Contents lists available at ScienceDirect

Materials Today Communications

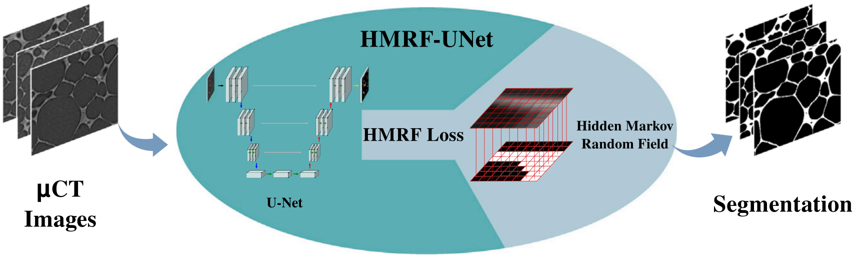
journal homepage: www.elsevier.com/locate/mtcomm

Unsupervised segmentation of micro-CT scans of polyurethane structures by combining hidden markov random fields and a U-Net

Julian Grolig[a,b,c,d,*], Lars Griem[a,b], Michael Selzer[a,e], Hans-Ulrich Kauczor[c,d], Simon M.F. Triphan[c,d], Britta Nestler[a,b,e], Arnd Koeppe[a,b,*]

[a] Institute of Nanotechnology (INT), Karlsruhe Institute of Technology (KIT), Straße Am Forum 7, Karlsruhe, 76131, Germany
[b] Institute for Applied Materials - Microstructure Modelling and Simulation (IAM-MMS), Karlsruhe Institute of Technology (KIT), Straße Am Forum 7, Karlsruhe, 76131, Germany
[c] Diagnostic and Interventional Radiology, University Hospital of Heidelberg, Im Neuenheimer Feld 420, Heidelberg, 69120, Germany
[d] Translational Lung Research Center Heidelberg (TLRC), German Center for Lung Research (DZL), Im Neuenheimer Feld 130.3, Heidelberg, 69120, Germany
[e] Institute of Digital Materials Science (IDM), Karlsruhe University of Applied Sciences, Moltkestrasse 30, Karlsruhe, 76133, Germany

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Segmentation
Hidden markov random fields
Porous structure

ABSTRACT

Extracting a digital representation of a material from images is a prerequisite for any quantitative structure-property analysis. Supervised convolutional neural networks (CNNs) now deliver state-of-the-art segmentation accuracy, but their performance depends on large, manually annotated training sets—an impractical requirement for most bulk micro-computed-tomography (μCT) studies. Classical unsupervised techniques such as Hidden-Markov Random Fields (HMRF) avoid the need for ground-truth labels, yet they are typically slow and yield lower-quality segmentations. Here, we introduce HMRF-UNet, a hybrid framework that embeds the probabilistic neighborhood model of HMRF directly into the U-Net's loss function. The loss simultaneously (i) enforces spatial smoothness through higher-order neighborhood terms, (ii) respects class-wise intensity distributions, and (iii) benefits from data-driven feature learning. By combining HMRF's label-free regularization with the fast inference of CNNs, the method delivers unsupervised segmentation at a speed comparable to that of supervised networks. We evaluate the approach on a μCT dataset of polyurethane (PU) foam. An ablation study quantifies the contribution of each neighborhood term, and the HMRF-UNet attains a Dice similarity coefficient of 0.957 ± 0.017, while processing a 256×256 slice in 200 ms on a single GPU; performance that rivals supervised baselines. To further diminish the reliance on annotated data, we propose a two-stage pre-training strategy: the network is first optimized with the HMRF loss on unlabeled data and subsequently fine-tuned on a minimal labeled subset. This approach recovers 98.4% of the fully supervised performance while using only 1% of ground-truth annotations. The proposed framework provides accurate, high-throughput segmentation without extensive manual labeling, enabling rapid, data-driven characterization of complex porous architectures across a broad range of material systems.

[*] Corresponding authors at: Institute of Nanotechnology (INT), Karlsruhe Institute of Technology (KIT), Straße Am Forum 7, Karlsruhe, 76131, Germany.
Email addresses: julian.grolig@kit.edu (J. Grolig), arnd.koeppe@kit.edu (A. Koeppe).

https://doi.org/10.1016/j.mtcomm.2026.114817
Received 26 November 2025; Received in revised form 26 January 2026; Accepted 6 February 2026
Available online 7 February 2026

1. Introduction

Micro-Computed Tomography (μCT) is a commonly used imaging technique to analyze the microstructure of materials. Materials often consist of several components, which are often only discernible on the microscale. The components of a foam structure are, for example, the solid matrix phase and the air phase. Before any quantitative analysis of a material can be conducted, the different components have to be distinguished. A common approach to identify and cluster unique components in images is called segmentation. Segmentation can be defined as the task of grouping voxels in an image into groups by meaning. Segmentation approaches can be classified into four main categories: threshold-based, region-based, model-based, and pixel-classification techniques [1].

In threshold-based methods, one or more thresholds for pixel values are determined to divide an image into different classes [2,3]. Region-based approaches, such as region growing, start from seed-points before connecting and grouping voxels, if they have similar properties and are spatially linked [4,5]. Other approaches combine threshold- and region-based methods to segment structures in images [6]. Model-based approaches usually combine (statistical) shape models and registration approaches to fit image structures to a template shape [7,8]. Another recent model-based approach used a combination of skeletonization and pruning for microstructure segmentation [9]. The last category, pixel-classification techniques, combines both unsupervised methods, such as Self-Organizing-Maps (SOM) [10] or Hidden Markov Random Fields (HMRF) [11,12], weakly supervised methods [13], and supervised methods such as convolutional neural networks (CNNs) [14,15]. Supervised CNNs, e.g., the commonly used U-Net model [16–22], or models using a ResNet backbone [23] generate state-of-the-art segmentations. However, the main limitation of supervised models is their requirement for a lot of annotated ground-truth data, which is often not available, because it is unknown or expensive to collect. Another drawback of using ground-truth data is the potential for added error if the data was incorrectly collected. Hidden Markov Random Fields are a widely used method for unsupervised segmentation, and are often iteratively optimized using the time-consuming combination of the expectation-maximization (EM) and iterated conditional modes (ICM) algorithms [1,11]. Recently, evolutionary algorithms (EAs) were shown to be good alternatives for solving the NP-Hard HMRF optimization problem, which is non-linear, complex, and has several local minima [24,25]. There are several limitations of the mentioned HMRF segmentation models. Firstly, evolutionary algorithms must be rerun for every new image to be segmented. The time required for segmentation using EAs is comparable to that of the EM-ICM techniques, but both are still significantly slower than predictions by trained CNNs. Furthermore, HMRF evolutionary methods require excessive computational resources for each new segmentation. The final drawback of HMRF segmentation is that, due to the one-shot learning, no experience from previous segmentations can be learned.

Recent segmentation approaches in the area of Materials Science focus on supervised approaches [20,22,23] or apply simple unsupervised clustering approaches such as K-Means clustering [21]. There is a lack of research in the area of unsupervised segmentation methods in Materials Science. When voxel-wise ground-truth masks are impossible or prohibitively expensive to obtain, we introduce a universally applicable unsupervised loss that delivers accurate segmentations without any manual labeling. Our technical contribution is the novel unsupervised segmentation network HMRF-UNet, which combines the HMRF concept with the common U-Net structure for unsupervised image segmentation. The HMRF-UNet uses HMRF-based losses to train the U-Net network and thereby combines the advantages of both methods. The HMRF-UNet learns without any ground-truth, but also learns from known segmentations, and generates fast and resource-saving segmentation predictions with segmentation accuracy comparable to supervised approaches. The objective of this paper is to demonstrate the performance and influence of different neighborhood terms and weights on segmentation results for a dataset of Polyurethane foam μCT images and compare them to the segmentation results of a supervised U-Net.

2. Methods

2.1. HMRF segmentation theory

Let the image $\boldsymbol{y}$ be a realization of a random field $\boldsymbol{Y}$, representing the observed pixel intensities. Let the segmentation $\boldsymbol{x}$ be a configuration of a second random field $\boldsymbol{X}$, which assigns a label (e.g., class or region) to each voxel in the image. Then we can call $\boldsymbol{X}$ a Hidden-Markov-Random-Field of $\boldsymbol{Y}$. Let s be a voxel on the image grid S and N_s be the neighborhood clique of s from the neighborhood system N, which connects all voxels in S. Then y_s can be defined as the value (e.g. density in a μCT image) of the voxel s within the image $\boldsymbol{y}$. The assigned class of voxel s can be described by $x_s \in L$, where L is the set of possible classes in the image. The image can be represented by a Gaussian mixture model, where each class $l \in L$ in the image $\boldsymbol{y}$ is represented by a Gaussian distribution with the parameter set $\theta_l = \{\mu_l, \sigma_l\}$ with mean μ_l and standard deviation σ_l. The posterior distribution is defined as:

$$P(y_s|\theta_l) = \frac{1}{\sqrt{2\pi\sigma_l^2}} \exp\left[-\frac{(y_s-\mu_l)^2}{2\sigma_l^2}\right] \tag{1}$$

and the conditional likelihood energy function $\boldsymbol{U_{ll}}(\boldsymbol{y}|\boldsymbol{x})$ is given by

$$\boldsymbol{U_{ll}}(\boldsymbol{y}|\boldsymbol{x}) = \sum_{s\in S}\left(\frac{(y_s-\mu_{x_s})^2}{2\sigma_{x_s}^2} + \ln\sigma_{x_s}\right). \tag{2}$$

Based on a possible segmentation $\boldsymbol{x}$, a stabilizing a priori probability term can be defined in the form of an energy function $\boldsymbol{U_n}(\boldsymbol{X}=\boldsymbol{x})$, which takes into account all neighborhood cliques N_s in image $\boldsymbol{Y}$ [11]:

$$\boldsymbol{U_n}(\boldsymbol{X}=\boldsymbol{x}) = \sum_{s\in S} E(x_s) \tag{3}$$

where $E(x_s)$ is the neighborhood penalty of the neighborhood N_s. One common neighborhood penalty is the Potts neighborhood [11,26], which only takes into account the class of neighboring voxels:

$$E_{\text{potts}}(x_s) = \alpha \sum_{t\in N_s}\left(1 - 2\delta_{x_s x_t}\right). \tag{4}$$

Here, $\delta_{x_s x_t}$ is the Kronecker delta between x_s and x_t, and α is a weight for the neighborhood term. Another possible neighborhood penalty term, which also includes Gaussian distribution properties of the neighbors' classes, was defined by Banerjee and Maji [27]:

$$E_{\text{ban}}(x_s) = \frac{\alpha_s}{2|N_s|}\sum_{t\in N_s}\left((\mu_{x_s}-\mu_{x_t})^2\left(\frac{1}{\sigma_{x_s}^2}+\frac{1}{\sigma_{x_t}^2}\right)-1\right) \tag{5}$$

where α_s is a voxel-specific weight factor, which is determined by

$$\alpha_s = \begin{cases} 2\alpha, & \text{if}\left(r_{s(1)}^{(2)} - r_{s(2)}^{(2)}\right) > T \quad \text{and} \quad \left(r_{s(1)}^{(1)} - r_{s(2)}^{(1)}\right) > T \\ \alpha, & \text{if}\left(r_{s(1)}^{(2)} - r_{s(2)}^{(2)}\right) > T \quad \text{and} \quad \left(r_{s(1)}^{(1)} - r_{s(2)}^{(1)}\right) \leq T \\ \alpha, & \text{if}\left(r_{s(1)}^{(2)} - r_{s(2)}^{(2)}\right) \leq T \quad \text{and} \quad \left(r_{s(1)}^{(1)} - r_{s(2)}^{(1)}\right) > T \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where T is a threshold and $r_{s(i)}^{(k)}$ is the relative number of neighbors belonging to the i^{th} most common class for the k-order neighborhood. A detailed explanation can be found in [27].

The optimal segmentation $\hat{\boldsymbol{x}}$ can be found by minimizing the sum of both energy terms in Eqs. (2) and (3):

$$\hat{\boldsymbol{x}} = \arg_{min}\left(\sum_{s\in S}\left(\frac{(y-\mu_{x_s})^2}{2\sigma_{x_s}^2} + \ln\sigma_{x_s} + E(x_s)\right)\right). \tag{7}$$

2.2. Novel fuzzy HMRF penalty formulations for neural networks

The HMRF segmentation task is usually described for the discrete case. However, the prediction of a CNN for segmentation is typically a fuzzy confidence map $\mathbf{c}$, which represents the probability that the voxel belongs to a certain class. The word fuzzy in this context means that a voxel is not assigned to a single class, but to multiple classes using a confidence value for each class. Therefore, we propose a reformulation of the discrete energy functions for the fuzzy case, where the confidences weight the contributions to the energy functions. The mean and standard deviation for each class were calculated using the weighted mean approach, where the value of each voxel contributes to the fuzzy mean $\widetilde{\mu}_l$ and the fuzzy standard deviation $\widetilde{\sigma}_l$ of a class based on its class probability $c_{s,l}$.

$$\widetilde{\mu}_l = \frac{\sum_{s\in S}\left(c_{s,l} \odot y_s\right)}{|c_{s,l}|} \tag{8}$$

$$\widetilde{\sigma}_l = \sqrt{\frac{\sum_{s\in S}\left(c_{s,l} \odot (y_s - \mu_l)^2\right)}{|c_{s,l}|}} \tag{9}$$

Here, $\odot$ is the Hadamard product and $|c_{s,l}|$ is the sum of all confidence values for class l. With the fuzzy mean and standard deviation for each class, we can calculate the fuzzy mean $\widetilde{\mu}_{x_s}$ and standard deviation $\widetilde{\sigma}_{x_s}$ belonging to a voxel s:

$$\widetilde{\mu}_{x_s} = \sum_{l\in L}(c_{s,l} \cdot \widetilde{\mu}_l) \tag{10}$$

$$\widetilde{\sigma}_{x_s} = \sum_{l\in L}(c_{s,l} \cdot \widetilde{\sigma}_l). \tag{11}$$

Using this definition, we can reformulate Eq. (2) as:

$$\widetilde{\boldsymbol{U}}_{II}(\mathbf{y}|\mathbf{x}) = \sum_{s\in S}\left(\frac{(y_s - \widetilde{\mu}_{x_s})^2}{2\widetilde{\sigma}^2_{x_s}} + \ln\widetilde{\sigma}_{x_s}\right). \tag{12}$$

The Potts neighborhood energy term from Eq. (4) was also reformulated to work for fuzzy labels. Instead of calculating the number of dissimilar labels inside the neighborhood, the average Euclidean distance between the confidence vector $\mathbf{c}_s$ of the voxel x_s and the confidence vectors $\mathbf{c}_t$ of all its neighbors x_t was calculated:

$$\widetilde{E}_{\text{potts}}(x_s) = \frac{\alpha}{|N_s|}\sum_{t\in N_s}\|\mathbf{c}_s - \mathbf{c}_t\|^2 . \tag{13}$$

The Banerjee neighborhood energy term from Eq. (5) was also reformulated for the fuzzy case:

$$\widetilde{E}_{\text{ban}}(x_s) = \frac{\alpha}{2|N_s|}\sum_{t\in N_s}\left((\widetilde{\mu}_{x_s} - \widetilde{\mu}_{x_t})^2\left(\frac{1}{\widetilde{\sigma}^2_{x_s}} + \frac{1}{\widetilde{\sigma}^2_{x_t}}\right) + 1\right). \tag{14}$$

For the custom voxel weighting α_s of the neighborhood term, we adapted Eq. (6) by using the standard deviation inside the neighborhood instead of the relative class occurrence:

$$\widetilde{\alpha}_s = \begin{cases} 2\alpha, & \text{if} \quad \sigma^{(1)}_{N_s} < \sigma_{\text{thresh}} \quad \text{and} \quad \sigma^{(2)}_{N_s} < \sigma_{\text{thresh}} \\ \alpha, & \text{if} \quad \sigma^{(1)}_{N_s} \geq \sigma_{\text{thresh}} \quad \text{and} \quad \sigma^{(2)}_{N_s} < \sigma_{\text{thresh}} \\ \alpha, & \text{if} \quad \sigma^{(1)}_{N_s} < \sigma_{\text{thresh}} \quad \text{and} \quad \sigma^{(2)}_{N_s} \geq \sigma_{\text{thresh}} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $\sigma^{(k)}_{N_s}$ is the standard deviation in the k-th order neighborhood of voxel s and σ_{thresh} is a threshold for the standard deviation in the neighborhood of voxel s. With these adaptations, we can now reformulate Eq. (3) to

$$\widetilde{\boldsymbol{U}}_{\boldsymbol{n}}(\boldsymbol{X} = \boldsymbol{x}) = \sum_{s\in S}\left(\alpha\widetilde{E}_*(x_s)\right) \tag{16}$$

where α can be either a fixed constant (normal neighborhood term) or a custom weight for each voxel (weighted neighborhood term), calculated using Eq. (15). $\widetilde{E}_*$ is calculated with Eqs. (13) or (14).

2.3. HMRF-UNet

We used a vanilla U-Net for segmentation of the μCT images. The U-Net consists of an encoder and a decoder path, where the corresponding encoder and decoder levels are connected using skip connections. In each level, several convolution blocks are used, each consisting of a 3×3 Conv2D layer, a batch-normalization layer, and a ReLU layer. Between levels in the encoding path, a pooling layer with a pool size of 2 was used, while between levels in the decoding path, UpConv2D layers were used. The number of kernels per level doubled or halved with each level for the encoding and decoding path, respectively. The final layer was a 1×1 Conv2D layer with softmax activation and two output features. An exemplary representation of the architecture is shown in Fig. 1.

The loss of the model consisted of two loss components, a distribution loss $\mathcal{L}_d$ and a neighborhood loss $\mathcal{L}_n$, which were weighted using the weights λ_d and λ_n accordingly:

$$\mathcal{L} = \lambda_d\mathcal{L}_d + \lambda_n\mathcal{L}_n \tag{17}$$

The distribution loss $\mathcal{L}_d$ was calculated using Eq. (12) and the neighborhood loss was calculated using Eq. (16) based on a first-order neighborhood system with eight neighbors. For all the following studies, the distribution weight was set to $\lambda_d = 1 - \lambda_n$.

2.4. Polyurethane foam datasets

In this study, two different datasets were used. The first dataset (*RealPUFoam*) contains real μCT images of Polyurethane (PU) foam structures. The images have a size of 1300 × 951 × 960 voxels with a spatial resolution of 2.75 µm × 2.75 µm × 2.75 µm. The second dataset (*ArtPUFoam*) consisted of 20,000 artificially generated 2D μCT images of PU foam structures with a size of 256 × 256 pixels and their ground-truth segmentations. The images of the artificial dataset were generated using a multistep workflow as described by Griem et al. [28]. In this workflow, a digital twin is created from an initial set of real μCT images. The PACE3D simulation framework [29] extracts structural features to generate similar binary foam structures. These binary foam structures are converted into grayscale images and paired into an artificial input image and a corresponding segmentation map to train a U-Net for image segmentation. The trained U-Net segments the original μCT images, which are then used to train a generative adversarial network (GAN) for the generation of binary foams. Finally, a CycleGAN translates the GAN-generated binary images into grayscale images, guided by the structural characteristics of the real μCT data.

2.5. Data preprocessing

Both datasets were preprocessed before being split into training, validation, and test sets. For both datasets, the grayscale images were normalized to the range $[0, 1]$. Since the artificially generated dataset already contained large amounts of heterogeneous and independent 2D images, no further processing was necessary, and the images were split into training, validation, and testing sets with a split of $(70\%/15\%/15\%)$. Since the real PU foam images were 3D volumes, 2D slices had to be extracted. The adjacent slices of μCT images share a lot of similarity, which could cause a leak between training, validation, and the test sets. Therefore, we developed a cuboid-based dataset split as shown in Fig. 2.

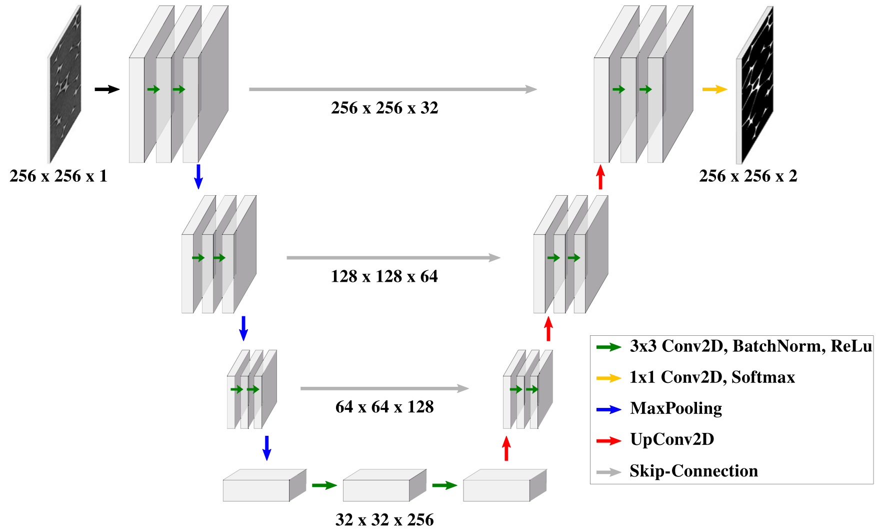


Fig. 1. Overview of the segmentation U-Net architecture.

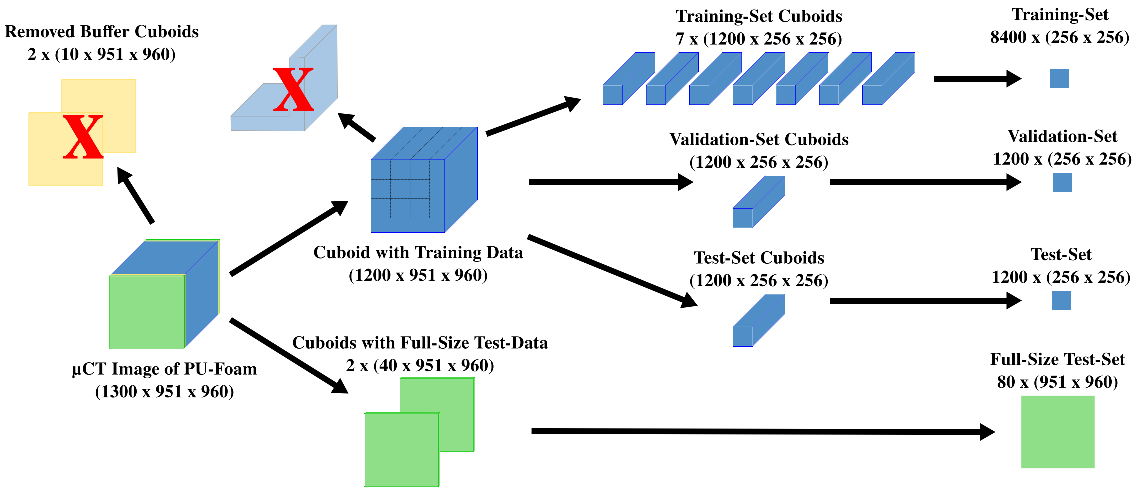


Fig. 2. Visualization of the proposed cuboid-based dataset split. By extracting non-overlapping cuboids and splitting the data based on the cuboids, a leakage between training, validation and test set can be prevented.

Similarly to the common group-wise split, the separation of cuboids prevents a leakage between the different datasets. We first removed 40 slices, from the front and back of the image, for an independent full-size test set. The following ten slices from front and back were discarded to prevent dataset leakage from the full-size test set to the training set. The cuboids were generated from the remaining volume with 1200 slices using a non-overlapping moving window of size 1200 × 256 × 256, resulting in nine separate cuboids. From each cuboid, slices of shape 256 × 256 were extracted. In addition, to increase the amount and heterogeneity of data, five different augmentations per slice were generated. A random contrast adjustment in the range [0.8, 1.2] was combined with one of five transformations consisting of 90°-rotation, 180°-rotation, 270°-rotation, x-axis-flip, and y-axis-flip. Overall this resulted in 42,000 training, 6000 validation, and 6000 test images.

2.6. Network optimization and training

The artificial dataset with pairs of images and ground-truth segmentations was used to find the optimal hyperparameters and to determine the optimal loss setup. A first hyperparameter search for only network parameters was performed using Bayesian optimization with the Dice score (see Section 2.7) as optimization metric. Here, different values were tested for the size and starting number of kernels, levels, and convolutions per level, as well as type of pooling. The optimal resulting architecture was a U-Net with three levels of three convolutions each, max-pooling, a kernel size of 3×3, and 64 kernels in the top level. Using this fixed model setup, another Bayesian optimization was performed to find the optimal weight for the neighborhood loss term. The five best-performing values for the neighborhood loss weight were then used to train all models for further experiments. Based on empirical experiments, the learning rate was set to 1×10^{-5} and the batch size to 128 for all experiments. Unless mentioned otherwise, all trainings were run for a total of 200 epochs. The computational setup consisted of an Nvidia A100 with 80GB of GPU-RAM and 128 AMD EPYC 7513 32-core processors with an overall RAM of 2100GB. All searches and trainings were performed using Tensorflow 2.11.

2.7. Evaluation metrics

For the evaluation of the different models, which were trained using the *ArtPUFoam* dataset, we used the commonly used Dice score (DSC) [30] as shown in Eq. (18) between the predicted segmentation x

and the ground-truth segmentation $\hat{x}$. The Dice score values range from 0 (bad segmentation) to 1 (perfect segmentation). To ensure a fair comparison of models, a threshold sweep was run on the predicted confidence maps, to find the optimal threshold for creating the binary segmentation map, which was then used for the final calculation of the Dice score.

$$\mathrm{DSC} = \frac{2\left|x \cap \hat{x}\right|}{\left|x\right| + \left|\hat{x}\right|} \tag{18}$$

3. Results

3.1. Numerical studies

In a first study, the influence of the type and weight λ_n of the neighborhood term on the segmentation performance was investigated using the *ArtPUFoam* dataset. For this, we trained four different models with normal Potts neighborhood (HMRF-UNet$_{\mathrm{pot}}$), weighted Potts neighborhood (HMRF-UNet$_{\mathrm{wpot}}$), normal Banerjee neighborhood (HMRF-UNet$_{\mathrm{ban}}$), and weighted Banerjee neighborhood (HMRF-UNet$_{\mathrm{wban}}$) for the five different best-performing neighborhood loss weights from the hyperparameter search. Another study investigated the influence of the parameter σ_{thresh} in Eq. (15). For this HMRF-UNet$_{\mathrm{wban}}$ models with $\lambda_n = 0.31$ and σ_{thresh} values of 0.05, 0.10, 0.15, and 0.20 were trained. To test the usage of the HMRF-UNet for pre-training, a study was set up in which the performance of a supervised U-Net model trained with fractions of the *ArtPUFoam* dataset with and without starting with pre-trained weights from an unsupervised HMRF-UNet was compared. The weights of the pre-trained model were extracted from an HMRF-UNet$_{\mathrm{pot}}$ model with $\lambda_n = 0.31$. For the supervised training of the U-Net we used the Dice loss suggested by Milletari et al. [15]. In a final study, it was investigated how models trained on the *RealPUFoam* dataset compare to models trained with the artificial *ArtPUFoam* dataset. For this purpose, identical models with $\lambda_n = 0.31$ and normal Potts neighborhood were trained both on the *ArtPUFoam*, and on the *RealPUFoam* dataset.

3.2. Influence of neighborhood term

The neighborhood loss can be defined by one of the four different neighborhood term choices (normal/weighted Potts and normal/weighted Banerjee). Table 1 shows the average Dice scores for different combinations of neighborhood terms and weights. All models trained with the normal Banerjee term had considerably worse segmentations based on their average Dice scores compared to all other models. The models with the Potts neighborhood had slightly higher Dice scores compared with the models with the weighted Banerjee neighborhood. For custom weighted neighborhood terms, a decrease in the neighborhood-loss weight factor λ_n led to an improvement of the segmentation, while the terms without custom weighting showed no influence of λ_n. All HMRF-UNet models significantly outperformed the common baseline approach based on thresholding using Otsu's method [2]. Fig. 3 shows a visual comparison of the predicted segmentations of a model trained without a neighborhood term and the predicted segmentations of models trained with each of these neighborhood terms and a neighborhood weight of $\lambda_n = 0.31$. In all cases, except for the worst image in the last row, the models with normal and weighted Potts neighborhoods and the model with weighted Banerjee neighborhood generated better segmentations than the model without the neighborhood term. The model without the neighborhood term HMRF-UNet$_{\mathrm{noneigh}}$ over-segmented the PU structure inside the pore space. We can also see that the model with normal Banerjee neighborhood always generated the worst segmentation result. For the bottom image, this was due to over-segmented PU structures around the PU walls (blue areas). The custom weighting led to improved segmentations for the Banerjee neighborhood formulation, but resulted in worse segmentations when used with the Potts neighborhood. For all models, the main segmentation errors were due to under-segmented PU structures (red areas).

3.3. Threshold for custom neighborhood-weighting

An analysis revealed the influence of the threshold σ_{thresh} used for the calculation of the custom neighborhood weight. Fig. 4 shows the influence of σ_{thresh} for two exemplary images of the test set. For both images, a reduction of the threshold led to improved segmentation results. Specifically, thin PU struts were partially detected for the smaller thresholds of 0.05, which were not detected using higher thresholds of 0.20, as can be seen in the areas marked with red circles in Fig. 4.

3.4. HMRF-UNet for pre-training

Table 2 compares models that were trained using a supervised Dice loss with and without utilizing the weights of a model trained with unsupervised HMRF-loss as a starting configuration for different amounts of training data. It can be observed that for the models without any pre-training, only the models trained with at least 1000 images achieved high Dice scores. The models with pre-training already achieved high Dice scores for the smallest training set of only five images. The supervised training improved the Dice scores of the unsupervised HMRF-UNet from 0.957 to 0.977, 0.983 and 0.992, when fine-tuning on five, ten or 100 images, respectively. The effect of pre-training can also be observed in Fig. 5, which compares the segmentation results of the images in the test set with the smallest (first row), median (second row) and largest change (third row) for supervised models with and without pre-training using the HMRF-UNet. The overlay plots in the third and fourth columns show how fine-tuning with more images improved the segmentation. It can be observed that the unsupervised HMRF-UNet already generated good segmentations (white structures). Most of the missing fine PU walls were recovered after fine-tuning with ten images. When fine-tuning with 100 supervised examples, we achieved near-perfect segmentation results. For the supervised model without pre-training (third column), a training with 1000 images was necessary to successfully segment the fine PU walls. The boxplots of the Dice scores of both model variants in Fig. 6 confirm these observations. For all amounts of fine-tuning/training data, the model with HMRF-UNet pre-training outperformed the model without pre-training. The significance of these differences was confirmed by a Wilcoxon signed-rank test, which resulted in p-values of p<0.001 for all five cases.

Table 1
Comparison of average Dice scores (Mean ± Std) for different neighborhood terms and weights.

	Dice Score				
Otsu's Thresholding	0.882 ± 0.025				
NoNeighborhoodLoss	0.950 ± 0.015				
λ_n	0.09	0.18	0.31	0.35	0.56
NormalPottsLoss	0.927 ± 0.077	**0.956 ± 0.016**	**0.957 ± 0.017**	**0.957 ± 0.017**	**0.951 ± 0.019**
WeightedPottsLoss	**0.956 ± 0.015**	**0.956 ± 0.017**	0.954 ± 0.018	0.952 ± 0.019	0.950 ± 0.016
NormalBanerjeeLoss	0.952 ± 0.015	0.914 ± 0.091	0.882 ± 0.103	0.882 ± 0.103	0.771 ± 0.088
WeightedBanerjeeLoss	0.955 ± 0.016	0.953 ± 0.018	0.947 ± 0.019	0.946 ± 0.019	0.943 ± 0.013

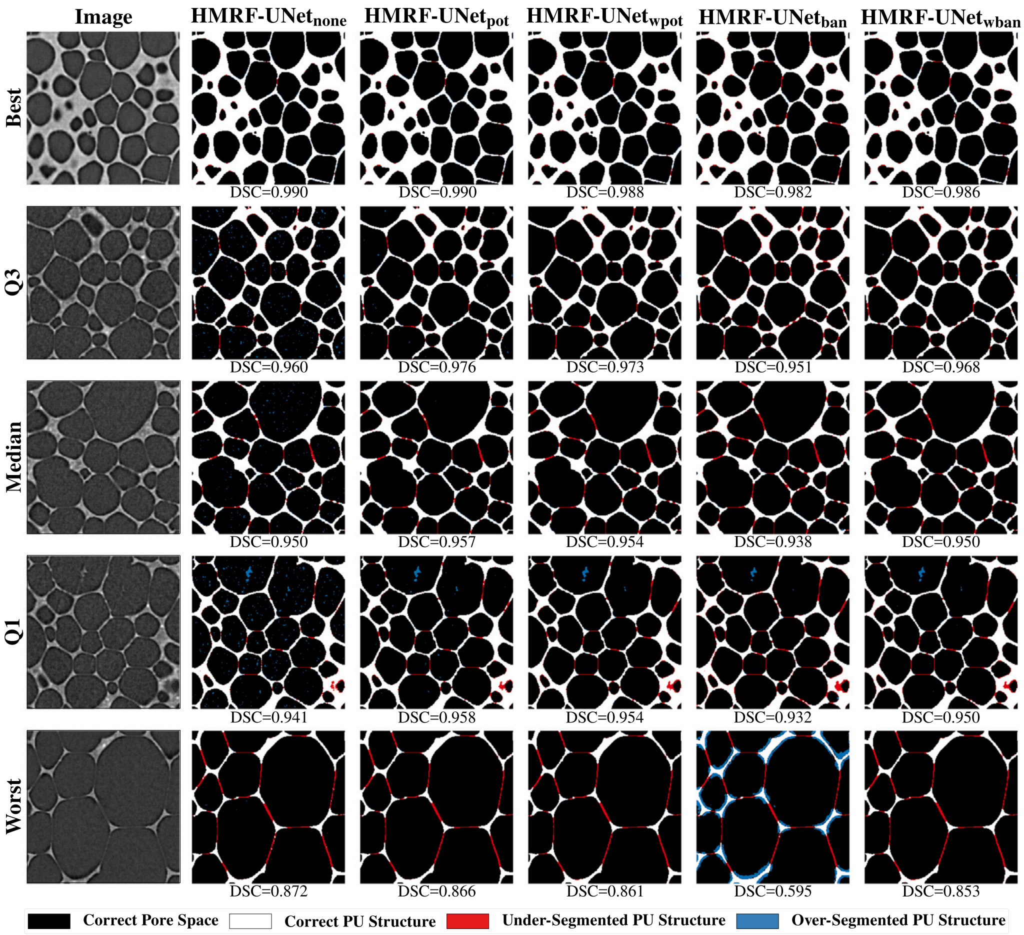


Fig. 3. Comparison of segmentation results for different images from the testing set for five models trained with different neighborhood terms. The models trained with neighborhood term (columns 2-6) were trained with a neighborhood weight of $\lambda_n = 0.31$. The model without neighborhood term (second column) over-segments the PU structure inside the pore space (blue areas). Training with the neighborhood term removes these over-segmentations. For the normal Banerjee neighborhood term an over-segmentation around the PU walls can be observed for the worst test image in the bottom row (blue areas). Otherwise most segmentation errors resulted from under-segmented PU structures (read areas).

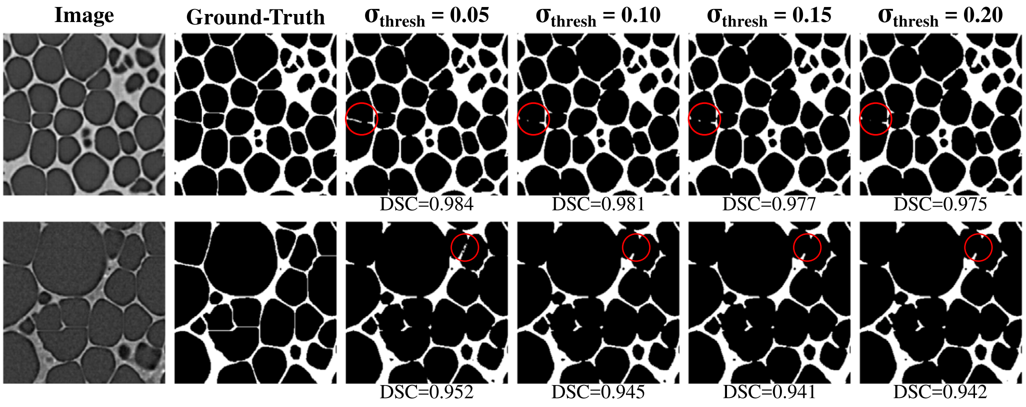


Image
Ground-Truth
$\sigma_{\text{thresh}} = 0.05$
$\sigma_{\text{thresh}} = 0.10$
$\sigma_{\text{thresh}} = 0.15$
$\sigma_{\text{thresh}} = 0.20$
DSC=0.984
DSC=0.981
DSC=0.977
DSC=0.975
DSC=0.952
DSC=0.945
DSC=0.941
DSC=0.942

Fig. 4. Comparison of segmentation results of models trained with weighted Banerjee neighborhood with different thresholds σ_{thresh} for two exemplary images. In the regions marked with red circles, differences of segmentation accuracy become apparent, showing improved segmentation of thin structures for smaller thresholds.

3.5. Real μCT validation

To validate the segmentation performance of the HMRF-UNet on the real μCT dataset, three different models all with normal Potts neighborhood and $\lambda_n = 0.31$ were trained. The first model was trained on the *ArtPUFoam* dataset, and the second model was based on the first model but finetuned with 100 ground-truth images. The third model was trained on the *RealPUFoam* dataset without any finetuning afterwards. The predicted segmentations for an exemplary full-size test image are shown in Fig. 7. It can be observed that some thin PU walls were never segmented (red circles), while other thin PU walls were at least segmented by the fine-tuned model (yellow dotted circle). It also becomes

Table 2
Analysis of the influence of using a pretrained HMRF-UNet as a starting point for supervised learning of segmentation tasks. Comparison of average Dice scores (Mean ± Std) for models trained with and without pre-training for different supervised training set sizes.

	Dice Score				
Training image amount	5	10	100	1000	14,000 (all)
Unsupervised HMRF-UNet	-	-	-	-	0.957 ± 0.017
Supervised U-Net without Pre-Training	0.848 ± 0.040	0.859 ± 0.043	0.854 ± 0.038	0.999 ± 0.001	1.000 ± 0.000
Supervised U-Net with HMRF Pre-Training	0.977 ± 0.015	0.983 ± 0.010	0.992 ± 0.005	0.999 ± 0.000	-

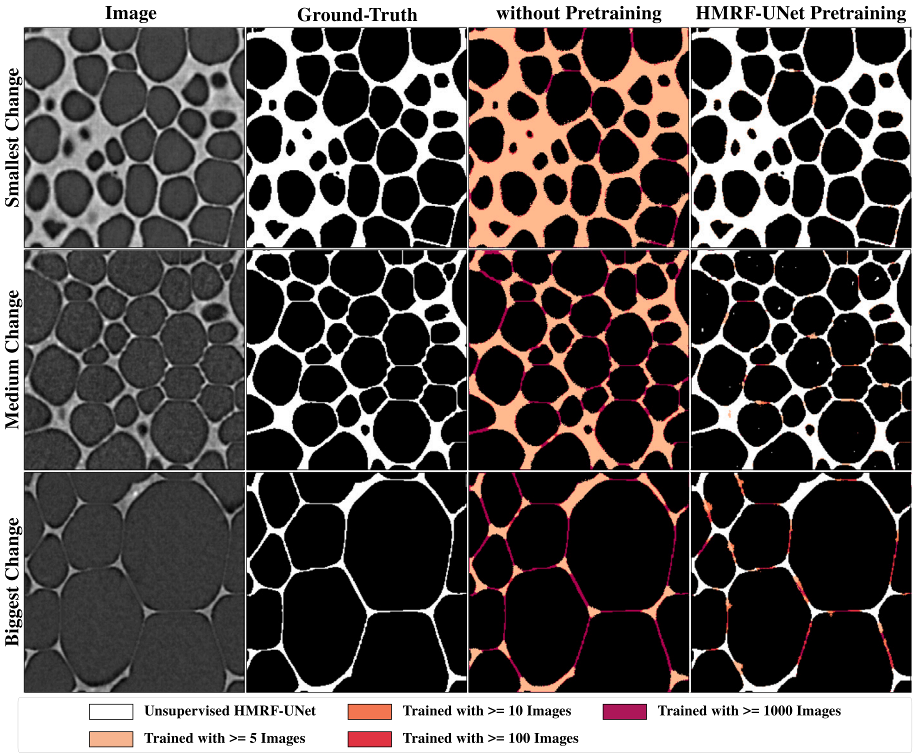


Fig. 5. Effect of using HMRF-UNet as a pretrained model for finetuning with different amount of ground-truth pairs using supervised loss. The grayscale image (first column), the ground-truth segmentation (second column), the segmentation of the supervised model without pretraining (third column) and the segmentation of the supervised model with HMRF-UNet pretraining are shown for examples of the test set with the smallest change (first row), median change (second row) and biggest change (third row). The colour coding of the segmentations in the third and fourth column show the improvement of using more images for fine-tuning.

apparent that the model trained on the real dataset generated over-segmented PU structures and produced the visually worst segmentation results.

4. Discussion

4.1. Choice of neighborhood term

We have shown that the choice of neighborhood loss definition has an important influence on the segmentation quality of the model. In general, the experiments suggest that both Potts neighborhood types outperform the newer Banerjee neighborhood types. A reason for this might be that for a binary segmentation the regular Banerjee neighborhood does not bring any benefit over the Potts neighborhood, since calculating the difference of the means of a class is not too different from calculating the distance between the confidence map predictions. Another possible reason might be that the mean and standard deviations in the original implementation of the Banerjee neighborhood are based on non-fuzzy, fixed class assignments and are therefore just the properties of one class. In our case, the means and standard deviations are, however, calculated using a weighting with the confidence value of a voxel.

It could also be observed that custom neighborhood weighting is especially important for the Banerjee neighborhood loss. Without custom weighting, the segmentation results were remarkably worse compared to the segmentation results from the models with custom weighting. The exemplary segmentations in Fig. 3 suggest that the low Dice scores may be attributed to over-segmentations for input images with lower contrast. The contrast in the input image appeared to influence not only the segmentation quality of the HMRF-UNet$_{\mathrm{ban}}$, but also all the different models, since for all four models the Dice score was remarkably lower for the image with low contrast in the last row. To test this hypothesis, we calculated the standard deviation inside the input images as a measure of the contrast of the image. The average standard deviations of the 50 worst and 50 best predictions for the HMRF-UNet$_{\mathrm{ban}}$ model were calculated and resulted in an average standard deviation of 0.187 ± 0.020 for the best predictions and 0.097 ± 0.011 for the worst predictions. This observation supports our hypothesis that a lower image contrast leads to worse segmentations. The choice of the right threshold

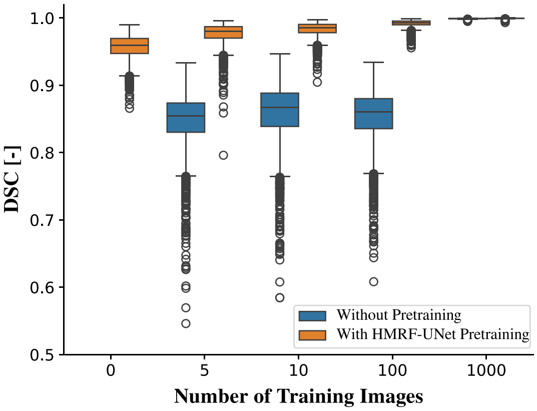


Fig. 6. Boxplots comparing the Dice scores on the test set for the supervised models without HMRF-UNet pretraining (blue) and the supervised models with HMRF-UNet pretraining (orange). For all amount of training images, the pre-training significantly (p<0.001) improved the Dice scores (DSC) of the segmentation.

value in the weighting term can also help to improve the segmentation of small structures in the image, as shown in Fig. 4.

4.2. Weaknesses of the HMRF loss

When investigating the segmentations obtained with the HMRF-UNet, we could observe that the model often struggles with the thin walls of the PU foam structure. One reason for this is the low intensities of some of the small walls. This could lead to possible assignments to the darker background/air class, since the voxel intensity fits better with the air class distribution. In supervised approaches, the model receives feedback from the ground-truth labels, which enables these models to even learn the segmentation of structures that are hardly visible in the image itself. This is a challenge that most unsupervised methods share. Another reason for missing some of the small walls might be the neighborhood loss. Since these thin walls are often only one voxel thin, there are only two neighbors belonging to the same PU class. Since most of the neighbors belong to the air class, the neighborhood loss punishes the model if it classifies the voxel as PU structure.

The redundant class problem described by Nan et al. [31] could also influence the quality of the segmentation. We set the number of classes of the HMRF model to the expected number of unique classes, which is two in our case. However, some of the voxels in our image might jump between these two classes. Therefore, using more than two classes could be beneficial, allowing these uncertain voxels to be assigned to their own class. Also, artifacts like dark shadows, or bright points influence the properties of the distribution of their class, which could lead to incorrect distribution properties. If these artifacts get assigned to their own, additional class, this could allow us to ignore these artifacts.

4.3. Effectiveness of unsupervised pre-training

The positive effect of unsupervised pre-training due to a regularization effect was shown before [32–34]. Our results suggest that a pre-training using the unsupervised HMRF loss can reduce the required amount of ground truth data for training segmentation models. We could show that fine-tuning the HMRF-UNet with only five to ten images significantly improved the segmentation quality, and fine-tuning with 100 ground-truth images led to near-perfect segmentation performance on the *ArtPUFoam* dataset. This is in good agreement with the results of Kalapos and Gyires-Tóth, who showed that they could achieve near-perfect segmentations by fine-tuning a pre-trained model with only 100 ground truth images [34]. This demonstrates the potential of unsupervised HMRF loss for any segmentation task, suggesting it as a viable alternative or supplementary pre-training method to common approaches such as clustering, contrastive learning, and generative models.

4.4. Training with artificial vs. real datasets

When comparing the segmentation results for the real μCT images in Fig. 7, it can be observed that the HMRF-UNet trained on the *RealPUFoam* dataset surprisingly performed worse than the model trained on the *ArtPUFoam* dataset. The oversegmentations can be explained by a dark border between the pore space and the PU walls. This dark border has extremely low intensities, which are far away from the mean intensities of both classes. Since we predefined the number of classes to two, the dark borders had to be merged with one of the other two classes. The mentioned dark borders were not present in the artificial dataset, and therefore, the predictions of the models trained with

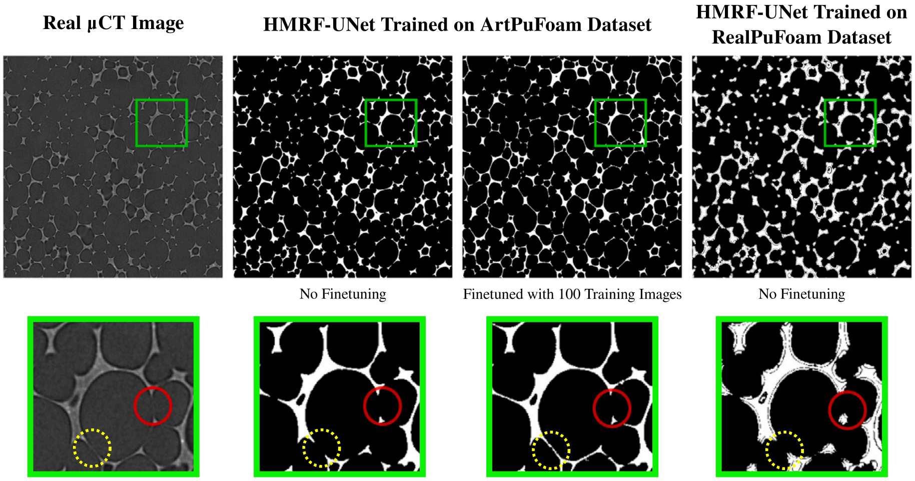

Real µCT Image
HMRF-UNet Trained on ArtPuFoam Dataset
HMRF-UNet Trained on RealPuFoam Dataset
No Finetuning
Finetuned with 100 Training Images
No Finetuning

Fig. 7. Comparison of predicted segmentations for an exemplary image from the *RealPUFoam* dataset for different models trained with regular potts neighborhood and $\lambda_n = 0.31$. Top row: full segmentation, bottom row: enlarged regions of interest. The HMRF-UNet trained on the *RealPUFoam* generates worse segmentations compared to the two models trained on the *ArtPUFoam* dataset. Red circles indicate an area where all models missed fine PU walls, yellow dotted circles mark areas where finetuning improves the segmentation of thin walls.

this dataset did not generate over-segmentations. The overall worse segmentation quality for the real dataset can be attributed to two reasons. Again, the dark borders around PU walls likely also influence the prediction and prevent the correct segmentation of thin PU walls. The second reason could be partial volume artifacts, which might be present in the real μCT and lead to mixed signals of PU and air. This effect might be especially pronounced in the thin PU walls, which could lead to worse segmentation accuracy in these areas.

4.5. Limitations

Our study has several limitations. For the training of the model, only pixel intensities were used. We did not use any textural features. In addition, we only tested our proposed HMRF-UNet on real and artificial μCT data of PU foams. Since PU foam structures consist of only two classes, the segmentation task was easier, compared to images with more than two classes. Furthermore, only ground-truth data for the artificial dataset was available. Therefore, the performance of our model could only be quantitatively evaluated for this artificial dataset and not for the dataset with real μCT images, where only a qualitative evaluation was possible. In general, we want to emphasize that this study was a proof-of-concept study to show the potential of the HMRF-UNet. More experiments should be carried out to explore additional influencing factors and possible optimizations in the loss function or other model components.

5. Conclusion

In this work, the application of HMRF theory in constructing an HMRF loss for unsupervised segmentation using a U-Net was demonstrated. By combining the advantages of Hidden Markov Random Fields and CNN segmentation, we trained an unsupervised model that predicts segmentation maps in around 200 ms for an image of size 256×256. Analysis of the loss components highlighted the critical role of the neighborhood loss and the impact of its configuration. Pre-training with the HMRF loss markedly reduced the amount of ground-truth data needed to achieve high-quality segmentations. To our knowledge, this is the first integration of an HMRF loss into a CNN for end-to-end unsupervised segmentation, establishing the HMRF-UNet as a valuable complement to existing unsupervised methods such as contrastive learning and SOMs.

Other authors have shown that using a combination of supervised and unsupervised loss terms can also improve the performance of the model [35]. Therefore, a subsequent investigation could explore coupling the HMRF loss with a supervised Dice loss in a U-Net. We also plan to examine synergistic effects between the HMRF loss and contrastive learning to further enhance fully unsupervised segmentation. Additional research could assess whether extending the model to more than two classes improves accuracy and validate the approach on diverse material-science and medical datasets. With minor adaptations, the HMRF-UNet can be trained on 3D volumes. Further studies should evaluate whether a 3D HMRF-UNet, incorporating 3D neighborhood systems, improves the segmentation of the real μCT images, especially for thin PU walls.

CRediT authorship contribution statement

Julian Grolig: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Lars Griem:** Writing – review & editing, Data curation, Conceptualization. **Michael Selzer:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Hans-Ulrich Kauczor:** Writing – review & editing, Supervision, Conceptualization. **Simon M.F. Triphan:** Writing – review & editing, Supervision, Conceptualization. **Britta Nestler:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Arnd Koeppe:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Methodology, Conceptualization.

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.

Acknowledgments

The study was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - #NE822/31 & 554598213. Contributions were also funded by the Helmholtz Association within the joint lab MDMC and the program MSE, no. 43.31.01 which is gratefully acknowledged.

Data availability

The used image datasets and the scripts for training the HMRF-UNet are publicly available at https://doi.org/10.5281/zenodo.17590658.

References

[1] N. Gordillo, E. Montseny, P. Sobrevilla, State of the art survey on MRI brain tumor segmentation, Magn. Reson. Imaging 31 (2013) 1426–1438, https://doi.org/10.1016/j.mri.2013.05.002

[2] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1979) 62–66, https://doi.org/10.1109/TSMC.1979.4310076

[3] T.W. Ridler, S. Calvard, Picture thresholding using an iterative slection method, IEEE Trans. Syst. Man Cybern. SMC-8 (1978) 630–632, https://doi.org/10.1109/tsmc.1978.4310039

[4] R. Adams, L. Bischof, Seeded region growing, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 641–647, https://doi.org/10.1109/34.295913

[5] S.A. Hojjatoleslami, J. Kittler, Region growing: a new approach, IEEE Trans. Image Process. 7 (1998) 1079–1084, https://doi.org/10.1109/83.701170

[6] L. Bogunia, S. Buchen, K. Weinberg, Microstructure characterization and stochastic modeling of open-cell foam based on μCT-image analysis, GAMM-Mitteilungen 45 (2022) e202200018, https://doi.org/10.1002/gamm.202200018

[7] A. Neumann, C. Lorenz, Statistical shape model based segmentation of medical images, Comput. Med. Imaging Graph. 22 (1998) 133–143, https://doi.org/10.1016/S0895-6111(98)00015-9

[8] A. Kelemen, G. Szekely, G. Gerig, Elastic model-based segmentation of 3-D neuroradiological data sets, IEEE Trans. Med. Imaging 18 (1999) 828–839, https://doi.org/10.1109/42.811260

[9] V.V. Deshpande, R. Piat, A skeletonization based image segmentation algorithm to isolate slender regions in 3D microstructures, Mater. Des. 252 (2025) 113765, https://doi.org/10.1016/j.matdes.2025.113765

[10] T. Kohonen, The self-organizing map, Proc. IEEE 78 (1990) 1464–1480, https://doi.org/10.1109/5.58325

[11] Y. Zhang, M. Brady, S. Smith, Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm, IEEE Trans. Med. Imaging 20 (2001) 45–57, https://doi.org/10.1109/42.906424

[12] B. Panić, M. Borovinšek, M. Vesenjak, S. Oman, M. Nagode, A guide to unsupervised image segmentation of mCT-scanned cellular metals with mixture modelling and markov random fields, Mater. Des. 239 (2024) 112750, https://doi.org/10.1016/j.matdes.2024.112750

[13] J. Na, S.-J. Kim, H. Kim, S.-H. Kang, S. Lee, A unified microstructure segmentation approach via human-in-the-loop machine learning, Acta Mater. 255 (2023) 119086, https://doi.org/10.1016/j.actamat.2023.119086

[14] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, arXiv preprint arXiv:1505.04597, 2015.

[15] F. Milletari, N. Navab, S.-A. 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

[16] O. Furat, M. Wang, M. Neumann, L. Petrich, M. Weber, C.E. Krill, et al., Machine learning techniques for the segmentation of tomographic image data of functional materials, Front. Mater. 6 (2019), https://doi.org/10.3389/fmats.2019.00145

[17] F. Isensee, P.F. Jaeger, S.A.A. Kohl, J. Petersen, K.H. Maier-Hein, nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation, Nat. Methods 18 (2021) 203–211, https://doi.org/10.1038/s41592-020-01008-z

[18] J.R. Astley, A.M. Biancardi, P.J.C. Hughes, H. Marshall, G.J. Collier, H.-F. Chan, L.C. Saunders, L.J. Smith, M.L. Brook, R. Thompson, S. Rowland-Jones, S. Skeoch, S.M. Bianchi, M.Q. Hatton, N.M. Rahman, L.-P. Ho, C.E. Brightling, L.V. Wain, A. Singapuri, R.A. Evans, A.J. Moss, G.P. McCann, S. Neubauer, B. Raman, C-MORE/PHOSP-COVID Collaborative Group, J.M. Wild, B.A. Tahir, Implementable deep learning for multi-sequence proton MRI lung segmentation: a multi-center, multi-vendor, and multi-disease study, J. Magn. Reson. Imaging 58 (2023) 1030–1044, https://doi.org/10.1002/jmri.28643

[19] S. Medghalchi, J. Kortmann, S.-H. Lee, E. Karimi, U. Kerzel, S. Korte-Kerzel, Automated segmentation of large image datasets using artificial intelligence for microstructure characterisation and damage analysis, Mater. Des. 243 (2024) 113031, https://doi.org/10.1016/j.matdes.2024.113031

[20] W.D. Romero, Y. Gutierrez, L.M. Tami-Pimiento, S. Torres-Bermudez, A.M. Meléndez, F. Martínez, Automatic pore shape characterization in metal

foams templated by hydrogen bubbles from a deep learning strategy, Mater. Today Commun. 41 (2024) 110937, https://doi.org/10.1016/j.mtcomm.2024.110937

[21] G. Bo, H. Zhou, C. Wang, C. Zhang, C. Deng, D. Jiang, et al., Automatic Si phase extraction from microscopic images of Al-Si alloys by unsupervised machine learning and supervised deep learning, Mater. Today Commun. 42 (2025) 111468, https://doi.org/10.1016/j.mtcomm.2024.111468

[22] L.A. Jara-Lugo, J. Caro-Gutierrez, F.F. Gonzalez-Navarro, M.A. Curiel-Alvarez, A. Armenta-Garcia, O.M. Perez-Landeros, Semantic and instance segmentation deep learning methods for nanoparticles detection, Mater. Today Commun. 45 (2025) 112074, https://doi.org/10.1016/j.mtcomm.2025.112074

[23] N. Soboleva, A. Mushnikov, Improving the accuracy of semantic segmentation of carbides in the microstructure of composite coatings by the neural network, Mater. Today Commun. 38 (2024) 108276, https://doi.org/10.1016/j.mtcomm.2024.108276

[24] E.-H. Guerrout, R. Mahiou, D. Michelucci, B. Randa, O. Assia, Hidden markov random fields and cuckoo search method for medical image segmentation, arXiv preprint arXiv:2005.09377, 2020.

[25] R. Mahiou, E.-H. Guerrout, M.E. Sannef, Taguchi design for setting EHO variants parameters: application in brain image segmentation using HMRF, SN Comput. Sci. 4 (2023) 794, https://doi.org/10.1007/s42979-023-02197-y

[26] S. Ait-Aoudia, R. Mahiou, E. Guerrout, Evaluation of volumetric medical images segmentation using hidden markov random field model, in: 2011 15th International Conference on Information Visualisation, 2011, pp. 513–518, https://doi.org/10.1109/IV.2011.83

[27] A. Banerjee, P. Maji, A spatially constrained probabilistic model for robust image segmentation, IEEE Trans. Image Process. 29 (2020) 4898–4910, https://doi.org/10.1109/TIP.2020.2975717

[28] L. Griem, A. Koeppe, A. Greß, T. Feser, B. Nestler, Synthetic training data for CT image segmentation of microstructures, Acta Mater. 296 (2025) 121220, https://doi.org/10.1016/j.actamat.2025.121220

[29] J. Hötzer, A. Reiter, H. Hierl, P. Steinmetz, M. Selzer, B. Nestler, The parallel multi-physics phase-field framework PACE3D, J. Comput. Sci. 26 (2018) 1–12, https://doi.org/10.1016/j.jocs.2018.02.011

[30] A.P. Zijdenbos, B.M. Dawant, R.A. Margolin, A.C. Palmer, Morphometric analysis of white matter lesions in MR images: method and validation, IEEE Trans. Med. Imaging 13 (1994) 716–724, https://doi.org/10.1109/42.363096

[31] Y. Nan, P. Tang, G. Zhang, C. Zeng, Z. Liu, Z. Gao, et al., Unsupervised tissue segmentation via deep constrained Gaussian network, IEEE Trans. Med. Imaging 41 (2022) 3799–3811, https://doi.org/10.1109/TMI.2022.3195123

[32] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11 (2010) 625–660, https://doi.org/10.1145/1756006.1756025

[33] M. Caron, P. Bojanowski, J. Mairal, A. Joulin, Unsupervised pre-training of image features on non-curated data, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2959–2968, https://doi.org/10.1109/ICCV.2019.00305

[34] A. Kalapos, B. Gyires-Tóth, Self-supervised pretraining for 2D medical image segmentation, in: L. Karlinsky, T. Michaeli, K. Nishino (Eds.), Computer Vision – ECCV 2022 Workshops, Springer Nature Switzerland, Cham, 2023, pp. 472–484, https://doi.org/10.1007/978-3-031-25082-8_31

[35] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, T. Raiko, Semi-supervised learning with ladder networks, in: Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2, Volume 2 of NIPS'15, MIT Press, Cambridge, MA, USA, 2015, pp. 3546–3554, https://doi.org/10.48550/arXiv.1507.02672