 cycles, depicted in black. The 2D-input U-Net model's segmentation of this scan is highlighted by the red dashed line. In (b), the integrated crack area versus the number of fatigue cycles seen. Each colored point in (b) corresponds to the integral of the curve with the same color. The integrated crack area at the final state at 9500 cycles is marked by a red plus. We note that monotonicity is preserved in (b), up to the limits of the vertical registration between datasets.

#### 4.6. General discussion

Some of the deep learning-based approaches outlined convincingly mimic the human-performed annotations. Considering the high costs of producing annotated data volumes, the viability of these techniques represents significant potential for cost and time savings. These successes also broaden the possibility frontier associated with the processing of large datasets produced by myriad imaging modalities including x-ray characterization, surface microscopy, serial section data, and others.

We may consider ways to further increase performance of the models explored. During data augmentation, Poisson noise could be added to desensitize the model to random errors endemic to the experimental or reconstruction pipelines. Pixel-by-pixel weights may also be supplied with each training image, in order to further sensitize the model to features of particular interest. One might also apply a sliding-window majority voting scheme or combine these networks into an ensemble learning model which aggregates the predictions of each.

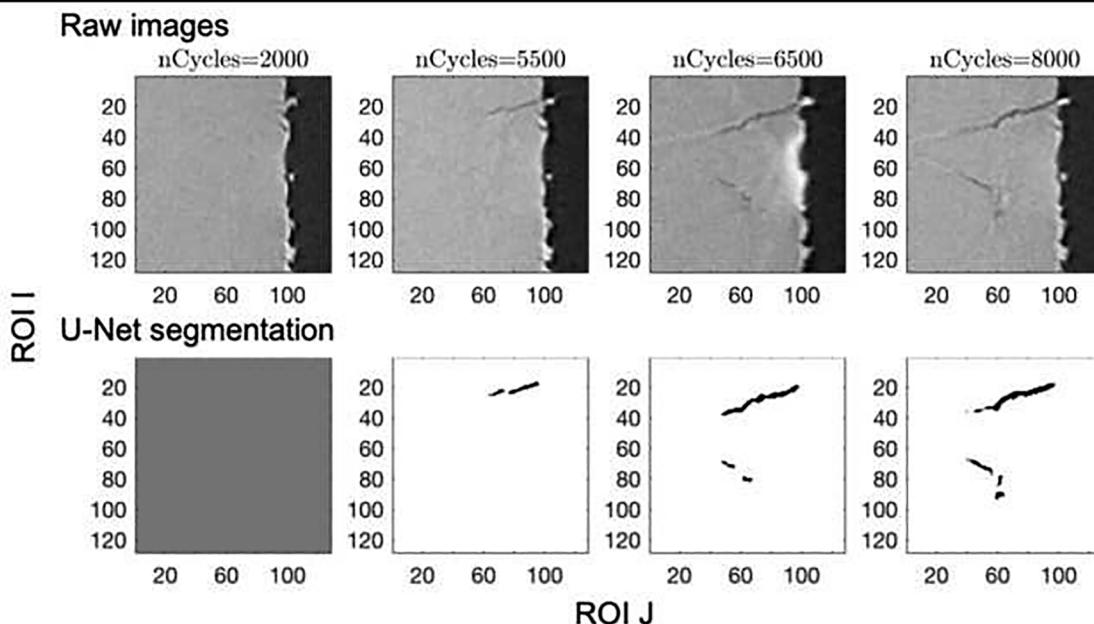
Several avenues for additional work are apparent. First, reconstruction artifacts sometimes inhibit the ability of current computational tools to discriminate carefully between features, accurately estimate phase or void fractions, or confirm the presence of certain precipitates. It would be straightforward to annotate a  $\mu$ CT dataset containing these artifacts and apply a framework similar to those discussed above to detect them. Once identified, pixel intensity values can be corrected selectively after reconstruction, specifically within regions marred by the artifacts. By this method, existing algorithms for which these artifacts pose prohibitive issues may be extended to new use cases. Similarly, many existing microscopy modes besides  $\mu$ CT also produce 3D datasets of similar size and breadth. These methods could equally be adapted to fit those cases, as well.

#### 5. Conclusion

In this work, we test five convolutional neural network models in their ability to segment the material density field of a LSHR nickel-base superalloy sample repeatedly imaged by micro-computed tomography. We find these approaches demonstrate a high degree of efficacy and efficiency, segmenting all pixels in an image in seconds. These segmentations provide annotations similar in quality to those furnished by experts, making possible analysis of high data volumes and novel scientific frontiers.

#### 6. Author work statement

DBM performed analysis, collected data, and wrote the manuscript. PAS and WDM organized the experiment, collected data, managed the



**Fig. 8.** Example model segmentation performance at nCycles < 9500. In the first row, ground truth raw tomographic reconstructions from scans after 2000, 5500, 6500, and 8000 fatigue cycles. In the second row, the U-Net segmentation of any crack features determined by the model to be within the ROI.

project, and edited the manuscript. SS and JSP collected data. PK assisted with data collection and performed reconstructions.

#### 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.

#### Acknowledgements

The financial support of the Air Force Office of Scientific Research with Dr. Jay Tiley as Program Manager is gratefully acknowledged. WDM and PAS also acknowledge support from the Materials and Manufacturing Directorate of the U.S. Air Force Research Laboratory. DBM is thankful for the high-performance computer time and resources provided by the U.S. Air Force Research Laboratory (AFRL) DoD Supercomputing Resource Center (DSRC) that supported the model training and validation studies performed in this work. 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.

#### Data Availability Statement

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.

#### References

- [1] J.C. Schuren, P.A. Shade, J.V. Bernier, S.F. Li, B. Blank, J. Lind, P. Kenesei, U. Lienert, R.M. Suter, T.J. Turner, D.M. Dimiduk, J. Almer, New opportunities for quantitative tracking of polycrystal responses in three dimensions, *Curr. Opin. Solid State Mater. Sci.* 19 (4) (2015) 235–244, <https://doi.org/10.1016/j.coosms.2014.11.003>.
- [2] D.P. Naragani, P.A. Shade, P. Kenesei, H. Sharma, M.D. Sangid, X-ray characterization of the micromechanical response ahead of a propagating small fatigue crack in a Ni-based superalloy, *Acta Mater.* 179 (2019) 342–359, <https://doi.org/10.1016/j.actamat.2019.08.005>.
- [3] S.R. Stock, Recent advances in X-ray microtomography applied to materials, *Int. Mater. Rev.* 53 (3) (2008) 129–181, <https://doi.org/10.1179/174328008X277803>.
- [4] D.B. Menasche, P.A. Shade, J. Lind, S.F. Li, J.V. Bernier, P. Kenesei, J.C. Schuren, R. M. Suter, Correlation of thermally induced pores with microstructural features using high energy X-rays, *Metall. Mater. Trans. A* 47 (11) (2016) 5580–5588, <https://doi.org/10.1007/s11661-016-3712-3>.
- [5] S.S. Singh, J.J. Williams, P. Hruby, et al., *In situ* experimental techniques to study the mechanical behavior of materials using X-ray synchrotron tomography, *Integrating Mater.* 3 (2014) 109–122, <https://doi.org/10.1186/2193-9772-3-9>.
- [6] J.W. Gibbs, K.A. Mohan, E.B. Gulsoy, A.J. Shahani, X. Xiao, C.A. Bouman, M. De Graef, P.W. Voorhees, The three-dimensional morphology of growing dendrites, *Sci. Rep.* 5 (1) (2015), <https://doi.org/10.1038/srep11824>.
- [7] J. Berkowitz, *Big data hits beamline*, *DEIXIS Magazine* (2013).
- [8] A.-O. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc., 1st edition, 2017.
- [9] M. Everingham, L. Van Gool, C.K.I. Williams, et al., The PASCAL Visual Object Classes (VOC) Challenge, *Int. J. Comput. Vision* 88 (2010) 303–338, <https://doi.org/10.1007/s11263-009-0275-4>.
- [10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L.i. Fei-Fei, ImageNet large scale visual recognition challenge, *Int. J. Comput. Vision* 115 (3) (2015) 211–252, <https://doi.org/10.1007/s11263-015-0816-y>.
- [11] X. Zhou, R. Takayama, S. Wang, et al. Automated segmentation of 3D anatomical structures on CT images by using a deep convolutional network based on end-to-end learning approach. *Medical imaging 2017: image processing* 10133:1013324. International Society for Optics and Photonics, 2017.
- [12] P. Sahu, D. Yu, H. Qin, Apply lightweight deep learning on internet of things for low-cost and easy-to-access skin cancer detection. In Chen P-H, Zhang J., (eds) *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications*. SPIE. 1057912 (2018). doi: 10.1117/12.2293350.
- [13] M.H. Hesamian, W. Jia, X. He, P. Kennedy, Deep learning techniques for medical image segmentation: achievements and challenges, *J. Digit. Imaging* 32 (4) (2019) 582–596, <https://doi.org/10.1007/s10278-019-00227-x>.
- [14] M. Weber, A. Buerle, M. Schmidt, et al., Automatic identification of crossovers in cryo-EM images of murine amyloid protein A fibrils with machine learning, *J. Microsc.* 277 (1) (2020) 12–22, <https://doi.org/10.1111/jmi.v277.11111/jmi.12858>.
- [15] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444, <https://doi.org/10.1038/nature14539>.
- [16] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*, pp. 3431–3440.
- [17] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015. Lecture Notes in Computer Science, 9351 (2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- [18] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM* 60 (6) (2017) 84–90, <https://doi.org/10.1145/3065386>.

- [19] X. Liu, Z. Deng, Y. Yang, Recent progress in semantic image segmentation, *Artif. Intell. Rev.* 52 (2) (2019) 1089–1106, <https://doi.org/10.1007/s10462-018-9641-3>.
- [20] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, *Comput.-Aided Civ. Infrastruct. Eng.* 32 (5) (2017) 361–378, <https://doi.org/10.1111/mice.2017.32.issue-510.1111/mice.12263>.
- [21] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, *Autom. Constr.* 104 (2019) 129–139, <https://doi.org/10.1016/j.autcon.2019.04.005>.
- [22] Ö. Çiçek, A. Abdulkadir, S.S. Lienkamp, et al., 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin S., Joskowicz L., Sabuncu, M., Unal, G., Wells, W. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. MICCAI 2016. Lecture Notes in Computer Science, 9901 (2016). doi: 10.1007/978-3-319-46723-8\_49.
- [23] F. Milletari, N. Navab, S. Ahmadi, V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, in: 2016 Fourth International Conference on 3D Vision (3DV), 2016, pp. 565–571, <https://doi.org/10.1109/3DV.2016.79>.
- [24] K. Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, *Computing Resource Repository* 1409 (2015) 1556.
- [25] C. Szegedy, et al., Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1–9 (2015), doi: 10.1109/CVPR.2015.7298594.
- [26] T. Strohmann, K. Bugelnig, E. Breitbarth, Semantic segmentation of synchrotron tomography of multiphase Al-Si alloys using a convolutional neural network with a pixel-wise weighted loss function, *Sci. Rep.* 9 (1) (2019), <https://doi.org/10.1038/s41598-019-56008-7>.
- [27] S. Evsevleev, S. Paciornik, G. Bruno, Advanced deep learning-based 3D microstructural characterization of multiphase metal matrix composites, *Adv. Eng. Mater.* 22 (4) (2020) 1901197, <https://doi.org/10.1002/adem.v22.410.1002/adem.201901197>.
- [28] S.M. Azimi, D. Britz, M. Engstler, et al., Advanced steel microstructural classification by deep learning methods, *Sci. Rep.* 8 (1) (2018), <https://doi.org/10.1038/s41598-018-20037-5>.
- [29] W.D. Musinski, P.A. Shade, D.C. Pagan, J.V. Bernier, Statistical aspects of grain-level strain evolution and reorientation during the heating and elastic-plastic loading of a Ni-base superalloy at elevated temperature, *Materialia* 16 (2021) 101063, <https://doi.org/10.1016/j.mtla.2021.101063>.
- [30] S.L. Semiatin, K.E. McClary, A.D. Rollett, C.G. Roberts, E.J. Payton, F. Zhang, T.P. Gabb, Microstructure evolution during supersolvus heat treatment of a powder metallurgy nickel-base superalloy, *Metallurgical Mater. Trans. A* 43 (5) (2012) 1649–1661.
- [31] S. Semiatin, K.E. McClary, A.D. Rollett, C.G. Roberts, E.J. Payton, F. Zhang, T.P. Gabb, Plastic flow and microstructure evolution during thermomechanical processing of a pm nickel-base superalloy, *Metallurgical Mater. Trans. A* 44 (6) (2013) 2778–2798.
- [32] P.A. Shade, D.B. Menasche, J.V. Bernier, et al., Fiducial marker application method for position alignment of in-situ multimodal X-ray experiments and reconstructions, *J. Appl. Crystallogr.* 49 (2) (2016) 700–704.
- [33] P.A. Shade, B. Blank, J.C. Schuren, et al., A rotational and axial motion system load frame insert for in situ high energy x-ray studies, *Rev. Sci. Instrum.* 86 (9) (2015) 093902, <https://doi.org/10.1063/1.4927855>.
- [34] H.F. Poulsen, Three-dimensional X-ray diffraction microscopy: mapping polycrystals and their dynamics, *Springer Tracts in Modern Physics* (2004).
- [35] U. Lienert, S.F. Li, C.M. Hefferan, J. Lind, R.M. Suter, J.V. Bernier, N.R. Barton, M.C. Brandes, M.J. Mills, M.P. Miller, B. Jakobsen, W. Pantleon, High-energy diffraction microscopy at the Advanced Photon Source, *JOM* 63 (7) (2011) 70–77.
- [36] F. Wang, F. De Carlo, I. Foster, et al., Quasi-real-time x-ray microtomography system at the advanced photon source, *SPIE Proceedings* 3772 (1999) 318.
- [37] B.A. Dowd, G.H. Campbell, R.B. Marr, et al., Developments in synchrotron x-ray computed microtomography at the national synchrotron light source, *SPIE Proceedings* 3772 (1999).
- [38] A. Khounsary, P. Kenesei, J. Collins, G. Navrotski, J. Nudell, High energy x-ray micro-tomography for the characterization of thermally fatigued GlidCop specimen, *J. Phys. Conf. Ser.* 425 (21) (2013) 212015, <https://doi.org/10.1088/1742-6596/425/21/212015>.
- [39] J.M. Johnson, T.M. Khoshgoftaar, Survey on deep learning with class imbalance, *J. Big Data* 6 (2019) 27, <https://doi.org/10.1186/s40537-019-0192-5>.
- [40] G. King, L. Zeng, Logistic regression in rare events data, *Political Analysis* 9 (2) (2001) 137–163.
- [41] J.T. Springenberg, A. Dosovitskiy, T. Brox, et al., Striving for simplicity: the all convolutional net, *Computing Research Repository* 1412 (2015) 6806.
- [42] Anaconda Software Distribution. Computer software. Vers. 2.3.6 Anaconda, (2016). <https://anaconda.com>.
- [43] F. Chollet et al. Keras. (2015). <https://keras.io>.
- [44] Abadi, M. et al. TensorFlow: Large-scale machine learning on heterogeneous systems. (2015). [tensorflow.org](https://tensorflow.org).
- [45] F. Pedregosa, G. Varoquaux, V. Gramfort, et al., Scikit-learn: Machine learning in Python, *J. Machine Learning Res.* 12 (2011) 2825–2830.
- [46] P. Virtanen, R. Gommers, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272 (2020).
- [47] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *IEEE International Conference on Computer Vision (ICCV 2015)*. 1502 (2015). 10.1109/ICCV.2015.123.
- [48] B.W. Matthews, Comparison of the predicted and observed secondary structure of t4 phage lysozyme, *Biochim. Biophys. Acta (BBA) – Protein Structure* 405 (2) (1975) 442–451.