Contents lists available at ScienceDirect

Acta Materialia

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

Full length article

Synthetic training data for CT image segmentation of microstructures

Lars Griem [a],*, Arnd Koeppe [a], Alexander Greß [b], Thomas Feser [b], Britta Nestler [a,c]

[a] Institute for Applied Materials (IAM-MMS), Karlsruhe Institute of Technology (KIT), Straße am Forum 7, 76131 Karlsruhe, Germany
[b] Institute for Vehicle Concepts (FK-WVG), German Aerospace Center (DLR), Pfaffenwaldring 38-40, 70569 Stuttgart, Germany
[c] Institute for Digital Materials Science (IDM), Karlsruhe University of Applied Sciences, Moltkestraße 30, 76133 Karlsruhe, Germany

ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.13982736

Keywords:
Segmentation
Synthetic training data
Generative adversarial networks

ABSTRACT

The segmentation of images obtained through techniques such as computed tomography is a key step in generating digital twins of porous microstructures. A common approach to segmentation is the use of supervised machine learning algorithms, such as U-Net. The training data required for such algorithms are usually obtained by manual labelling, which is extremely time consuming and often inaccurate. We present a method for synthesising realistic training data for segmentation algorithms. This method generates the data in a two-step process that iteratively improves the quality of the synthesised training data. Finally, we validate the similarity between synthetic and real data using quantitative and qualitative metrics and further demonstrate the effectiveness of the synthetic data by experimentally validating segmentation results against measured material properties.

1. Introduction

Computer-aided materials investigation is a common approach to accelerate materials development. These investigations rely on digital material twins used for simulations and experiments. One approach to generating such digital twins is reconstruction from imaging techniques such as computed tomography (CT). Typically, the first step in this reconstruction is to segment the images into foreground and background. A common practice for segmentation is using histogram metrics to determine a global threshold [1–4]. These algorithms aim to find a threshold value in the image's histographic description, resorting to different metrics to determine whether a pixel belongs to the background or foreground. These techniques often lack accuracy in noisy images and images without a distinct separation of the histogram into two or more peaks. This is because some foreground pixels share the same values as background pixels, resulting in faulty segmentations (Fig. 1).

Therefore, more sophisticated segmentation techniques and algorithms such as variable thresholding [6–9], region growing [10,11] and level set methods [12,13] have been developed and applied. More recently, the portfolio of available segmentation methods has been extended by machine learning techniques, of which Minaee at al. [14] and Seo et al. [15] provide comprehensive overviews. One of these methods uses models based on the U-Net architecture [16]. This machine learning model was first introduced in the field of life sciences but has proven to be a reliable segmentation and prediction technique in many other disciplines, such as photogrammetry [17,18], satellite imaging [19], and physical field predictions [20–24]. As a supervised learning algorithm, conventional U-Nets require ground-truth data, i.e., the segmented image for each input image to be trained. This ground-truth data is usually obtained by segmenting images manually, which is exceptionally time-consuming and notoriously inaccurate due to human errors [25–28]. Therefore, the artificial generation of realistic and physically consistent ground-truth data offers a promising approach to accelerate the training process and improve prediction performance [29–34]. When employing artificial data, ensuring the reliable performance of the trained network on real data is crucial. Validating this transferability typically involves comparing the geometric properties of the physical sample – such as porosity or surface area – with those of its digital twin [35–37]. In cases where the (geometric) properties of the physical sample are unknown, validation may become impossible.

To address this challenge, we propose a novel two-step approach for generating artificial training data suitable for microstructure segmentation tasks. Our method combines algorithmic and machine learning techniques to gradually enhance the similarity of the synthetic data to real-world examples. In our use case, this corresponds to CT images of foam structures. To assess the quality of the generated data, we compare synthetic and real images using image embeddings derived from a pretrained encoder network. Furthermore, we evaluate the segmentation accuracy of neural networks trained solely on synthetic data

* Corresponding author.
E-mail address: lars.griem@kit.edu (L. Griem).

https://doi.org/10.1016/j.actamat.2025.121220
Received 16 December 2024; Received in revised form 13 May 2025; Accepted 4 June 2025
Available online 18 June 2025

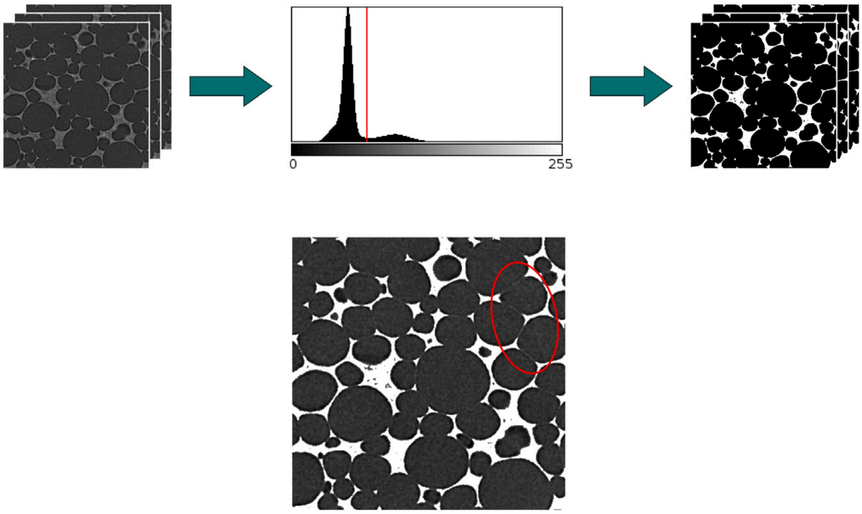

Fig. 1. Histogram-based Otsu [5] segmentation of grayscale images. The chosen threshold (centre, red vertical line) binarises the input images (left), resulting in the binary images (right). Overlaying the binary images onto the input images (bottom) visualises the mis-segmentation of some pixels, particularly the erroneous segmented area at the top right of the image (bottom, red oval). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

by comparing their predictions to experimentally measured material properties. This dual validation strategy confirms that segmentation accuracy obtained on synthetic data can be reliably transferred to real data, effectively mitigating the lack of labelled physical samples and supporting model screening in low-data regimes.

2. Methods

2.1. U-Net architecture

The U-Net is a machine learning architecture originally developed for biomedical image segmentation [16] and is based on a Fully Convolutional Network [38]. The basic concept of the U-Net is the sequential use of an encoder and a decoder, which are interconnected via skip connections. The encoder of the network reduces the size of the input images while increasing their feature space, capturing the context of the images. This is achieved by convolving the images using multiple convolution layers and then reducing their size using a pooling layer. The combination of convolution and pooling is called a level in the U-Net. Depending on the number of levels, this process is repeated several times. After passing through the encoder, the images are expanded back to their original size in the decoder. This is implemented by a series of up-convolution and convolutional layers symmetric with the encoder. The skip connections between the encoder and the decoder are essential parts of the U-Net. These connections concatenate the output of each encoder level to the input of the corresponding decoder level. This helps to locate the extracted features within the image as it adds spatial information of the image to the decoder (Fig. 2). To use the U-Net for binary image segmentation, a final convolutional layer in the network with a softmax or sigmoid activation function transforms the output to unitary values.

2.2. Generative adversarial network

Generative Adversarial Networks (GANs) provide a useful framework in machine learning, particularly in generative modelling [39–41]. Introduced by Goodfellow et al. in 2014 [42], GANs consist of two neural networks — a generator and a discriminator — that engage in a competitive training process. The generator synthesises data samples from random noise, aiming to produce outputs resembling real data, while the discriminator attempts to discriminate between real and synthetic samples. Through iterative adversarial training, GANs gradually improve and converge to an equilibrium where the generator produces highly realistic samples. This adversarial setup enables GANs to capture complex data distributions and generate high-quality synthetic data. A schematic illustration of the GAN architecture is shown in Fig. 3.

2.3. CycleGAN

CycleGAN was introduced in 2017 by Zhu et al. [43] and presents a powerful algorithm for unpaired image-to-image translation. In the field of image segmentation, it has proven to be a reliable method for data augmentation [44–46].

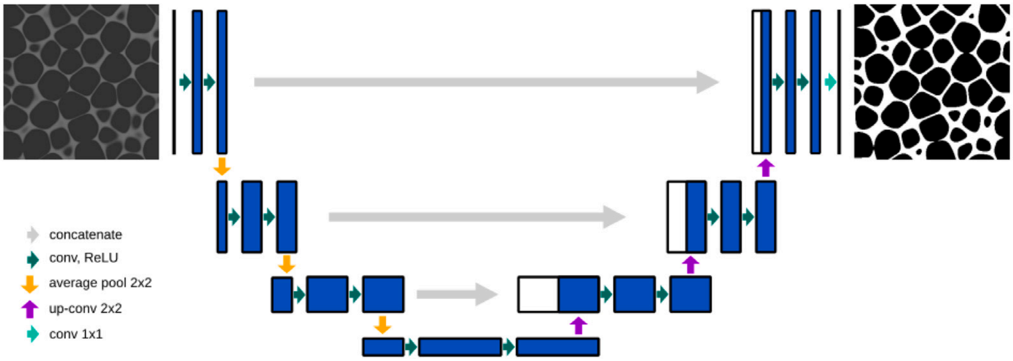

Fig. 2. Qualitative scheme of the U-Net architecture.

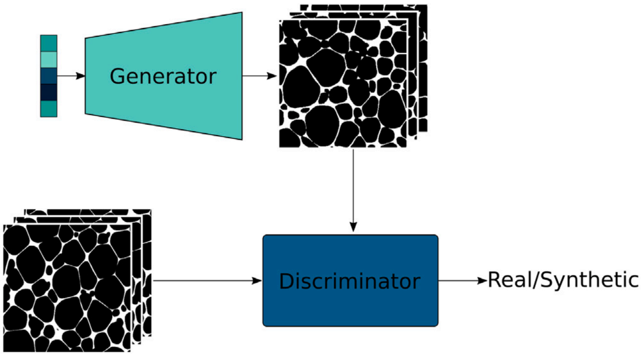

Fig. 3. Schematic illustration of the GAN architecture. The generator creates new synthetic datasets seeking to convince the second model, the discriminator, that the synthetic data is real. The discriminator learns to differentiate between real and synthetic images. Training these networks adversarially gradually improves the generator and discriminator, resulting in a generator capable of creating highly realistic samples.

Fig. 4 shows a schematic overview of the CycleGAN architecture. A CycleGAN consists of two GAN networks G and F. These networks learn the mapping $G : X \mapsto Y$ and $F : Y \mapsto X$ such that the distribution of images from $G(X)$ is indistinguishable from Y and $F(Y)$ from X. To facilitate training, the networks are constrained through a cycle-consistency loss which enforces that $F(G(X)) \approx X$ and $G(F(Y)) \approx Y$. [43] For more detailed information on CycleGANs, see Zhu et al. [43].

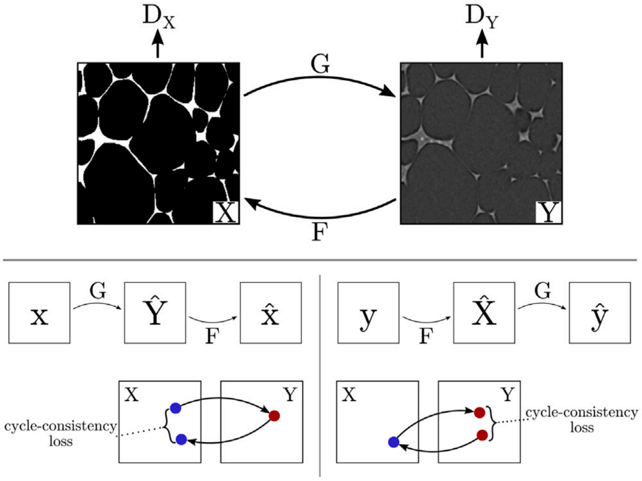

Fig. 4. CycleGAN architecture. *Source:* Adapted from Zhu et al. [43].

2.4. Pycnometry

Liquid pycnometry, as described in EN ISO 1183-1 [47], is an experimental technique for determining a material's density by weighing. The procedure uses a pycnometer, a glass container of known volume. First, the empty pycnometer is weighed. It is then filled with the immersion liquid and weighed again. Next, a defined amount of the sample material is placed into the pycnometer, which is subsequently filled with the immersion liquid and weighed once more. From these measurements, the density of the material can be calculated according to $\rho_S = (m_S \times \rho_{IL})/(m_1 - m_2)$,where m_S is the mass of the specimen, ρ_{IL} is the density of the immersion liquid, m_1 is the mass of the immersion liquid required to fill the empty pycnometer and m_2 is the mass of the immersion liquid required to fill the pycnometer containing the specimen. In this work ethanol is used as immersion liquid which has a density of $0.81\,\mathrm{g/cm^3}$.

3. Data generation

Our method for generating realistic training data consists of two steps that iteratively refine the data quality. Each step generates synthetic data and trains a U-Net for segmentation. Since a U-Net relies on supervised learning, the generated datasets always consist of grayscale images as input and their corresponding segmented binary labels as output. In the following, we describe the two steps of our data generation process in detail. The process requires exemplary image data, for which training data should be generated. In our case, we use CT scans of a polyurethane foam. The scans are taken from the same foam but at different positions, resulting in slightly different microstructures. We refer to these CT scans as Sample 1, Sample 2, and Sample 3.

3.1. First step: algorithmic

The purpose of the first step is to train a U-Net that offers improved image segmentation, which the second step of our data generation process can build upon. For this, we algorithmically synthesise artificial training data using the workflow shown in Fig. 5.

In this workflow, we first binarise the available CT images of foam structures using a conventional thresholding technique in the form of the Otsu algorithm [5]. From the binarised images, we reconstruct a three-dimensional digital model of the foam structure. Through structural analyses of this digital model, we obtain approximate properties of the real foam, such as the porosity, the pore size distribution, and the mean wall thickness. These analyses are performed using the simulation framework PACE3D [48]. The use of other publicly available tools [49–51] is also possible at this point. Next, based on the obtained structural properties, we algorithmically generate similar foam structures using a generation algorithm [52] available in PACE3D. Other publicly available generation algorithms [50,51,53] can also be used for generating the foam structures. Slicing the generated structures into binary images along their height provides binary image stacks. To derive the corresponding grayscale images, the binary images are turned into 8-bit grayscale images and smoothed using Gaussian blur. The contrast of the 8-bit images is further adjusted, and Gaussian Noise is added to them. These manipulations are all performed in the open-source software Fiji [49]. A macro that performs this conversion from binary to grayscale images is publicly available [54]. At this point, the first synthetic dataset has been successfully created and is available for further use. Exemplary input and corresponding target images are shown in step 7 of Fig. 5.

3.1.1. Model training

The first dataset enables improved image segmentation, which the second data generation step can build upon. We train a U-Net on the just-created data. For this, the workflow described above generated 12 000 input and output images of different foam structures. To obtain an information-rich database, the structural properties of the generated structures vary around the properties measured in the binarised CT images. A 70-20-10 data split creates the training, validation, and test sets. Next, a Hyperband search followed by a more targeted Bayesian hyperparameter optimisation determined the optimal parameters for the U-Net architecture. Within this optimisation, the number of levels, kernels, convolutions, learning rate, and kernel size were varied. The search resulted in an architecture of 5 levels, 2 convolutions per level, average pooling, and a kernel size of 4. For training the model, the Dice Loss mitigates the negative effects of unbalanced distribution between foreground and background pixels on model performance, as suggested by Milletari et al. [55]. Additionally, preprocessing the input images by adding Gaussian noise with a standard deviation of 0.05 during training imitates the noise visible in real CT images. Training a U-Net model with these parameters on the first dataset created a network with an accuracy of 99.6% on the test data.

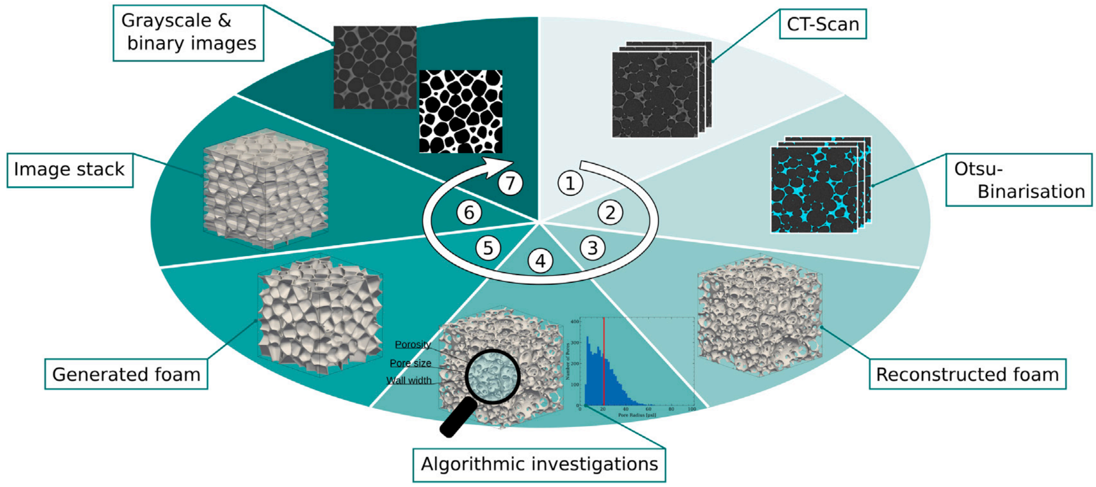

Fig. 5. The first step of the data generation process. CT scans (1) are binarised with Otsu thresholding (2), reconstructed (3), and analysed statistically and physically (4). From the statistical and physical properties, artificial foam structures are synthesised (5), sliced into stacked images (6), and processed to resemble CT scans with known ground truth (7).

3.2. Second step: data-driven

While the U-Net already performs well on algorithmically generated training data, it remains challenging to make quantifiable statements about its performance on real CT data. This is because the first training dataset may not capture all the details of real CT data and thus may not accurately mimic them. We therefore introduce the second step of our generation approach, as illustrated in Fig. 6. In this step, data-driven algorithms improve the quality of the synthetic data, pushing it closer to real data. The generation of this dataset follows a similar procedure as the one shown in Fig. 5. First, the U-Net trained in the first step of our developed method binarises the available CT images. Next, the binarised images are used to train a Generative Adversarial Network (GAN) to synthesise realistic binary images. After editing the GAN-synthesised binary images using the watershed algorithm, a CycleGAN processes them to obtain CT-like grayscale images. This results in a synthetic training dataset consisting of realistic grayscale images and their binary target counterparts. In the following, we describe the artificial data generation and the noise addition using GANs in more detail.

3.2.1. Artificial binary image generation

Training a GAN to generate realistic binary images of foam structures requires real data that the GAN should mimic. For this, the binarised CT images segmented by the U-Net trained in the first step are used (Section 3.1).

After training the GAN on this data, the generator of the GAN can produce realistic binary foam structures. It is important to note that the GAN was trained using CT data segmented by the first U-Net. Therefore, the generator of the GAN can only synthesise images similar to the output of this U-Net. Any structural features in the CT data not identified by the U-Net will not be present in the generated images.

To identify such possible features, we manually compare the segmented CT images to their original grayscale counterparts. This reveals that extraordinarily thin pore walls are not identified by the U-Net and are missing in the binary images. To ensure that these structural features are present in the new dataset, the GAN-generated binary images were post-processed using the Watershed algorithm [56]. This algorithm automatically identifies the pores in a binary image and separates them from each other by adding thin walls to the image. In this way, realistic closed-cell foam structures with filigree structural features intact are obtained. To avoid biasing the dataset by adding this structural information, only half of all GAN-generated images are post-processed in this way. Fig. 7 shows two exemplary GAN synthesised structures, one without and one with added structural information. The influence of this added structural information on the segmentation accuracy of a trained network is investigated in Section 4.

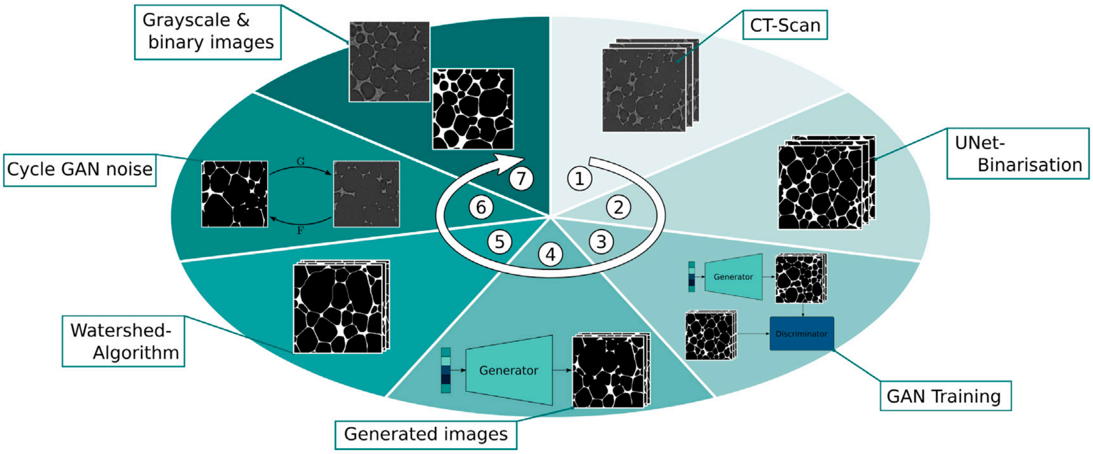

Fig. 6. The second step of the data generation process. CT scans (1) are binarised using the U-Net trained in the previous step of our method (2). A GAN is then trained on the binarised CT images to generate realistic binary foam structures (3). The binary images obtained from the GAN generator (4) are processed using the Watershed algorithm to add missing structural features (5). Using a CycleGAN trained on real CT data (6), the binary images are turned into grayscale images with known binary targets (7).

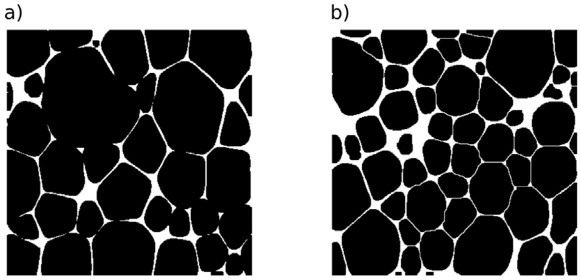

Fig. 7. Images generated by the GAN generator. (a) An exemplary output of the GAN generator. (b) A microstructure post-processed using the watershed algorithm to add thin structural elements.

3.2.2. Artificial noise addition

To complement the enhanced synthetic training data, the grayscale images corresponding to the GAN-generated binary images are created. A CycleGAN architecture generates these grayscale images by applying realistic noise to the previously synthesised binary images. As described in Section 2.3 this architecture learns unpaired image translation and consists of two GAN models.

In our use case, this architecture (see Fig. 4) translates into the following CycleGAN components. The generator G learns to map binary images (X) to grayscale images (Y), while the discriminator Dy learns to distinguish between real and fake grayscale images. The generator F, on the other hand, translates grayscale images (Y) into binary images (X) and the discriminator Dx discriminates between real and false binary images. The use of a cycle-consistency loss further ensures that a binary image X, which has been translated by G into grayscale and then back into binary by F, equals the original binary image X.

As CycleGANs can be trained on unpaired data, our network is trained using the binary images synthesised by the GAN described in Section 3.2.1 and the real grayscale CT images. Within the CycleGAN itself, the U-Net architecture is used for both generators. Training the CycleGAN creates a generator G that can augment binary images to resemble grayscale CT images. Applying this generator to the GAN synthesised binary images, finalises the second synthetic dataset and completes the two-step data generation process.

In Fig. 8, we compare a grayscale synthetic image of the just created dataset to a real CT image. To the human eye, the synthesised grayscale image is almost indistinguishable from the real data, suggesting a close resemblance between our training and real data.

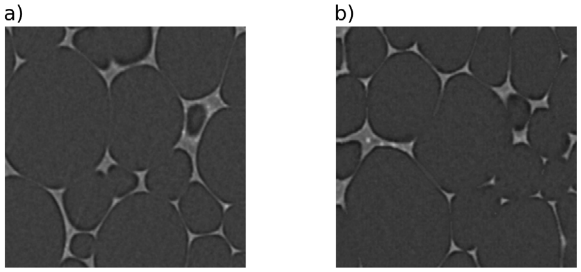

Fig. 8. CT and synthetic images for comparison. (a) An original CT image. (b) A synthetic grayscale image generated by applying noise to a GAN-generated binary image using our CycleGAN.

4. Results

The described generation process can synthesise highly realistic training data for images segmentation tasks. To validate the subjective impression of similarity between real and synthetic images, we conducted both data-driven and experimental evaluations.

4.1. Data driven investigation

To quantitatively assess the similarity between real and synthetic images, we employed the *Fréchet Inception Distance* (FID), a widely used metric for the evaluation of image generation models. FID computes the *Fréchet* distance between two multivariate Gaussians fitted to feature representations extracted from image datasets using a pretrained encoder network.

In our study, we utilised a *ResNet50* network trained on the *MicroNet* [57] dataset, a publicly available collection of grayscale material images acquired through methods such as *Scanning Electron Microscopy* or *Computed Tomograpy*. As this dataset closely resembles our CT data in structure and modality, the resulting feature representations are expected to be more relevant than those extracted by the commonly used *InceptionV3* model pretrained on *ImageNet*.

To extract features from each image, we removed the final fully connected layer of the *ResNet50* model. This yielded 2048-dimensional feature vectors, which were used to compute the FID scores. As our encoder differs from the standard *InceptionV3*, existing reference FID values cannot be directly applied. To establish a meaningful baseline, we computed the FID between the different CT scans that we acquired from different subvolumes of the same foam structure. These intra-CT comparisons serve as a reference for evaluating the similarity between real and synthetic images. For each of these CT scans a synthetic dataset was generated. Table 1 shows the measured FID between the real and synthetic datasets including the data generated in the first step of the generation process. In the table we refer to the data generated in the second step as GAN and to the data from the first step as algorithmic.

Table 1
Fréchet Inception Distance between the real and synthetic datasets.

	Sample 1	Sample 2	Sample 3	GAN	Algorithmic
Sample 1	0	18.72	37.97	3.85	99.69
Sample 2	18.72	0	13.62	2.61	78.10
Sample 3	37.97	13.62	0	4.47	51.09

The baseline intra-CT FID scores range from 13.62 to 37.97. In contrast, the FID scores between real and GAN-generated datasets are significantly lower, ranging from 2.61 to 4.47. This indicates that the synthetic data closely mimics the statistical properties of the real datasets. Furthermore, a clear improvement in realism is observed when comparing GAN-generated images to those produced by a traditional algorithmic approach. To assess the impact of post-processing, we investigated the effect of varying the proportion of images enhanced with watershed segmentation on the resulting FID scores. Fig. 9 shows that this post-processing step slightly affects the FID, with its influence varying across datasets, sometimes improving and sometimes worsening similarity.

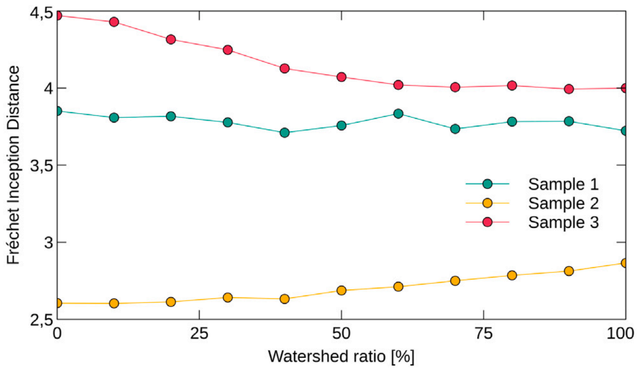

Fig. 9. Influence of watershed post processing on the image similarity quantified using *Fréchet Inception Distance*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Therefore, the decision to apply post-processing should be made on a case-by-case basis, depending on the characteristics of the dataset under investigation. Whether these FID variations translate into differences in segmentation performance is examined experimentally in Section 4.2.

To further evaluate how well the synthetic data captures the distribution of real CT data, we applied the *t-Distributed Stochastic Neighbour Embedding* (t-SNE) algorithm [58]. This unsupervised machine learning algorithm reduces the dimensionality of a dataset in a non-linear manner while maintaining data point similarities. To achieve this, it employs a distance metric to identify the nearest neighbours of each data point to then map the high-dimensional data points onto a lower-dimensional space while preserving the identified neighbourhood relations. For our dataset, we used the *cosine* distance metric to measure the data point similarity. Prior to applying t-SNE, we reduced the dimensionality of the image data by extracting latent embeddings using the *ResNet50* model pretrained on the *MicroNet* dataset, as described earlier. The proportion of watershed-processed images included in the analysis was chosen based on the configuration that yielded the lowest FID score, as shown in Fig. 9. The resulting image embeddings were visualised using t-SNE, and the output is shown in Fig. 10. Each dot represents a grayscale image. The different CT datasets and their respective synthesised image data are visualised in different colours. As illustrated in Fig. 10, the encoder network is able to distinguish between the different CT datasets (blue, orange, green), which form well-separated clusters. Notably, the GAN-generated synthetic images (purple, brown, pink) closely cluster with the real CT data they are designed to mimic, indicating a strong overlap in the feature space. This suggests that the synthetic data effectively cover the full distribution of the corresponding real datasets. In contrast, the algorithmic images (red dots) form clearly separated clusters, further underscoring the realism achieved by the GAN-based generation approach.

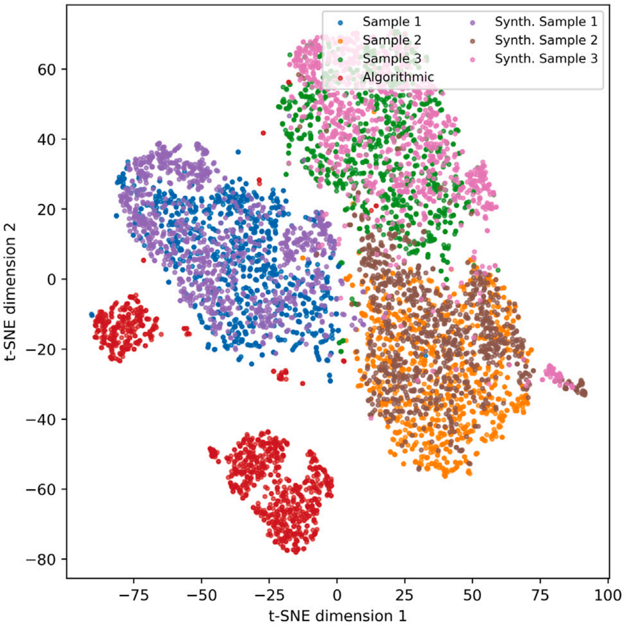

Fig. 10. T-SNE reduced visualisation of the image embeddings. GAN-generated images closely cluster with their corresponding CT datasets, while the algorithmic images of the first generation step form distinct, separate clusters. Cosine distance was used to compute similarity. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

These data-driven investigations confirm that our generation pipeline produces highly realistic image data that closely align with the statistical and structural characteristics of real CT scans.

Next, we present experimental evaluations to assess whether this realism translates into improved segmentation accuracy when training networks on the synthesised data.

4.2. Experimental investigation

To quantify the segmentation accuracy of a U-Net trained on synthetic datasets, we compared the porosity values derived from the segmented CT images with experimentally measured porosities of the corresponding foam structures. The porosities were determined by first measuring the effective density ρ_{eff} of each sample. The effective density ρ_{eff} is related to the material porosity ϵ and the density of the polyurethane matrix ρ_{PUR} via $\rho_{eff} = \rho_{PUR} \cdot (1 - \epsilon)$ which can be rearranged to express porosity as $\epsilon = 1 - \rho_{eff}/\rho_{PUR}$.

Accordingly, accurate porosity estimation requires knowledge of the polyurethane density ρ_{PUR}. This was determined experimentally via pycnometry, as described in Section 2.4. In our measurements, we used ethanol as immersion liquid, which has a density of $0.81\,\mathrm{g/cm^3}$. To eliminate air entrapment and ensure accurate volume displacement, the foam structure was ground to a coarse granulate. As defined in EN ISO 1183-1 three separate specimens were measured and the mean across all measurements was calculated. Table 2 shows the results of these measurements. The density of the polyurethane matrix was experimentally determined to be $1.411\,\mathrm{g/cm^3}$.

Table 2
Pycnometry results. Three separate measurements were conducted, and the mean between them was calculated.

	Unit	Measurement 1	Measurement 2	Measurement 3	Mean value
m_S	[g]	0.2013	0.1964	0.2066	
m_1	[g]	7.0893	7.0933	7.0924	
m_2	[g]	6.9758	6.9789	6.9734	
ρ_{PUR}	[g/cm^3]	1.437	1.391	1.406	**1.411**

Next, we measured the effective density of each sample to be able to calculate their porosities. This involved first determining the dimensions of the CT-scanned specimens, followed by weighing them with a precision scale with an accuracy of $0.01\,\mathrm{g}$. The resulting effective densities and corresponding porosities are summarised in Table 3.

Table 3
Effective density and porosity of each sample. By measuring the effective density of each sample, its porosity can be calculated.

	Unit	Sample 1	Sample 2	Sample 3
ρ_{eff}	[kg/m^3]	154.93	197.72	248.34
ϵ	[%]	89.02	85.99	82.40

These experimentally determined porosities serve as reference values for evaluating the segmentation performance of U-Net models trained on synthetic data. For each sample, U-Nets were trained using the respective synthetic datasets as well as algorithmically generated data from the first generation step (see Section 3.1). To investigate the effect of watershed-based post-processing on segmentation accuracy, we varied the ratio of watershed processed data in the training sets. The trained networks were then applied to the available CT data, and the porosities computed from the segmented volumes were compared to the experimentally determined reference values. Fig. 11 shows the results of this validation. As shown in Fig. 11, U-Nets trained on GAN-generated data consistently outperformed those trained on algorithmic data. In all datasets, applying watershed post-processing improved segmentation accuracy. To better visualise these differences, Fig. 12 shows example segmentation results. As evident from the figure, the U-Net trained on GAN-generated data is better at capturing fine structural features and more accurately preserves thin cell walls. In contrast, the model trained on the algorithmic data of step 1 tends to overestimate wall thicknesses, resulting in underestimation of porosity. This observation aligns with the quantitative findings and further supports the effectiveness of the GAN-based synthetic data.

5. Conclusion

In this work, we have introduced a method that creates synthetic training data tailored to improve supervised machine learning segmentation algorithms. Utilising generative AI in the form of Generative

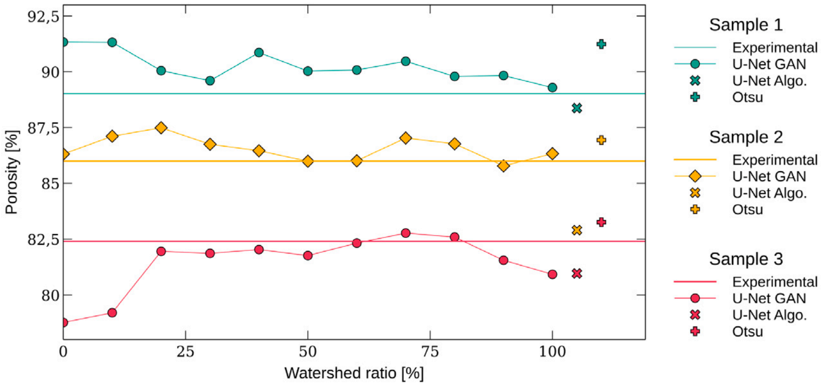

Fig. 11. Influence of watershed post processing on segmentation accuracy. For each sample, watershed processed images improve the achieved segmentation accuracy.

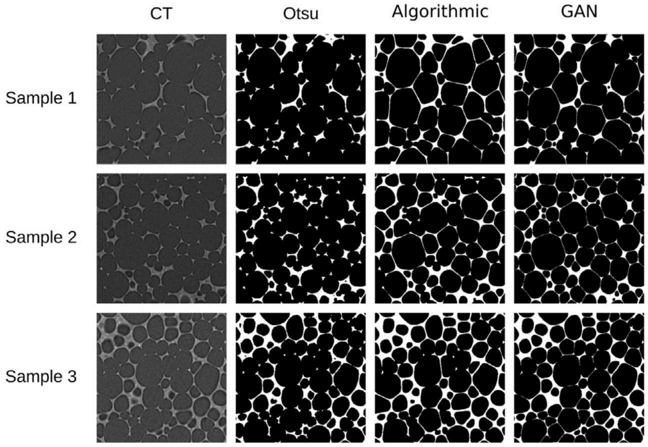

Fig. 12. Influence of watershed post processing on the segmentation of the available CT images. For each dataset, results from Otsu thresholding, a U-Net trained on algorithmic (step 1) data, and a U-Net trained on GAN-enhanced (step 2) data are shown.

Adversarial Networks (GAN), our approach yields synthetic data that closely mimics experimental data with remarkable similarity.

This resemblance was quantitatively evaluated using low-dimensional representations, including the Fréchet Inception Distance (FID) and t-SNE visualisations, both of which confirmed a strong similarity between synthetic and real datasets. Further experiments compared the segmentation results of models trained on our synthetic data with the actual structural properties of the foams. The resulting segmentations demonstrated high accuracy, supporting the effectiveness of our synthetic data in real-world scenarios. These findings suggest that segmentation models trained on our generated data can generalise well to real data, making this approach especially useful for model screening in the low-volume data problem. Central to our approach is the two-step, progressively refined generation of synthetic data. In the first step, we algorithmically generate a training dataset and train a segmentation network. In the second step, we build upon this trained segmentation network to binarise the available real images. On this binarised data, we train generative AI models to synthesise an improved training dataset. This two-step approach makes our method generic and easily adjustable to other use cases. To adapt to different scenarios, only the generation algorithm of the first step needs to be replaced by a suitable algorithm. For this, many different open-source libraries are available that provide a great variety of generation algorithms that cater to arbitrary use cases [50,51,53].

In summary, our method provides a broad approach capable of addressing the challenge of inadequate or inaccurate training and validation datasets, thus offering a promising solution to this pervasive problem in machine learning research.

CRediT authorship contribution statement

Lars Griem: Writing – original draft, Methodology, Conceptualization. **Arnd Koeppe:** Writing – review & editing, Conceptualization. **Alexander Greß:** Writing – review & editing, Experimental investigation. **Thomas Feser:** Writing – review & editing, Experimental investigation. **Britta Nestler:** Writing – review & editing, Supervision, Funding acquisition.

Code availability

The code to generate the binary and grayscale microstructural images and to train the U-Net has been made available at https://doi.org/10.5281/zenodo.13982736.

Declaration of competing interest

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

Acknowledgements

This work was partly carried out with the support of the Joint Laboratory Model and Data-driven Materials Characterisation (JL MDMC), a cross-centre platform of the Helmholtz Association and with the support of the Karlsruhe Nano Micro Facility, Germany (KNMFi, www.knmf.kit.edu), a Helmholtz Research Infrastructure at Karlsruhe Institute of Technology, Germany (KIT, www.kit.edu). The authors gratefully acknowledge additional support from the Helmholtz Association through programme 43.31.01 and the Helmholtz Artificial Intelligence Cooperation Unit (Helmholtz-AI), funded by the Initiative and Networking Fund (IVF) of the Helmholtz Association (grant number: ZT-I-PF-5-075). Part of the model formulation was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Germany, under project ID 390874152, POLiS Cluster of Excellence (grant number: UP 33/1). This work contributes to the research performed at CELEST (Center for Electrochemical Energy Storage Ulm-Karlsruhe).

Data availability

The used and generated image datasets are publicly available at https://doi.org/10.5281/zenodo.13982736.

References

[1] T. Ridler, S. Calvard, et al., Picture thresholding using an iterative selection method, IEEE Trans. Syst. Man Cybern. 8 (8) (1978) 630–632.

[2] A. Rosenfeld, P. De La Torre, Histogram concavity analysis as an aid in threshold selection, IEEE Trans. Syst. Man Cybern. (2) (1983) 231–235.

[3] J.N. Kapur, P.K. Sahoo, A.K. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Comput. Vis. Graph. Image Process. 29 (3) (1985) 273–285.

[4] N. Ramesh, J.-H. Yoo, I. Sethi, Thresholding based on Histogram approximation, IEE Proc., Vis. Image Signal Process. 142 (5) (1995) 271–279.

[5] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.

[6] S.D. Yanowitz, A.M. Bruckstein, A new method for image segmentation, Comput. Vis. Graph. Image Process. 46 (1) (1989) 82–95.

[7] A.P. Sheppard, R.M. Sok, H. Averdunk, Techniques for image enhancement and segmentation of tomographic images of porous materials, Phys. A 339 (1–2) (2004) 145–151.

[8] K.V. Mardia, T. Hainsworth, A spatial thresholding method for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 10 (6) (1988) 919–927.

[9] W. Oh, B. Lindquist, Image thresholding by indicator kriging, IEEE Trans. Pattern Anal. Mach. Intell. 21 (7) (1999) 590–602.

[10] T. Pavlidis, Y.-T. Liow, Integrating region growing and edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 12 (3) (1990) 225–233.

[11] G. Sivewright, P. Elliott, Interactive region and volume growing for segmenting volumes in MR and CT images, Med. Inform. 19 (1) (1994) 71–80.

[12] J.A. Sethian, et al., Level Set Methods and Fast Marching Methods, Vol. 98, Cambridge UP, Cambridge, 1999.

[13] S. Osher, N. Paragios, Geometric Level Set Methods in Imaging, Vision, and Graphics, Springer Science & Business Media, 2003.

[14] S. Minaee, Y.Y. Boykov, F. Porikli, A.J. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[15] H. Seo, M. Badiei Khuzani, V. Vasudevan, C. Huang, H. Ren, R. Xiao, X. Jia, L. Xing, Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications, Med. Phys. 47 (5) (2020) e148–e167.

[16] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[17] A. Marques, G. Racolte, E.M. De Souza, H.V. Domingos, R.K. Horota, J.G. Motta, D.C. Zanotta, C.L. Cazarin, L. Gonzaga, M.R. Veronez, Deep learning application for fracture segmentation over outcrop images from UAV-based digital photogrammetry, in: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, IEEE, 2021, pp. 4692–4695.

[18] E.B. Alexakis, C. Armenakis, Evaluation of UNet and UNet++ architectures in high resolution image change detection applications, Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. 43 (2020) 1507–1514.

[19] J. McGlinchy, B. Johnson, B. Muller, M. Joseph, J. Diaz, Application of UNet fully convolutional neural network to impervious surface segmentation in urban environment from high resolution satellite imagery, in: IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2019, pp. 3915–3918.

[20] A. Koeppe, F. Bamer, B. Markert, An intelligent meta-element for linear elastic continua, PAMM 18 (1) (2018) e201800283, http://dx.doi.org/10.1002/pamm.201800283.

[21] A. Koeppe, F. Bamer, B. Markert, An intelligent nonlinear meta element for elastoplastic continua: Deep learning using a new Time-distributed Residual U-Net architecture, Comput. Methods Appl. Mech. Engrg. 366 (2020) 113088, http://dx.doi.org/10.1016/j.cma.2020.113088.

[22] A. Koeppe, F. Bamer, B. Markert, An artificial intelligence approach to model nonlinear continua by intelligent meta-elements, PAMM 20 (1) (2021) e202000300, http://dx.doi.org/10.1002/pamm.202000300.

[23] A. Koeppe, D. Hesser, M. Mundt, F. Bamer, M. Selzer, B. Markert, Mechanics 4.0. Artificial Intelligence for the analysis of mechanical systems, in: W. Frenz (Ed.), Handbook Industry 4.0: Law, Technology, Society, Springer, 2022, pp. 455–470, http://dx.doi.org/10.1007/978-3-662-64448-5_23.

[24] A.B. Farimani, J. Gomes, V.S. Pande, Deep Learning the Physics of Transport Phenomena, 2017, arXiv:1709.02432, URL http://arxiv.org/abs/1709.02432, ArXiv170902432 Phys..

[25] Y. Hirabayashi, H. Iga, H. Ogawa, S. Tokuta, Y. Shimada, A. Yamamoto, Deep learning for three-dimensional segmentation of electron microscopy images of complex ceramic materials, Npj Comput. Mater. 10 (1) (2024) 46.

[26] A. Wijaya, J. Wagner, B. Sartory, R. Brunner, Analyzing microstructure relationships in porous copper using a multi-method machine learning-based approach, Commun. Mater. 5 (1) (2024) 59.

[27] T. Strohmann, K. Bugelnig, E. Breitbarth, F. Wilde, T. Steffens, H. Germann, G. Requena, Semantic segmentation of synchrotron tomography of multiphase Al-Si alloys using a convolutional neural network with a Pixel-wise weighted loss function, Sci. Rep. 9 (1) (2019) 19611.

[28] B. Ma, X. Wei, C. Liu, X. Ban, H. Huang, H. Wang, W. Xue, S. Wu, M. Gao, Q. Shen, et al., Data augmentation in microscopic images for material data mining, Npj Comput. Mater. 6 (1) (2020) 125.

[29] J. Yeom, T. Stan, S. Hong, P.W. Voorhees, Segmentation of experimental datasets via convolutional neural networks trained on phase field simulations, Acta Mater. 214 (2021) 116990.

[30] C. Fend, A. Moghiseh, C. Redenbach, K. Schladitz, Reconstruction of highly porous structures from FIB-SEM using a deep neural network trained on synthetic images, J. Microsc. 281 (1) (2021) 16–27.

[31] Z. Su, E. Decencière, T.-T. Nguyen, K. El-Amiry, V. De Andrade, A.A. Franco, A. Demortière, Artificial neural network approach for multiphase segmentation of battery electrode nano-CT images, Npj Comput. Mater. 8 (1) (2022) 30.

[32] M. Ragone, R. Shahabazian-Yassar, F. Mashayek, V. Yurkiv, Deep learning modeling in microscopy imaging: A review of materials science applications, Prog. Mater. Sci. (2023) 101165.

[33] S. Müller, C. Sauter, R. Shunmugasundaram, N. Wenzler, V. De Andrade, F. De Carlo, E. Konukoglu, V. Wood, Deep learning-based segmentation of Lithium-Ion battery microstructures enhanced by artificially generated electrodes, Nat. Commun. 12 (1) (2021) 6205.

[34] L. Rangel DaCosta, K. Sytwu, C. Groschner, M. Scott, A robust synthetic data generation framework for machine learning in high-resolution transmission electron microscopy (HRTEM), Npj Comput. Mater. 10 (1) (2024) 165.

[35] S. Chun, S. Roy, Y.T. Nguyen, J.B. Choi, H.S. Udaykumar, S.S. Baek, Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials, Sci. Rep. 10 (1) (2020) 13307.

[36] P. Altschuh, Skalenübergreifende analyse Makroporöser membranen im kontext digitaler zwillinge, Karlsruher Inst. Für Technol. (KIT) (2020).

[37] F. Bagherzadeh, J. Freitag, U. Frese, F. Wilhelms, Ice-core micro-CT image segmentation with deep learning and Gaussian mixture model, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–11.

[38] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for Semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651, http://dx.doi.org/10.1109/TPAMI.2016.2572683.

[39] P.C. Nguyen, N.N. Vlassis, B. Bahmani, W. Sun, H. Udaykumar, S.S. Baek, Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning, Sci. Rep. 12 (1) (2022) 9034.

[40] J. Feng, Q. Teng, X. He, X. Wu, Accelerating multi-point statistics reconstruction method for porous media via deep learning, Acta Mater. 159 (2018) 296–308.

[41] Z. Yang, X. Li, L. Catherine Brinson, A.N. Choudhary, W. Chen, A. Agrawal, Microstructural materials design via deep adversarial learning methodology, J. Mech. Des. 140 (11) (2018) 111416.

[42] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014, arXiv:1406.2661.

[43] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.

[44] V. Sandfort, K. Yan, P.J. Pickhardt, R.M. Summers, Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks, Sci. Rep. 9 (1) (2019) 16884.

[45] R. Barth, J. Hemming, E.J. Van Henten, Optimising realism of synthetic images using cycle generative adversarial networks for improved part segmentation, Comput. Electron. Agric. 173 (2020) 105378.

[46] V. Thambawita, P. Salehi, S.A. Sheshkal, S.A. Hicks, H.L. Hammer, S. Parasa, T.d. Lange, P. Halvorsen, M.A. Riegler, SinGAN-Seg: Synthetic training data generation for medical image segmentation, PLoS One 17 (5) (2022) e0267976.

[47] European Committee for Standardization, EN ISO 1183-1:2024 – Plastics - Methods for Determining the Density of Non-Cellular Plastics - Part 1: Immersion Method, Liquid Pycnometer Method and Titration Method , Standard, 2024, European Standard (EN ISO 1183-1:2024).

[48] 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.

[49] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al., Fiji: an open-source platform for biological-image analysis, Nature Methods 9 (7) (2012) 676–682.

[50] J.C. Ferguson, F. Semeraro, J.M. Thornton, F. Panerai, A. Borner, N.N. Mansour, Update 3.0 to “PuMA: The porous microstructure analysis software”,(PII: S2352711018300281), SoftwareX 15 (2021) 100775.

[51] F.L. Usseglio-Viretta, P. Patel, E. Bernhardt, A. Mistry, P.P. Mukherjee, J. Allen, S.J. Cooper, J. Laurencin, K. Smith, MATBOX: An open-source microstructure analysis toolbox for microstructure generation, segmentation, characterization, visualization, correlation, and meshing, SoftwareX 17 (2022) 100915.

[52] P. Altschuh, Y.C. Yabansu, J. Hötzer, M. Selzer, B. Nestler, S.R. Kalidindi, Data science approaches for microstructure quantification and feature identification in porous membranes, J. Membr. Sci. 540 (2017) 88–97.

[53] D. Niblett, M. Mamlouk, O.E.G. Brizuela, S. An, Porous microstructure generator, 2022, http://dx.doi.org/10.25405/data.ncl.20448471.v8, URL https://data.ncl.ac.uk/articles/software/Porous_Microstructure_Generator/20448471.

[54] L.C. Griem, Fiji macro for binary to grayscale image translation, 2024, http://dx.doi.org/10.5281/zenodo.13711248, URL https://doi.org/10.5281/zenodo.13711248.

[55] 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, IEEE, 2016, pp. 565–571.

[56] P. Soille, L.M. Vincent, Determining watersheds in digital pictures via flooding simulations, in: Visual Communications and Image Processing'90: Fifth in a Series, Vol. 1360, SPIE, 1990, pp. 240–250.

[57] J. Stuckner, B. Harder, T.M. Smith, Microstructure segmentation with deep learning encoders pre-trained on a large microscopy dataset, Npj Comput. Mater. 8 (1) (2022) 200.

[58] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).