Fast Segmentation of 4D Microtomography Volumes from Core-flooding Experiments in Porous Rock using Convolutional Neural Network

YILI YANG[1], Sohan Seth[1], Ian B. Butler[1], and Florian Fusseis[1]

[1]University of Edinburgh

November 21, 2022

Abstract

Multiphase fluid flow in porous media has been extensively studied for its applications in carbon capture and storage, hydrocarbon recovery, aquifer contamination, soil hydrology and subsurface energy resources. Fluid displacement in porous media can be investigated using highly time-resolved synchrotron X-ray microtomography. One consequence of extremely fast imaging can be compromised image quality, including noise and decreased contrast, which makes the images hard to be segmented. We trained an established convolutional neural network (CNN) architecture (U-Net) with 18,072 images from multiphase flow experiments generated by synchrotron μCT. The trained neural network can segment synchrotron μCT images of core-flooding experiments rapidly and accurately without any pre-processing of the raw image. Segmenting one μCT scan volume of size 1004x496x496 takes 5.6 minutes on an Nvidia Quadro K5200 GPU, while a conventional segmentation pipeline using CPU for the same size data takes 50.2 minutes. On the test dataset, the AUC-ROC score of individual class reached above 0.99 and the mean accuracy of the three segmentation classes reached 99%. The average IoU of the three classes is 0.98. The accuracy of the CNN segmentation is of the same order as conventional methods but it is significantly faster.

Fast Segmentation of 4D Microtomography Volumes from Core-flooding Experiments in Porous Rock using Convolutional Neural Network

Y. Yang, S. Seth, I. B. Butler, F. Fusseis

University of Edinburgh

Key Points:

- Machine learning segmentation enables fast segmentation of µCT data and can reduce processing time by ten-fold.
- Machine learning segmentation quality is insensitive to µCT noise and ring artefacts.
- Machine learning segmentation is applicable to the fast processing of large datasets such as those from time resolved synchrotron microtomography.

Corresponding author: Yili Yang, yili.yang@ed.ac.uk

Abstract

Multiphase fluid flow in porous media has been extensively studied for its applications in carbon capture and storage, hydrocarbon recovery, aquifer contamination, soil hydrology and subsurface energy resources. Fluid displacement in porous media can be investigated using highly time-resolved synchrotron X-ray microtomography. One consequence of extremely fast imaging can be compromised image quality, including noise and decreased contrast, which makes the images hard to be segmented. We trained an established convolutional neural network (CNN) architecture (U-Net) with 18,072 images from multiphase flow experiments generated by synchrotron µCT. The trained neural network can segment synchrotron µCT images of core-flooding experiments rapidly and accurately without any pre-processing of the raw image. Segmenting one µCT scan volume of size $1004 \times 496 \times 496$ takes 5.6 minutes on an Nvidia Quadro K5200 GPU, while a conventional segmentation pipeline using CPU for the same size data takes 50.2 minutes. On the test dataset, the AUC-ROC score of individual class reached above 0.99 and the mean accuracy of the three segmentation classes reached 99%. The average IoU of the three classes is 0.98. The accuracy of the CNN segmentation is of the same order as conventional methods but it is significantly faster.

1 Introduction

Experimental studies of fluid flow related processes in porous media, including multiphase flow and reactive flow, have increasingly made use of X-ray microtomography (µCT) techniques to image fluid distributions and changes in porous media in-situ. These studies range from those which employ laboratory µCT instruments (e.g. Pak et al., 2015; AlRatrout et al., 2018) to those which employ synchrotron µCT to generate 4D time resolved data that reveals dynamic fluid flow processes (e.g. Berg et al., 2014; Reynolds et al., 2017; Berg et al., 2013). The high photon flux at synchrotron X-ray sources enables experimenters to acquire multiple µCT scan volumes, each scan lasting a few seconds or less, over hours of experimentation. With 4D datasets often approaching a few terabytes and comprising many 10s to 100s of discrete data volumes, the efficiency of image processing and segmentation can present a bottleneck to downstream data analysis. Furthermore, fast scans with brief exposure times and few projections, combined with progressive damage to scintillators resulting from high X-ray fluxes can lead to increased

noise, image artifacts and low contrast, making image segmentation both difficult and time consuming.

µCT image segmentation is part of a µCT image processing workflow. By segmentation, reconstructed µCT images are classified into labelled images such that objects of interest in the images are separated for further analysis. Conventionally, the image quality of the reconstructed µCT images is evaluated at first. Then a selection of denoising filters are tested based on the type of noise and artefacts present in the reconstructed images. The denoised images are evaluated again for the difficulty of segmentation, and then tested with different segmentation algorithms such as global thresholding, watershed (Neubert & Protzel, 2014), random walker (Grady, 2006) or adaptive thresholding (Sauvola & Pietikäinen, 2000). Due to the, often, extremely large volume of data, the optimal processing workflow is decided based on both effectiveness and efficiency.

Although the conventional segmentation workflow can produce accurate and reliable results, there are two factors that make it suboptimal for large datasets of 10s to 100s of µCT scan volumes. First, it is slow, and in the case of the data used in this study taking approximately 50 minutes per µCT scan volume on average (for computing specifications see section 2). Second, the conventional segmentation workflow involves significant human supervision and decision making, and due to variations in the degree of noise and artefacts between µCT scan volumes, the parameters of each processing step often need to be hand-tuned for each µCT scan volume to achieve reliable segmentation results.

Recently, deep learning segmentation methods have been increasingly used on medical CT data (e.g. Skourt et al., 2018; Weston et al., 2019). Although X-ray µCT has been a conventional imaging technique in multiphase fluid flow study, deep learning methods are mainly used for predicting fluid flow and porous media properties (e.g. Mo et al., 2019; Alqahtani et al., 2020; Kanin et al., 2019). For example, Karimpouli and Tahmasebi (2019) segmented different mineral phases of Berea sandstone µCT images using a deep learning method. However, to our knowledge, there is no published work on segmenting µCT images of multiphase fluid flow in porous rocks using a deep learning method yet.

Unlike conventional segmentation methods that only take a single, or a simple combination of features into account, deep neural networks allow abstraction of multiple lev-

els of features, such as curves, lines and even complex patterns of data, and use such features to perform segmentation (Rawat & Wang, 2017). CNNs (LeCun et al., 2015) are a class of deep neural network that have been proved effective in visual recognition tasks (e.g. Krizhevsky et al., 2012; Long et al., 2015; Girshick et al., 2014). Ronneberger et al. (2015) proved the reliability of CNN in segmenting low-contrast gray scale biomedical images, and introduced the U-Net architecture. Since experimental multiphase flow data may also show the characteristics of gray scale images with low-contrast objects, we have chosen to use the this architecture to train a segmentation model to yield accurate and precise segmentation of multiple tomographic volumes.

CNN is a supervised learning method that requires access to an adequate amount of manually annotated training samples, i.e., it requires training images where a human has established the boundaries between different classes. Although time consuming, manual annotation is sensible when objects are easily and readily identifiable. However, unlike conventional images with relatively smooth and simple boundaries, pore scale experimental images have highly irregular, multi-scale, and even fractal boundaries which make manual annotation difficult and imprecise (Figure 1). For the experimental µCT data in our study, manual annotation has three main problems: 1) the highly irregular and multi-scale boundaries, 2) similar grey scale for different objects, i.e., brine and rock, and 3) noise and artefacts. Therefore, we acquired the ground truth segmentation of the input images using a weakly supervised algorithm, with a few pre-processing steps such as denoising and masking (details of this approach are in section 2.2).

Our goal is to reproduce, or emulate, the segmentation generated by the weakly supervised approach, using machine learning to achieve an accurate, fast and scaleable result. By training the CNN segmentation model, it learns the 'relation' between the raw µCT image and the optimal segmentation regardless of the processing steps leading to the segmentation. Given an input of raw µCT image, the trained model can produce the desired segmentation result that has the same 'relation' to the input image as it learned from the training.

In this contribution we trained an U-Net architecture to efficiently segment synchrotron µCT images obtained during core-flooding experiments. The images comprise three classes of objects: rock, oil and water. The trained CNN network is able to segment the three classes directly from reconstructed images accurately and rapidly with-

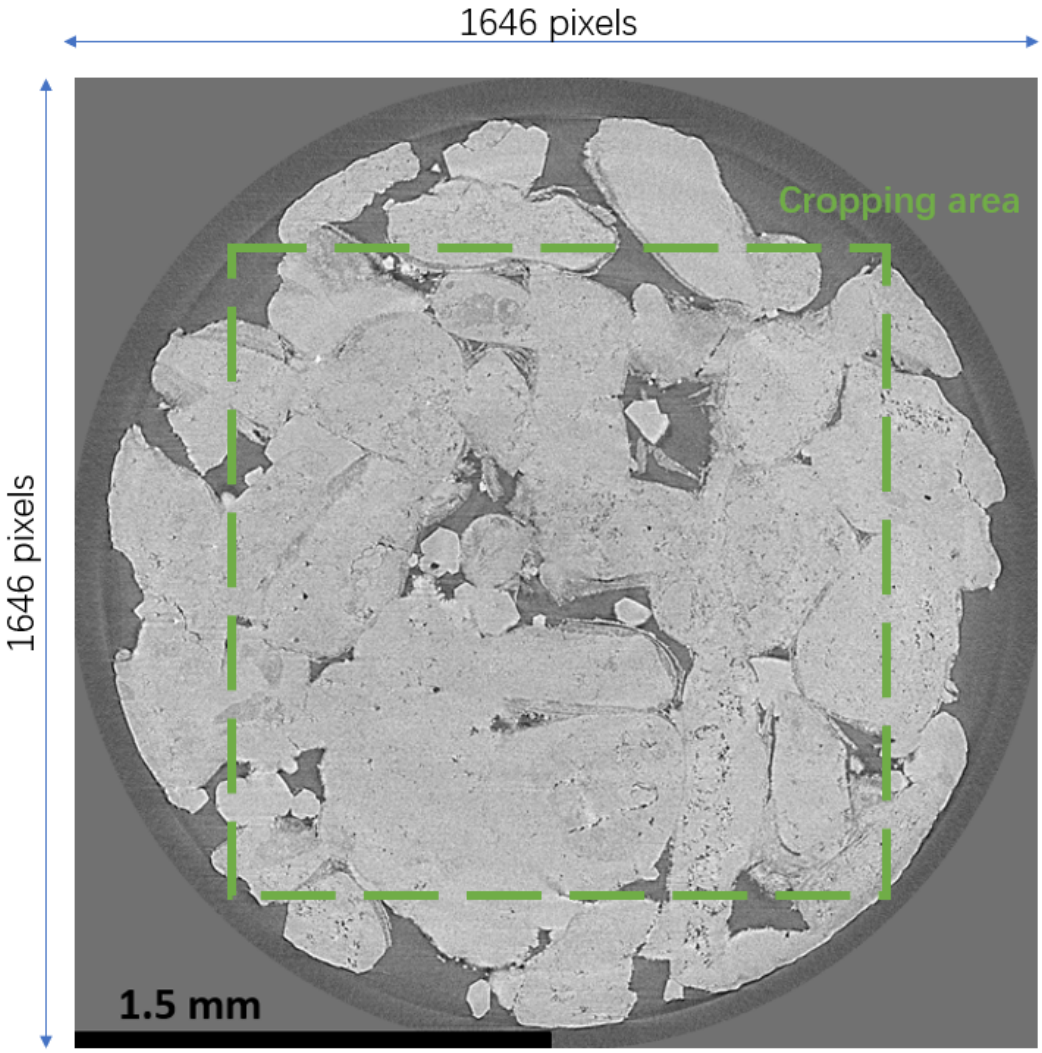


Figure 1. Cross section of a cylinder core of Indiana limestone imaged by synchrotron µCT. The core is fully saturated with H_2O, the lighter grey colour is oolitic limestone grains, the darker grey colour is H_2O in the pore space. The green box shows the cropping area which is the region of interest.

out pre-processing steps such as applying of noise filters. It can reduce processing times 10-fold compared to conventional segmentation methods. We have tested the accuracy of the trained CNN model on a similar dataset with brief retraining, and we also tested the robustness of the model against artificial ring artefacts. We show the model performance to be robust and accurate. We provide both the trained network and a template training code (Python) so users can fine-tune the model to their own data (https://github.com/yclipse/CNN-core-flooding-muCT).

2 Materials and Methods

2.1 Multiphase Fluid Flow Data Acquisition

We used the beam line 2-BM at the Advanced Photon Source in Argonne National Laboratory in Chicago, U.S. A carbonate rock, Indiana limestone, was used as porous medium in the experiments. The rock was cored ($21.2mm$ length $\times$ $3mm$ diameter) and installed in an X-ray transparent cell (Fusseis et al., 2014). For fluid injection we used n-dodecane and potassium iodide solution (2.4M KI as a contrast agent) as oil and aqueous phase respectively. The fluids are immiscible, and the processes involve no chemical reaction between the fluids and the rock. Synchrotron X-ray images were acquired using pink beam by one second acquisition time in every 20 seconds during fluid injections (600 projections per scan, $\frac{1}{600}s$ exposure time per projection, referred to as fast scan). The µCT image resolution is 2.2 micron. A total of 140 time stamps of µCT scan volumes each sized 1004 $\times$ 1646 $\times$ 1646 were acquired during the experimentation (Figure 2).

The 140 µCT scans recorded three individual experimental processes: brine injection (0-40), oil injection (40-80) and simultaneous injection of both fluids (80-140) (Figure 2). Although the physical processes during the experiments are complex and entirely different, the differences shown on the µCT images are simple and repetitive: pores switching between oil-filled, partially-filled and brine-filled states. Therefore, in terms of image segmentation, even a small segment of the full data is highly representative. In this study, we used the first half of the brine injection process to train, validate and test the CNN model, and we used the second half of the oil injection process for extended testing. These two sets of data cover all of the different attributes of the µCT images.

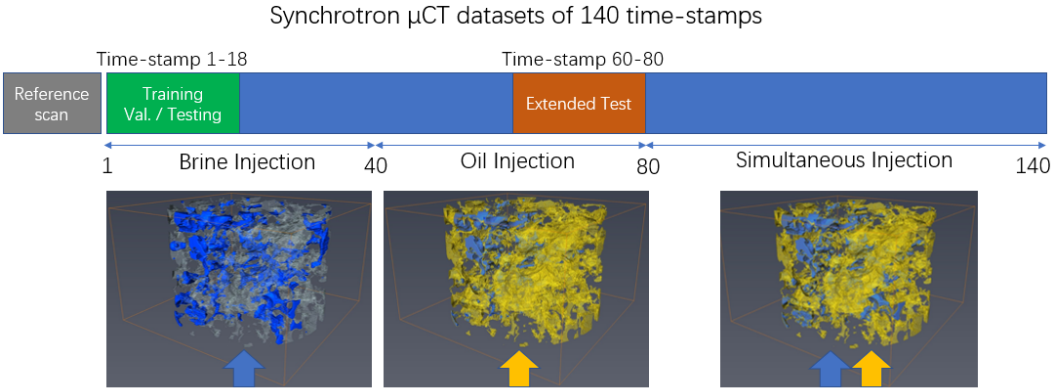


Figure 2. Experimental design. A total of 140 time stamps of µCT scan volumes were acquired during the experiment. The first 40 time stamps of scans recorded the process of brine displacing oil in the Indiana limestone core. The 40th-80th scans recorded the process of oil displacing brine. The last 60 scans recorded the fluid displacement process by injecting oil and brine simultaneously. The first 18 scans were used to train, validate and test the initial model, while 20 more scans from time stamp 60-80 were used to extend the initial model and test it further. An independent reference scan was acquired before the experimentation, which is used to generate a high quality binary mask of the rock.

Before the core-flooding experiment, we imaged the pore structure of the Indiana limestone core with a high-resolution reference scan (Figure 2 Reference Scan). The reference scan had 2.5× more exposure time than the fast scans and each scan involved 1500 projections, therefore the reference scan µCT images are of significantly higher image quality than the images acquired from the fast scans. On the reference scan images (Figure 1) there are fewer artefacts and less noise, the object edges are sharper, and the pore space is fully saturated with a single fluid that contrasts well with the rock. We segmented the high quality reference image into a binary image labelled with rock and pore space using the seeded random walker algorithm (Grady, 2006). The pore space binary image is used as a mask to further separate the two fluid phases (for details of the ground truth segmentation see section 2.2). All processing and computation was done on a HP Z820 workstation. It has 192Gb usable memory and 2 Intel(R) Xeon(R) E5-2640 V3 processors (32 cores). The GPU is an Nvidia™ Quadro K5200 (8Gb, 8 cores).

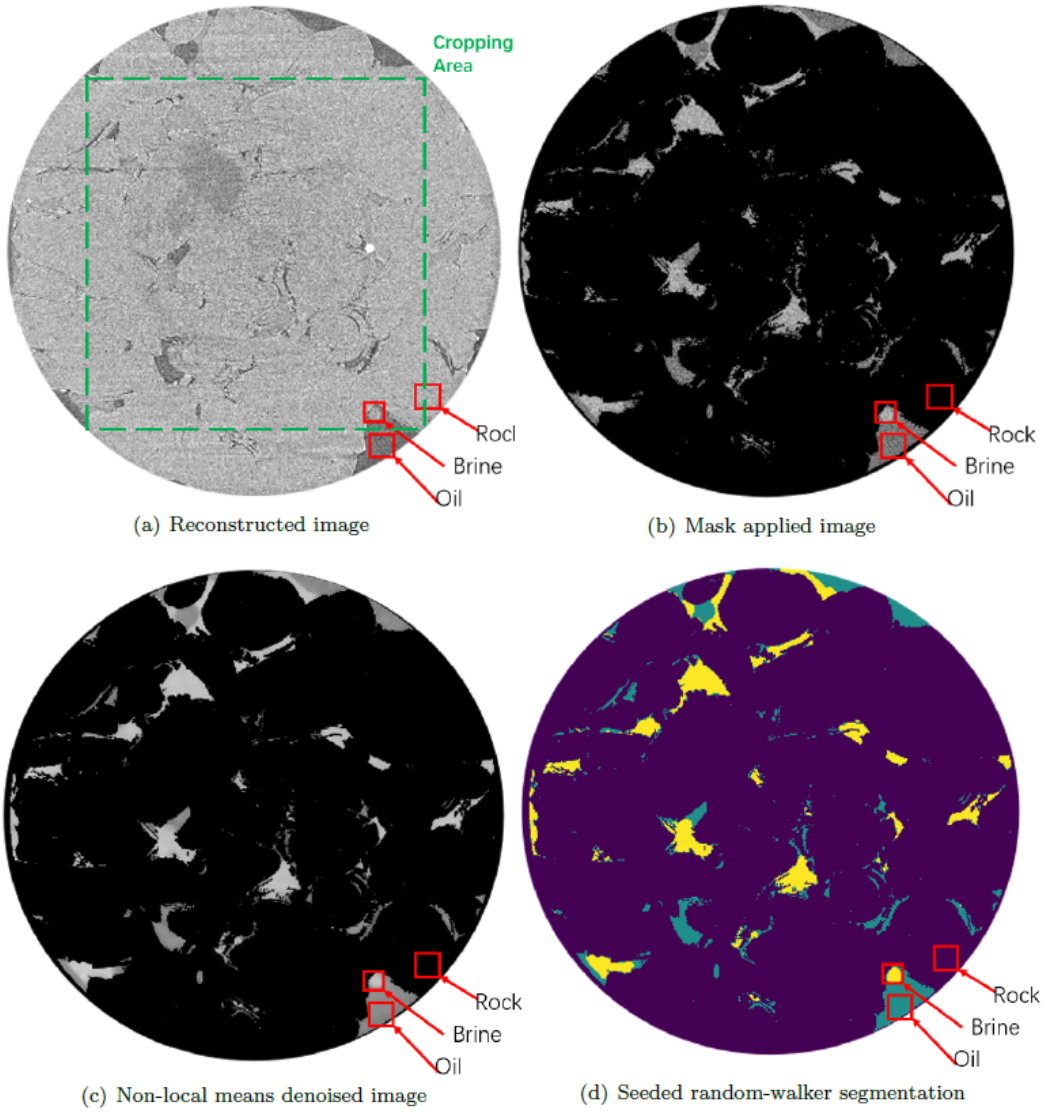


Figure 3. The image segmentation work flow for generating ground truth. **(a)** raw µCT image containing three object classes: oil, brine and rock. The fluid classes are hardly distinguishable because the rock has similar intensity characteristics as the brine. Green box shows the cropped area that was used as network input. **(b)** the raw µCT image with rock masked as black using the rock segmentation acquired from the dry reference scan which makes further segmentation of the two fluid classes possible. **(c)** masked image filtered by the non-local-means algorithm to reduce noise. **(d)** ground truth segmentation produced by seeded random-walker algorithm.

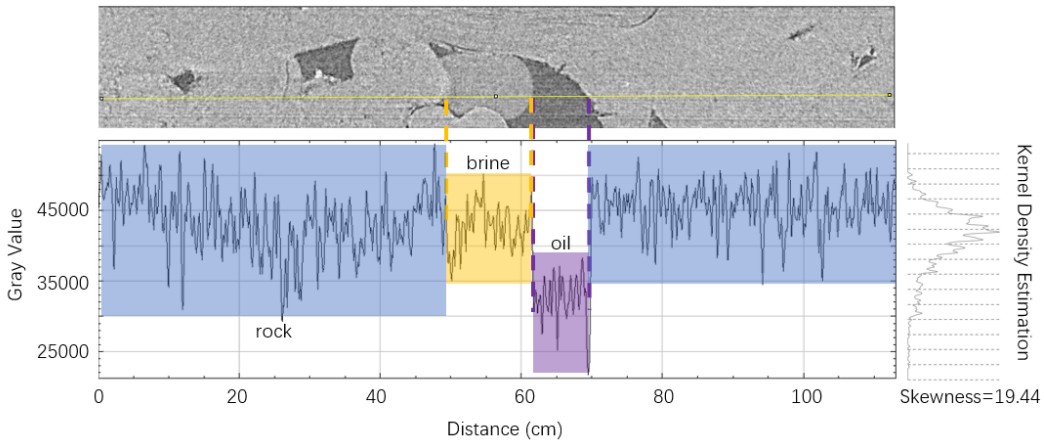


Figure 4. Gray value profile along a line in (yellow) raw reconstructed image. The three different classes,i.e., rock, brine and oil are all included in the profile line. These classes are not separable by their gray values, especially between brine and rock. The range of the grey values of the three classes are highlighted in blue, purple and yellow for rock, oil and brine respectively. The kernel density estimation also shows no sign of distinguishable difference between classes.

2.2 Ground Truth Segmentation

The µCT projections (raw data acquired from fast µCT scans) were reconstructed using filtered back projection (Octopus8.6™(Dierick et al., 2004)). The reconstructed images are referred to as the 'raw µCT images' that are a sequence of cross-sectional image slices of the scanned object normal to the rotation axis. The raw µCT images are the input images for the CNN segmentation. Shown in Figure 3(a), the raw µCT image has three target classes (inside the cylindrical rock core) marked in red boxes as examples: the appearance of oil is dark grey shade, the appearance of brine is light grey shade, and the appearance of rock is also light grey but lighter grey than brine. Figure 4 shows the grey values plotted along a profile line containing all three phases of an example raw reconstructed image. The figure shows that the classes are similar in terms of their gray value distribution, and therefore, they cannot be easily separated by their grey values alone. We acquired the ground truth images by processing the images as follows.

We registered and applied the binary pore space mask using Avizo9™ with the fast scan reconstructed µCT images. Figure 3(b) shows the raw CT images after applying the mask. The rock-brine-oil system now is more distinguishable, but the brine and oil are still very noisy. We used a non-local means filter (Buades et al., 2005) to denoise the

two classes in the masked images (Figure 3(c)). We used the ImageJ implementation of the filter and used a sigma value of 30 and the default window size. This reduces the noise in brine and oil, but at the cost of making the class boundary less pronounced. We implemented the seeded random-walker algorithm (Grady, 2006) to separate brine and oil (Figure 3(d)). Seeded random walker is a region growing method which is capable of inferring ambiguous boundaries. Implementation in Scikit-image was used and took 50.2 minutes to run, on average, on the same size data volume as above.

We chose a total of 18 consecutive µCT scan volumes from the start to the end of brine invasion, these volumes recorded the on-going process of brine displacing oil. These volumes of size $1004 \times 1646 \times 1646$ were segmented in this weakly supervised fashion to generate target annotation for the CNN training dataset.

For the ground truth segmentation, an internal validation of the conservation of pore space (i.e. pore space is fixed over all time stamps) was used to estimate the error in segmentation of the fluid volumes. The experimental procedures ensure that the pore space is filled with two fluids, i.e., either oil or brine. The proportion of the fluids can change through the experiment but the volume of oil and brine together should always sum up to the pore space volume. we measured the total fluid volume of each scan. We also measured the total pore space volume from the reference scan. We assume the average total fluid volume should be equal to the total pore space volume. We found that there is 1.86% of random error, determined from segmenting different fluids in replicate volumes, this is taken as the random error of the ground truth annotation. This amount of error is unlikely to affect any qualitative measurements in 3D such as connectivity. The amount of error also has negligible impact on quantitative measurements such as bulk volume. For such pore-scale images with very irregular and blurry class boundaries, manual segmentation of eighteen thousand images will be extremely time-consuming, and may not guarantee a reduction in error. In addition to determining the error based on volumetric measurements, we visually compared the results of the ground-truth segmentation with greyscale image data and found the differences in the fine details of pore structure to be insignificant.

This segmentation workflow is not scalable partly because of the high computational requirements, but also because segmentation parameters cannot be applied across different batches of scans. We acquired a total of 140 µCT scan volumes during the ex-

periment, and used only the first 18 µCT scan volumes (containing 18,072 images) to train the CNN model. This means that 90% of the data were processed using the more efficient CNN model.

2.3 Convolutional Neural Network Segmentation

2.3.1 Training Methods

Before training, we prepared the training dataset. Every raw reconstructed image was paired with the corresponding ground truth segmentation image to make a set of input-target image pairs to train the network. A total of 18 µCT scan volumes were randomly divided into 14 training sets, 2 test sets and 2 validation sets (Figure 2 Training, Val and Test). Each µCT scan volume has 1004 reconstructed slices. The raw reconstructed images (1646×1646 pixels) were down-sampled (2×2 mean pooling) to size 823×823 pixels. A rectangular area of size 496×496 pixels was cropped as the training area (shown as the green box in Figure 3). The down-sampling of the original images was: 1) to fit the images into the GPU memory. 2) To exclude pores adjacent to the perimeter which do not contain useful information because the perimeter has different wettability than the rock (for a solid material, wettability indicates the tendency of one particular fluid phase to spread over the solid surface in presence of another fluid). The choice of the particular size 496×496 was based on the architecture of the CNN network, which has four max-pooling layers which each will divide the input size by two, so the input size needs to be four times divisible by 2.

We used the CNN architecture U-Net introduced by Ronneberger et al. (2015) and implemented by van Vugt (2017) using the open-source deep learning platform Pytorch (Paszke et al., 2017). The network comprises a contracting path (Figure 5 left box) and an expanding path (Figure 5 right box). The contracting path, like conventional CNN, uses 3×3 convolution, followed by a rectified linear unit (ReLU), and then 2×2 max-pooling for down-sampling. The expanding path up-samples the feature map with 2×2 up-convolution followed by 3×3 transposed convolution with ReLU. A concatenation of the feature map of the corresponding contracting path is applied to the up-convolved feature map (Figure 5 grey arrow). In total the network has 23 convolutional layers.

Figure 6 shows the overall workflow of training, validating and applying the CNN model. In training, the model takes a training image as its input and produces a pre-

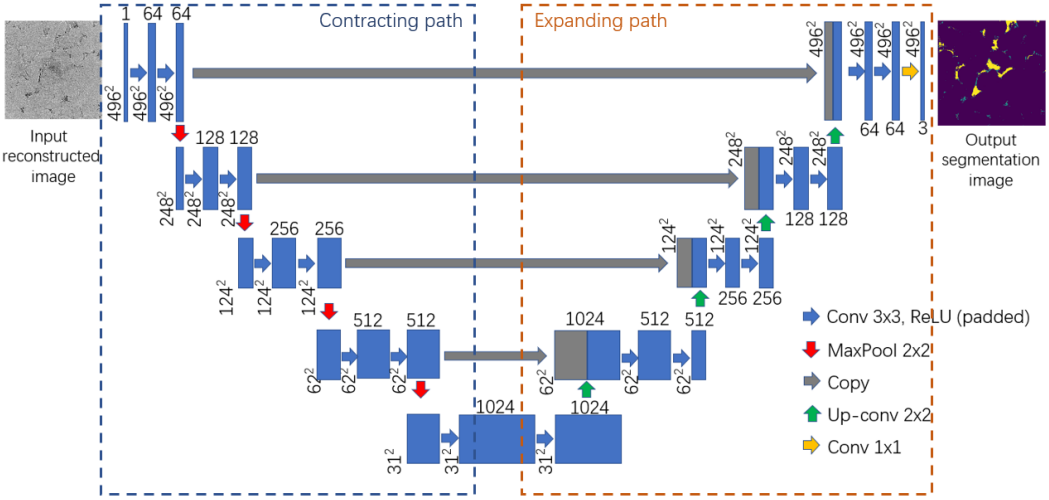


Figure 5. Modified from Ronneberger et al. (2015), this figure shows the architecture of the U-Net. It consists of a contracting path that extracts different levels of features, and an expanding path that up-convolves the image to produce the final segmentation. The two paths form a U-shaped network. The original paper uses VALID padding (i.e. no padding), so the height and width of each feature map decreases after each convolution. In this implementation SAME padding (i.e. zero padding by 1 on each side) is used so the height and width of the feature map will stay the same (e.g. Silburt et al., 2019).

diction of the probability of each pixel belonging to a class (i.e. rock, oil and brine). Then the prediction is compared with the ground truth image to yield a training loss value which quantifies the difference between the prediction and ground truth. The model iteratively updates the internal adjustable parameters (i.e. weights) as training proceeds, to minimize the loss and, thus, to improve the prediction in a step-wise manner. We trained the model for 34 epochs, where a training epoch is a full traverse of all training data.

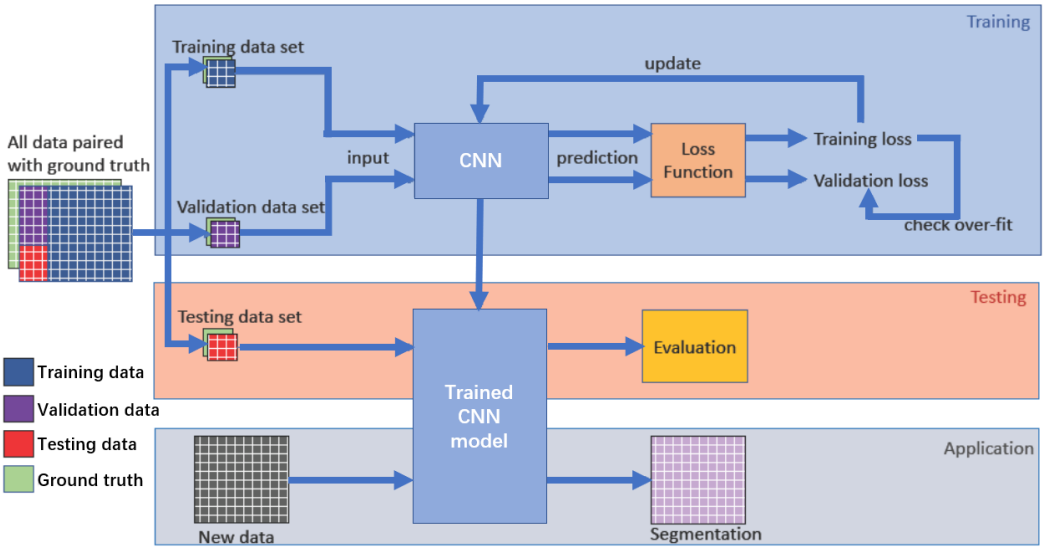


Figure 6. CNN training, testing and application work flow. The training process uses training datasets and validation datasets to adjust the best fit of a CNN model representing the relation between input images and ground-truth images. The testing process verify the model with a held-out dataset. The application process uses the trained CNN model to segment new µCT data.

Cross-entropy loss was used as the loss function during training. The loss function calculates the distance between the two probability distributions, i.e., the output prediction and the corresponding ground-truth. Cross-entropy decreases when the output and ground-truth have a higher resemblance. We used the Pytorch built-in cross-entropy loss function which is written as:

$$\text{Loss} = -\sum_{i \in I} \sum_{c \in C} y_{i,c} \log \hat{y}_{i,c} \tag{1}$$

where $y = (y_1, y_2, \ldots, y_n)$ and $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$ are the ground truth values and the network prediction values of all pixels of a training image flattened to 1D arrays. C is

the set of all classes and I is the set of all pixels. $y_{i,c}$ and $\hat{y}_{i,c}$ are values of the ith pixel in class c.

Hyper-parameters are user-defined parameters that control the overall learning behaviours of the network during training. Learning rate is one dimensionless hyper-parameter that defines the size of steps to adjust the network weights to minimize loss; a higher learning rate implies a larger adjustment in each iteration. Multiple learning rates were tested and we found the value of $5{\times}10^{-5}$ to be optimal for loss to converge quickly and smoothly. During training, we divided the training datasets into mini batches each containing two images. This can improve the efficiency of training. An ℓ_2 regularization parameter of 1×10^{-5} was applied to alleviate over-fitting (ℓ_2 regularization is also know as Ridge regression, it adds squared magnitude of coefficient as penalty term to the loss function to avoid over-fitting). The sequence of the input data was shuffled at the start of each epoch. This can eliminate bias that is produced by the high similarity of two adjacent µCT slices. The optimizer used to train was the Adam algorithm (Kingma & Ba, 2014) which is built into Pytorch.

2.3.2 Validation

At the end of each training epoch, a validation loss value is computed on the validation set (Figure 6 Training). The validation loss is not back-propagated and the network weights are not updated. Thus, the model occasionally 'sees' this data but never learns from it, and thus the process of validation provides an unbiased, on-going evaluation of a model fit to the training dataset. The validation loss value is plotted throughout training and compared with the training loss value. The model is best-trained at the epoch when the validation loss starts to increase, or the validation accuracy ceases to decrease. After this the model starts to overfit.

For the pixels in the network prediction of an input image, a pixel is assigned to a class if the corresponding probability is the highest among the three classes. There is a chance for a rare exception where a pixel has exactly equal probability for all three classes (33% oil, 33% brine, 33% rock), or equal probability for any two classes (e.g. 50% brine and 50% rock). These pixels are assigned to the rock class for the reason that such pixels are often on the contact surface between the fluids and the rock.

For each class of a prediction, we calculated the number of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) pixels with respect to the ground truth. For example, for the oil class in a CNN segmented image, true positive pixels are the pixels which are assigned to oil, and the corresponding ground truth for the same pixels are also oil. False positive pixels are the pixels which are assigned to oil, but the corresponding ground truth for the same pixels are not oil. True negative pixels are the pixels which not assigned to oil, and which agree with the ground truth. False negative pixels are the pixels which are not assigned to oil, but the ground truth for these pixels is oil.

We used three metrics to measure the segmentation performance, they are the Rand index (accuracy, (Rand, 1971)), IoU (intersection over union) and AUC-ROC (area-under-curve of the receiver operating characteristic (Bradley, 1997)). The Rand index is the ratio of correctly classified pixels to total pixels. The Rand index was calculated using Eq. (2)

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

The intersection over union measures how well the prediction is spatially overlapped with the ground truth. It is the ratio between two values: the area of overlap and the area of union (Eq. (3)).

$$\text{IoU} = \frac{\text{Area of intersection}}{\text{Area of union}} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \tag{3}$$

The AUC-ROC is a common metric for assessing the performance of a classifier. ROC is a curve plotted in the coordinate system with true positive rate ($\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$) against false positive rate ($\text{FPR} = \frac{\text{FP}}{\text{FP+TN}}$), over various thresholds. An ROC curve essentially represents the trade-off between sensitivity and specificity of a classification model. AUC is the area under the ROC curve which measures the model's separability of different classes. The AUC-ROC value varies between 0.5 and 1. When AUC equals 1, it represents a perfect classification model that classifies all samples accurately. When the AUC value is around 0.5, it represents the case where all classifications are random due to a completely overlapped predictive probability distribution of true and false values.

After training, the model is tested on 'unseen' data to check its generalizability and assess the segmentation quality. The separate test dataset was only used once the model had been trained (Figure 6 Testing). The test dataset provides a standard to evaluate the model.

3 Results

3.1 Training Result

Figure 7 shows the training process by plotting the validation accuracy and loss over training epochs. The top graph shows that as training progresses the validation accuracy increases before finally stabilizing for all three classes. The bottom graph shows that the training loss drops sharply at the outset of training but became stable as the training progresses while the validation error starts at a relatively low value and decreases slowly. Theoretically it is assumed that a validation loss curve is U-shaped since it first drops as the algorithm learns to generalize, then stabilizes, and finally starts to rise as the model tends to overfit. Therefore the criterion to stop training is set as a continual increase of validation loss over 5 consecutive epochs. The network was trained for 34 epochs and the best fit was identified at the 29th epoch. The total training time was 205.4 hours.

The segmentation by the model during training and after training are shown in Figure 8. The first row shows the raw reconstructed image and the ground truth. The second row shows the segmentation of the same image during the 11th epoch of training and after 29 epochs of full training. This result illustrates an improvement of the model during training as at the 11th epoch the segmentation result is sub-standard since it only partially captures the pore space structure and the boundary decision is not well defined. However, at the 29th epoch the segmentation result is very similar to the ground truth.

3.2 Testing Result

The model trained for 29 epochs was identified as the best segmentation model and tested on the full testing dataset. The performance measured on the test set shows high accuracy for all three classes. The test result shows that the accuracy of rock, brine, oil and average accuracy over all three classes are 98.5%, 99.3%, 99.1% and 99.0% respectively. The AUC-ROC score for brine, rock and oil are 0.99, 0.99 and 0.99 respectively. It shows very high separability of the model. Table 1 shows that the CNN segmentation

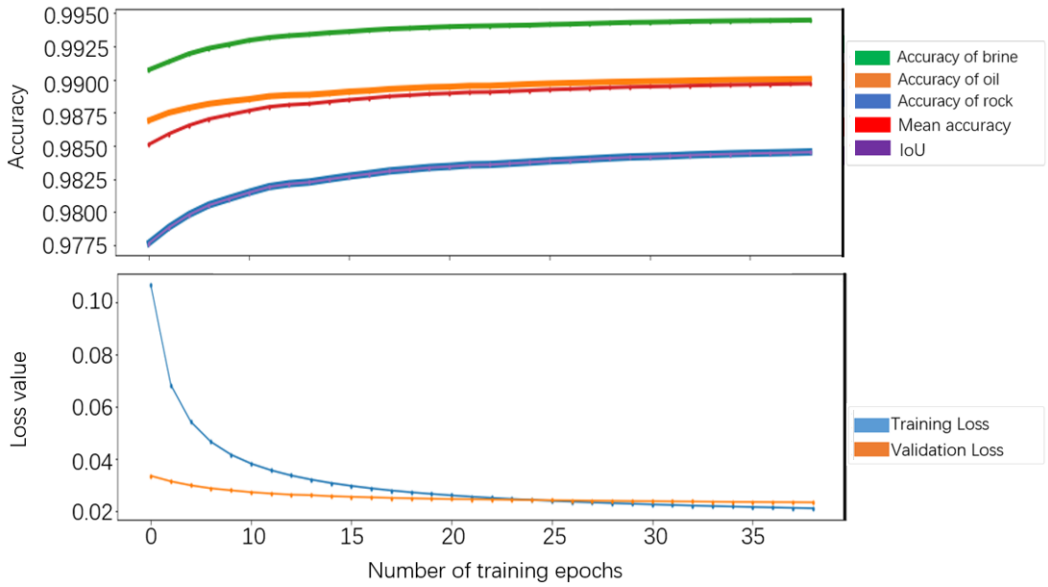


Figure 7. **Top:** The plots shows the variation of accuracy and IoU score over training epochs for three classes separately. **Bottom:** the image shows the variation of training loss and validation loss over training epochs. The IoU curve completely overlaps with the curve of rock accuracy.

result correctly labelled 93.6% oil phase as oil, 96.3% brine as brine, and 99.5% rock as rock. The total oil phase and brine phase were slightly underestimated with regard to the ground truth. We measured an IoU score of 0.98 showing that the CNN segmentation is spatially overlapping with the ground truth correctly.

The probability map (Figure 9) visualises the probability distribution of all three classes. It is generated by the softmax function that transforms the model output into a probability distribution such that the probability of three classes at the same pixel add up to 1. The colour map indicates that the likelihood of a pixel belonging to one particular phase, with red implying extremely likely and blue implying extremely unlikely while white implying ambiguous likelihood. The three classes are well separated since the ambiguous likelihood pixels are of minor amounts and are mostly located at the boundaries of two phases. It also illustrates a very robust performance of the model in segmentation.

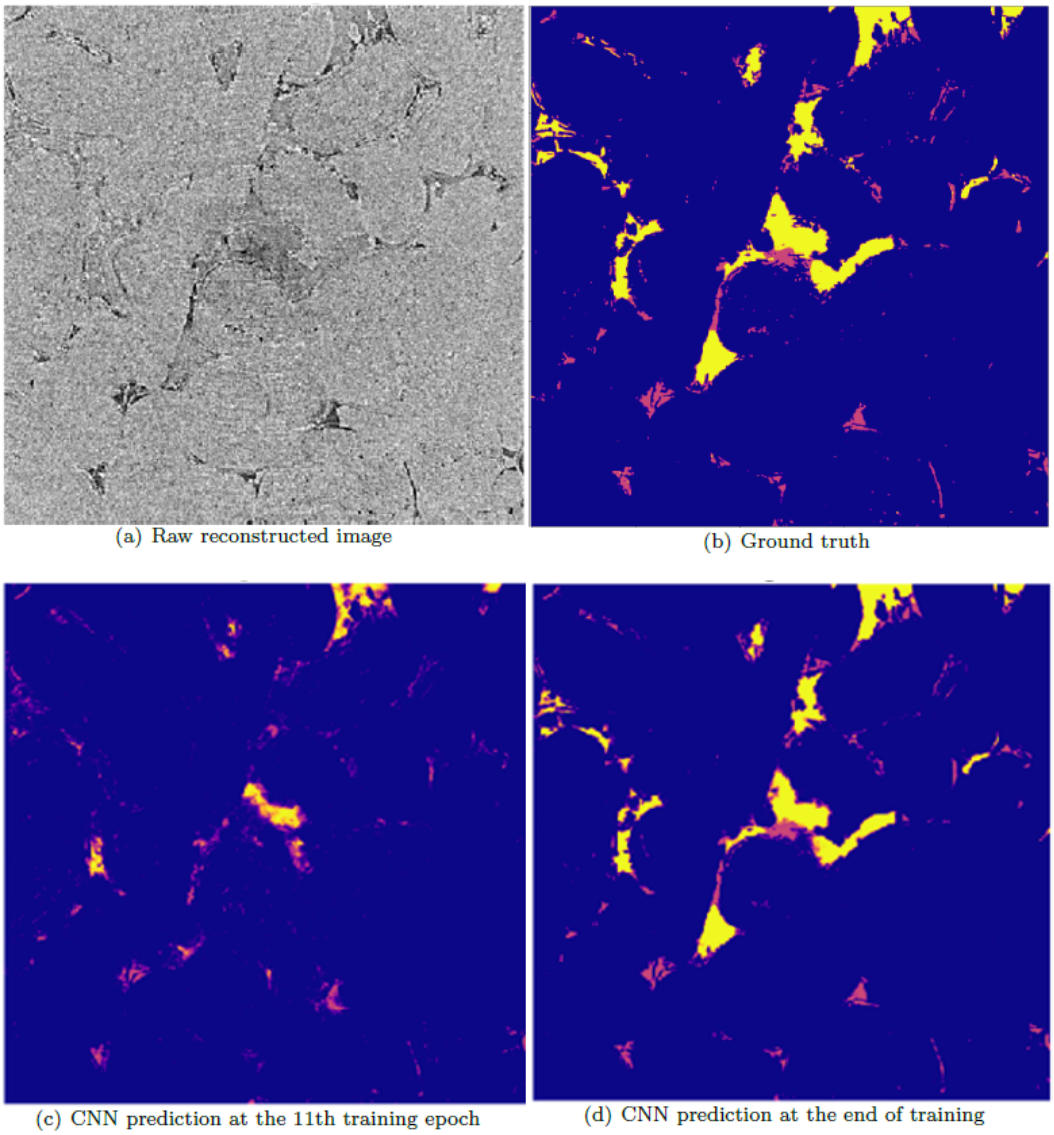


Figure 8. CNN prediction during training and after training. **(a)** Raw reconstructed image before segmentation. **(b)** Ground truth image. **(c)** CNN prediction of 3-phase probability at the 11th training epoch. **(d)** CNN probability prediction after 29 epochs of training. The segmentation was on the same image from the testing dataset. This illustrates the improvement of segmentation quality during training.

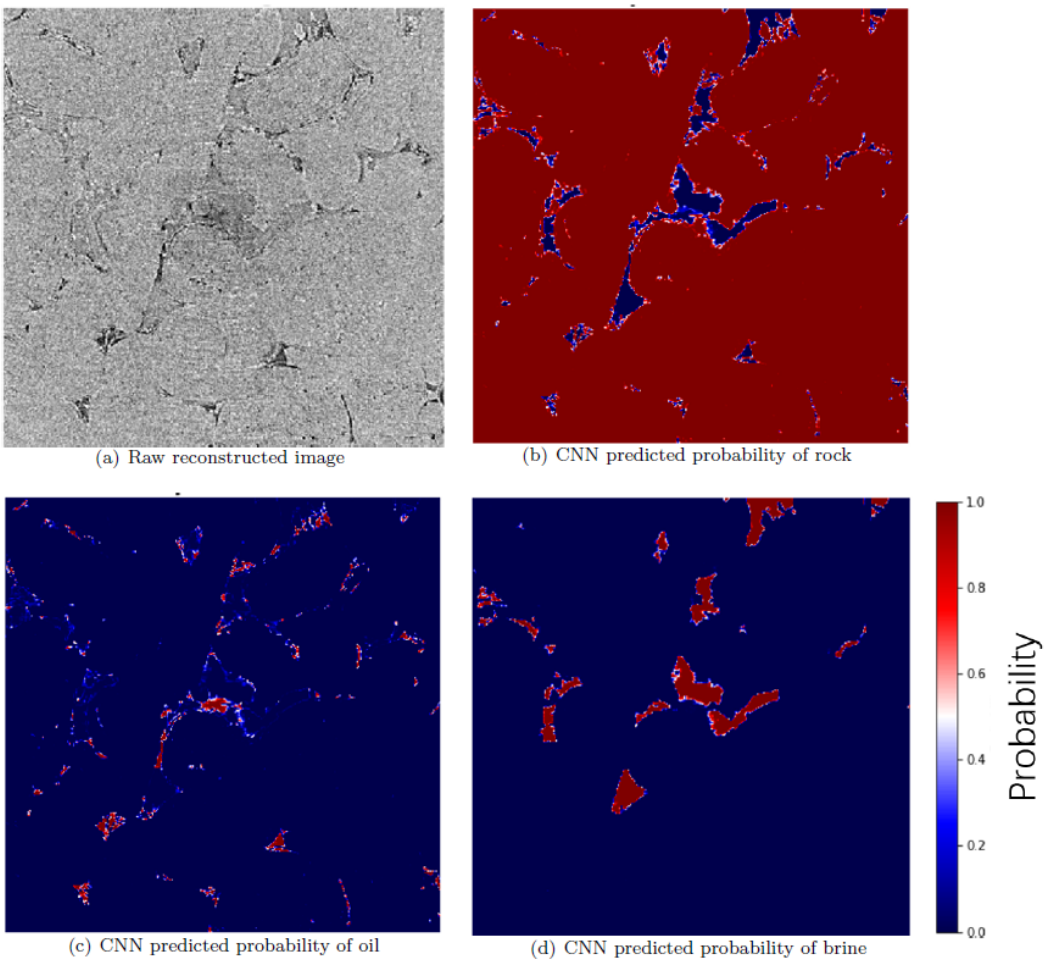


Figure 9. Probability distribution of each phase at the network output. The color bar shows probability of a pixel belonging to that phase. This illustrates that the majority of segmentation was certain (probability close to either 0 or 1) with very few uncertain pixels (probability equals 0.5). The uncertain pixels were classified into the rock class as they are always presented between the contact between the rock and two fluids.

Table 1. Contingency table of each class segmented by the CNN model compared with the ground truth.

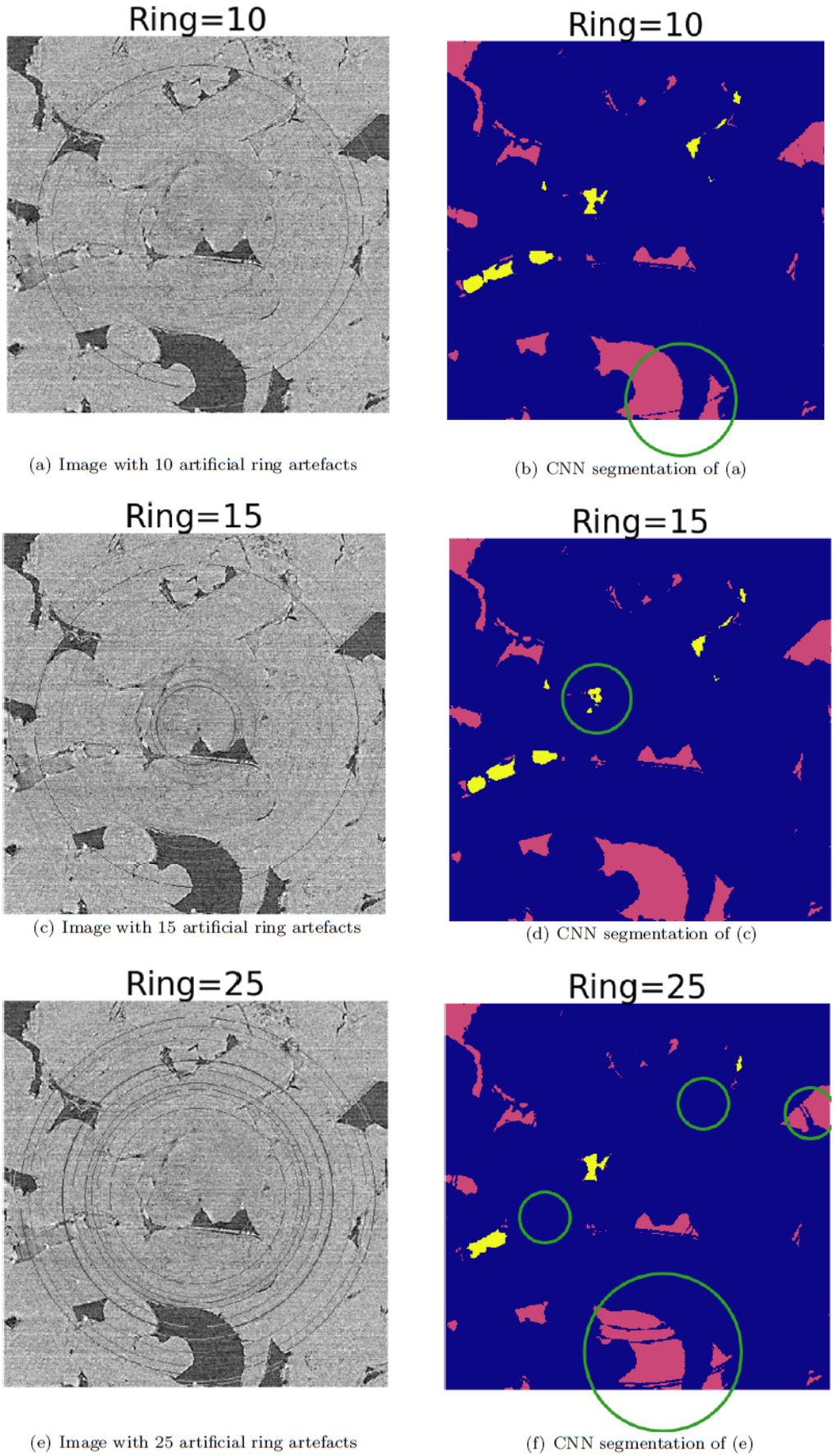
		Ground truth		
		Oil	Brine	Rock
Estimation	Oil	0.93	0.02	0.06
	Brine	0.04	0.96	0.03
	Rock	0.04	0.01	0.99

4 Discussion

4.1 Segmentation Robustness

Ring artefacts are concentric rings caused by mis-calibration or failure of a detector element. They are one of the major and most common type of artefacts that hinders µCT image segmentation. We added artificial ring artefacts to an image selected from the test set (scan 17) to examine the robustness of the model to image quality degradation due to ring artefacts. The chosen test image is visually separable and does not suffer from original µCT ring artefacts. Artificial ring artefacts were generated by adding concentric circles with random radii and intensities at a fixed coordinate on the image. We measured the severity of this artefact in terms of the number of rings present (Figure 10). The direct impact of the artificial added ring artefacts on the image quality is statistically shown in Figure 11: compared with the histogram curve of the original image (blue), the ring artefacts caused spikes on a histogram curve of the same distribution (orange).

Figure 12 shows the variation of AUC-ROC score and mean accuracy over degree of severity. The CNN segmentation performed surprisingly well for artificial ring artefacts. Only the brine phase of the severely affected images shows some minor mis-classification. It is also surprising because the training images do not have any significant ring artefacts. This test provides an insight on the insensitivity of the CNN segmentation with ring artefacts, which is one of the most intrusive feature for most conventional segmentation methods.



(a) Image with 10 artificial ring artefacts

(b) CNN segmentation of (a)

(c) Image with 15 artificial ring artefacts

(d) CNN segmentation of (c)

(e) Image with 25 artificial ring artefacts

(f) CNN segmentation of (e)

Figure 10. CNN segmentation model tested on artificial ring artefact. **Left:** Same µCT image with increasing severity of artificial ring artefacts where the severity is measured in terms of the number of rings added. **Right:** Corresponding CNN segmentation results where major mis-classifications are marked by green circles.

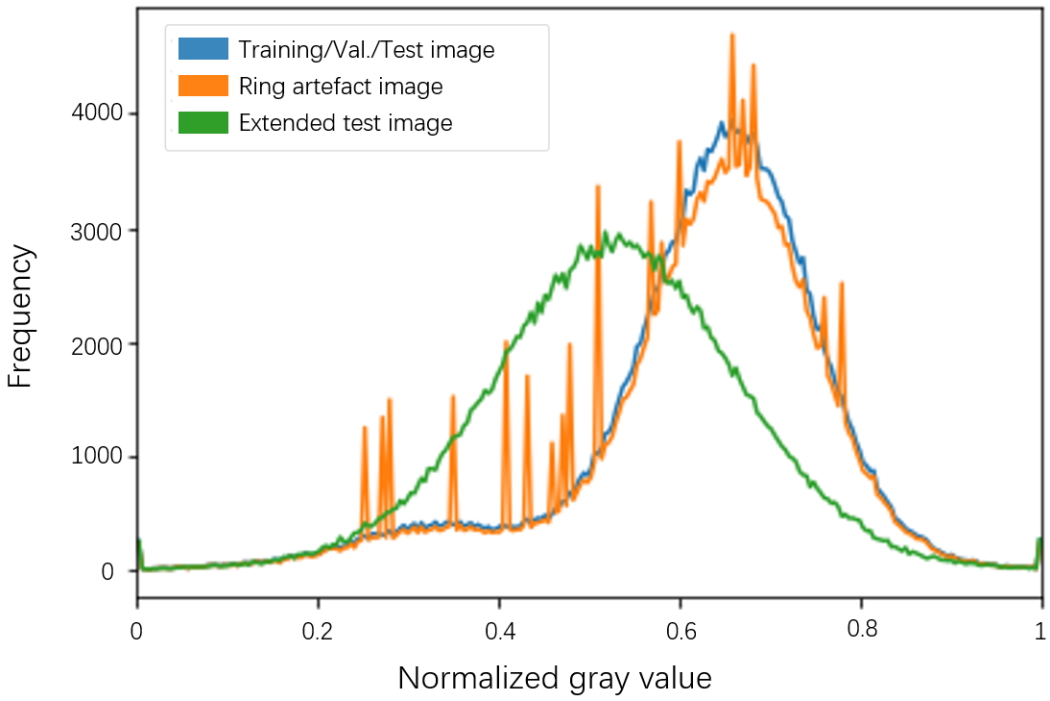


Figure 11. Image histogram of the three datasets segmented by the CNN model. All three histograms show unimodal distribution with the maximum indicating the rock class. The ring artefact image histogram shows the same distribution with the training/validation/testing image. The extended test image histogram shows a shifted distribution which is due to different µCT reconstruction method and parameters.

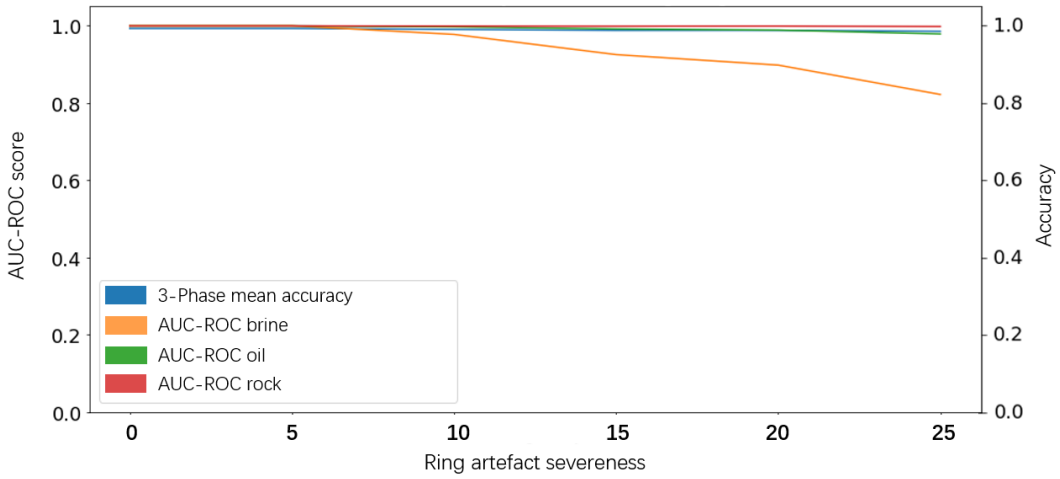


Figure 12. The variation of accuracy and AUC-ROC of CNN segmentation over varying degree of ring artefact. The segmentation quality is stable with a minor decrease of performance at heavily degraded images.

4.2 Extended Test on Similar Data With Extra Training

To exclude data leakage that may occur when using highly similar training and testing data, we also tested the trained model on 20 µCT scan volumes which belong to the same experiment as the training dataset but from later timestamps with the same rock and fluids (Figure 2 Extended Test, 60th-80th time stamps). This dataset is of the same size as the training dataset, and uses the same approach for determining the ground-truth. This dataset has more noise compared to the training data set, and has horizontal stripe artefacts which the training dataset does not have.

The difference of image quality is because this dataset was reconstructed using a different reconstruction tool, TomoPy (Gürsoy et al., 2014) and, the reconstructed image appearance has a small, but systematic variation due to the different reconstruction parameters. The systematic variation is shown in the green curve on Figure 11: images from the extended dataset have a gray value distribution the maximum of which is shifted to the left compared with other datasets. Therefore, the trained model might not work accurately on these data right away (Figure 13(c)). To mitigate this, the model was briefly trained for five epochs using only the first volume of the 20, and tested with the remaining 19 volumes.

Figure 13 shows the CNN segmentation result by directly using the trained model did not capture most of the pore structures and did not separate the two fluids (Figure 13(c)). After five epochs of additional training, the CNN segmentation converged towards the ground truth image (Figure 13(d)). The extra test results are: accuracy of rock is 93.5%, accuracy of oil is 95.2%, accuracy of brine is 99.1%, and the mean accuracy is 95.9%. The average IoU of the three classes is 0.98. The performance remains good with only a modest decrease in accuracy. This confirms that the CNN model can quickly be adapted to handle similar datasets with brief re-training.

We tested whether the prior training was essential, or whether 5 epochs alone were sufficient. The result (Figure 13(e)) shows that the result is inconsistent with the ground truth and confirms that the prior training is necessary and the CNN requires it in order to effectively adapt to new data, but needing relatively few epochs of additional training to achieve that adaptation.

To address the impact of CNN segmentation and conventional segmentation on experimental measurements of the fluids, we compared the fluid saturation measured from CNN segmentation results and conventional segmentation results. Fluid saturation measures how much of a fluid is present in the pore space of a rock. In this comparison, we expect the fluid saturation measured from both CNN segmentation and random-walker segmentation to give similar results.

Figure 14 shows the saturation of oil measured from both CNN segmentation and conventional segmentation are very close. The plot shows that measurements taken from the random-walker segmentation is overall higher than the measurements taken from the CNN segmentation. A one-sided Wilcoxon test was performed with the two saturation measurements to examine if the measurement taken from random-walker is statistically larger. The Wilcoxon test returned a p-value of less than 0.002 and confirms that the difference is statistically significant. The median of the Wilcoxon test values is 2, indicating, however, that the amount amount of difference is small. The CNN segmentation model is able to produce reliable segmentation that is close to the result obtained using conventional segmentation method. The CNN model has the ability to adapt to variations in the dataset with minimal retraining.

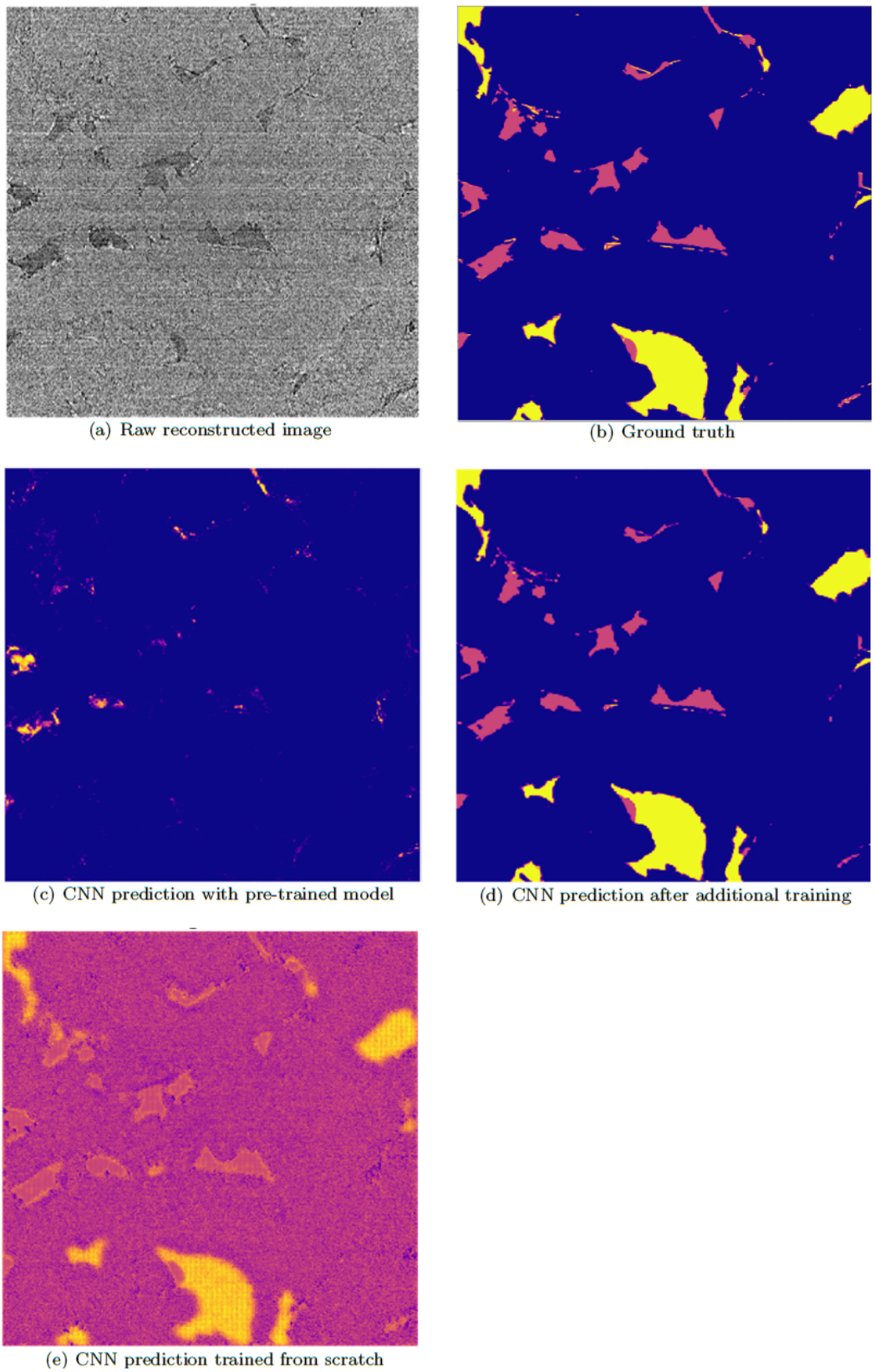


Figure 13. Segmentation result of the model before and after extra training. **(a)** Raw reconstructed image before segmentation. **(b)** Ground truth image. **(c)** CNN prediction of 3-class probability using the trained network without additional training. **(d)** CNN segmentation result after 5 epochs of additional training with one µCT scan volume. **(e)** CNN prediction trained from scratch. This illustrates that the CNN model can be adapted to similar dataset with brief re-training.

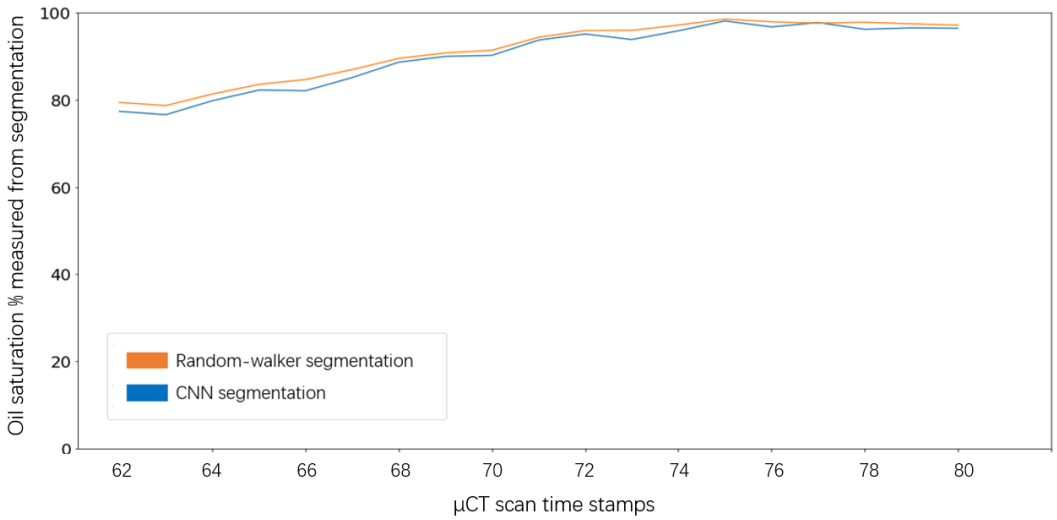


Figure 14. Comparison of oil saturation measured from random-walker and CNN segmentation. Oil saturation of consecutive 19 scans from timestamp 60 to 80 was measured from both CNN segmentation and conventional random-walker segmentation. The measurements of saturation from the segmentation using two methods are highly similar.

4.2.1 Impact of Different Image Quality Disturbance

Comparing the CNN model performance on the extended test and the ring artefact test, the CNN model can perform reliably with the impact of low to moderate degree of ring artefacts without additional training. But it needs a small amount of additional training to perform well on the extended test. Figure 11 shows that the disturbance in the ring artefact images are local spikes on the histogram, while the disturbance in the extended test images is a systematic shift of the image histogram. The model has better resistance to local disturbance produced by ring artefacts than to a systematic change of the image histogram. The results indicate that 1) the CNN segmentation can be applied to datasets that are affected by low to moderate degrees of ring artefacts without additional training, and 2) the CNN segmentation needs minor additional training to perform well on datasets that have a systematic shift of the histogram. It also suggests that for heterogeneous datasets, the training data set should be sampled across the entire range to decrease bias and reduce additional training. However it may require a greater amount of training to generalize the model.

4.2.2 Temporal Efficiency

For a dataset of size 1004×496×496, the CNN model took 5 minutes 40 seconds to segment the dataset. The conventional segmentation pipeline using the random-walker algorithm took 50 minutes 13 seconds to segment the same dataset. The comparison shows a ten-fold speed advantage using the CNN method. This comparison was done on a CPU-dedicated workstation which has 32 CPU cores, without which the processing time of the conventional segmentation can be up to 15 hours and 32 minutes if using a single thread. The GPU used in this study was released in 2014. The GPU has a compute capability of 3.5, while current GPUs have a compute capability of 8.6, meaning the micro-architecture is five-generations behind and lack of many powerful computational features. Although it is difficult to give a precise number of improvement, we estimate a further improvement in temporal efficiency at least ten times of magnitude if the latest GPU is used.

4.3 Future Improvements

The current training dataset can be augmented by introducing noise, mirroring, rotating, scaling, cropping or translating the original training images. Data augmentation allows amplification of the training dataset based on existing dataset and therefore can further generalize the CNN model to achieve a more robust segmentation of different datasets. This data augmentation strategy was tested with the U-Net architecture and was found to produced excellent segmentation (Ronneberger et al., 2015).

As a working CNN model of segmenting µCT data of core-flooding experiments, the current model learned the segmentation process, rather than applying an intrinsic knowledge of 'what is rock' and 'what are fluids'. The reason is that the training data itself is highly biased towards the particular rock and fluid type of this experiment. Different rocks have different internal structures. The appearance of a rock on µCT is highly dependent on the imaging settings, and even the same rock can look different under different conditions such as X-ray energy and sources. Different fluids can vary in µCT images too. It is beneficial to keep training on future experiments to generalise the model with more kinds of rocks and fluids. As more type of imaging conditions, noises, lithologies and fluid phases are collected in the training data, the model will be increasingly generalized. The future training needed will eventually decrease as the CNN network has

'seen' more rocks, fluids and experiments. It becomes a more mature model that can segment multiphase fluid flow experiment data regardless of noise, rock type, fluid type and beam line condition.

Comparison of different network architectures can be further tested. Apart from the U-Net, there are other powerful CNN architectures such as ResNet (He et al., 2016), GoogLeNet (Szegedy et al., 2015), VGGNet (Simonyan & Zisserman, 2014) etc. with different advantages and specialities. The approach introduced in this paper can be implemented on different CNN architectures for a variety of segmentation demands.

5 Conclusion

Use of a convolutional neural network can significantly improve the segmentation workflow for multiphase flow µCT images. The advantages are five-fold. First, once a network for segmenting multiphase flow images is trained, it can be applied to future data without retraining or only with fine-tuning. Second, the segmentation is directly performed on the reconstructed image, and so considerable time spent on the pre-processing (tuning of filtering, registration, masking etc.) can be avoided. This is significant because faster segmentation means that more time can be devoted to analysis and interpretation of data. Third, the CNN method can segment a dataset of size $1004 \times 496 \times 496$ in 5 minutes, which is ten-fold faster than a conventional segmentation pipeline using the random-walker method. The speed advantage is obvious even on our CPU-dedicated workstation, and the speed of CNN may be significantly increased by using a more recent GPU. Fourth, the algorithm is capable of segmenting images that are highly affected by ring artefacts, which often require additional correction or removal steps for conventional processing paths (e.g. Rashid et al., 2012; Brun et al., 2009). Finally, the performance of the CNN network improves as more data is available through re-training. With little re-training it can be easily adapted to new datasets when the previous training is biased. Overall the CNN segmentation is a powerful and efficient tool for µCT image segmentation, especially for large datasets.

Acknowledgments

We acknowledge the support of Petrobras and Royal Dutch Shell for financial support of the International Centre for Carbonate Reservoirs (ICCR). This study comes from the SatuTrack II sub-project of ICCR.

Data Availability Statement

The data that support the findings of this study are available from Petrobras and Shell. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the authors with the permission of Petrobras and Shell.

References

Alqahtani, N., Alzubaidi, F., Armstrong, R. T., Swietojanski, P., & Mostaghimi, P. (2020). Machine learning for predicting properties of porous media from 2D X-ray images. *Journal of Petroleum Science and Engineering*, *184*, 106514.

AlRatrout, A., Blunt, M. J., & Bijeljic, B. (2018). Wettability in complex porous materials, the mixed-wet state, and its relationship to surface roughness. *Proceedings of the National Academy of Sciences*, *115*(36), 8901–8906.

Berg, S., Armstrong, R., Ott, H., Georgiadis, A., Klapp, S. A., Schwing, A., ... Leu, L. (2014). Multiphase flow in porous rock imaged under dynamic flow conditions with fast X-ray computed microtomography. *Petrophysics*, *55*(04), 304–312.

Berg, S., Ott, H., Klapp, S. A., Schwing, A., Neiteler, R., Brussee, N., ... Schwarz, J.-O. (2013). Real-time 3D imaging of haines jumps in porous media flow. *Proceedings of the National Academy of Sciences*, *110*(10), 3755–3759.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, *30*(7), 1145–1159.

Brun, F., Kourousias, G., Dreossi, D., & Mancini, L. (2009). An improved method for ring artifacts removing in reconstructed tomographic images. , 926–929.

Buades, A., Coll, B., & Morel, J.-M. (2005). A non-local algorithm for image denoising. , *2*, 60–65.

Dierick, M., Masschaele, B., & Van Hoorebeke, L. (2004). Octopus, a fast and user-friendly tomographic reconstruction package developed in labview®. *Measurement Science and Technology*, *15*(7), 1366.

Fusseis, F., Steeb, H., Xiao, X., Zhu, W.-l., Butler, I. B., Elphick, S., & Mäder, U. (2014). A low-cost X-ray-transparent experimental cell for synchrotron-based X-ray microtomography studies under geological reservoir conditions. *Journal of synchrotron radiation*, *21*(1), 251–253.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies

for accurate object detection and semantic segmentation. , 580–587.

Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*(11), 1768–1783.

Gürsoy, D., De Carlo, F., Xiao, X., & Jacobsen, C. (2014). TomoPy: a framework for the analysis of synchrotron tomographic data. *Journal of synchrotron radiation*, *21*(5), 1188–1193.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. , 770–778.

Kanin, E., Osiptsov, A., Vainshtein, A., & Burnaev, E. (2019). A predictive model for steady-state multiphase pipe flow: Machine learning on lab data. *Journal of Petroleum Science and Engineering*, *180*, 727–746.

Karimpouli, S., & Tahmasebi, P. (2019). Segmentation of digital rock images using deep convolutional autoencoder networks. *Computers & geosciences*, *126*, 142–150.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. , 1097–1105.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. , 3431–3440.

Mo, S., Zhu, Y., Zabaras, N., Shi, X., & Wu, J. (2019). Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. *Water Resources Research*, *55*(1), 703–728.

Neubert, P., & Protzel, P. (2014). Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. , 996–1001.

Pak, T., Butler, I. B., Geiger, S., van Dijke, M. I., & Sorbie, K. S. (2015). Droplet fragmentation: 3D imaging of a previously unidentified pore-scale process during multiphase flow in porous media. *Proceedings of the National Academy of Sciences*, *112*(7), 1947–1952.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... Lerer, A. (2017). Automatic differentiation in PyTorch.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*(336), 846–850.

Rashid, S., Lee, S. Y., & Hasan, M. K. (2012). An improved method for the removal of ring artifacts in high resolution ct imaging. *EURASIP Journal on Advances in Signal Processing*, *2012*(1), 93.

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352–2449.

Reynolds, C. A., Menke, H., Andrew, M., Blunt, M. J., & Krevor, S. (2017). Dynamic fluid connectivity during steady-state multiphase flow in a sandstone. *Proceedings of the National Academy of Sciences*, *114*(31), 8187–8192.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. , 234–241.

Sauvola, J., & Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern recognition*, *33*(2), 225–236.

Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., ... Menou, K. (2019). Lunar crater identification via deep learning. *Icarus*, *317*, 27–38.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Skourt, B. A., El Hassani, A., & Majda, A. (2018). Lung CT image segmentation using deep neural networks. *Procedia Computer Science*, *127*, 109–113.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. , 1–9.

van Vugt, J. (2017). *PyTorch U-Net.* `https://github.com/jvanvugt/pytorch-unet`. GitHub.

Weston, A. D., Korfiatis, P., Kline, T. L., Philbrick, K. A., Kostandy, P., Sakinis, T., ... Erickson, B. J. (2019). Automated abdominal segmentation of CT scans for body composition analysis using deep learning. *Radiology*, *290*(3), 669–679.