

A Complete Strategy to Achieve High Precision Automatic Segmentation of Challenging Experimental X-Ray Computed Tomography Data Using Low-Resemblance Synthetic Training Data

Athanasios Tsamos,* Sergei Evsevlev, Rita Fioresi, Francesco Faglioni, and Giovanni Bruno

It is shown that preconditioning of experimental X-ray computed tomography (XCT) data is critical to achieve high-precision segmentation scores. The challenging experimental XCT datasets and deep convolutional neural networks (DCNNs) are used that are trained with low-resemblance synthetic XCT data. The material used is a 6-phase Al–Si metal matrix composite-reinforced with ceramic fibers and particles. To achieve generalization, in our past studies, specific data augmentation techniques were proposed for the synthetic XCT training data. In addition, two toolsets are devised: (1) special 3D DCNN architecture (*3D Triple_UNet*), slicing the experimental XCT data from multiple views (*MultiView Forwarding*), the *i.S.Sy.Da.T.A.* iterative segmentation algorithm, and (2) nonlocal means (NLM) conditioning (filtering) for the experimental XCT data. This results in good segmentation Dice scores across all phases compared to more standard approaches (i.e., standard UNet architecture, single view slicing, standard single training, and NLM conditioning). Herein, the NLM filter is replaced with the deep conditioning framework *BAM SynthCOND* introduced in a previous publication, which can be trained with synthetic XCT data. This leads to a significant segmentation precision increase for all phases. The proposed methods are potentially applicable to other materials and imaging techniques.

tomography (XCT) data,^[2–7] etc. When compared to their 3D counterparts, 2D DCNNs have been the prevalent option because of their lower computational costs, swift convergence, and more importantly, the need for less training data. In recent years, DCNNs have become the standard in XCT segmentation problems of complex materials possessing microstructural phases with similar densities^[6,7] (i.e., similar attenuation coefficients, thus, similar postreconstruction gray levels). The need for such sophisticated tools emerged from the fact that simplistic thresholding,^[6] or Otsu thresholding,^[8] performs poorly in such cases. In the literature, it has been reported that a well-trained DCNN can accelerate the segmentation process by 100 times compared to manual segmentation.^[9] The main problem with this approach is the effort and time required to acquire suitable training data. Customarily, for 2D DCNNs, the necessary training data are generated manually, annotating a few slices from the reconstructed


volumes to be used as training datasets for the DCNN. This approach provides good results in most cases. However, more complex problems (i.e., geometrically more complex material microstructures or more microstructural phases with similar gray levels) require more training data. Thus, the manual annotation can be arduous. Evsevlev et al.^[6] and Tsamos et al.^[7,10] argued that with the presence of many phases with similar gray

1. Introduction

1.1. Related Work

Deep convolutional neural networks (DCNNs) gained popularity in material science for the qualitative and quantitative analysis of material images, such as microscopy data,^[1] X-ray computed

A. Tsamos, S. Evsevlev, G. Bruno
Bundesanstalt für Materialforschung und-prüfung
Unter den Eichen 87, 12205 Berlin, Germany
E-mail: athanasios.tsamos@bam.de

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/adem.202301030>.

© 2023 The Authors. Advanced Engineering Materials published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/adem.202301030

R. Fioresi
Department of Farmacy and Biotechnology (FABIT)
University of Bologna
40126 Bologna, Italy

F. Faglioni
Department of Chemical and Geological Sciences (DSCG)
University of Modena and Reggio Emilia
41125 Modena, Italy

G. Bruno
Institute of Physics and Astronomy
University of Potsdam
Karl-Liebknecht-Str. 24-26, 14476 Potsdam, Germany

levels, the microstructure geometry is decisive for correct feature recognition (i.e., segmentation). Hence, they concluded that a 3D DCNN is better suited to capture the complexity of the microstructure. Unfortunately, manually annotating the required training datasets in 3D is a daunting task.^[7,10] A viable solution is to employ synthetic training data. This approach was used successfully by various researchers,^[4,11–13] but in all cases considered, the problem did not involve highly complex microstructures (i.e., only a couple of phases present with distinct gray levels), and most studies employed a 2D DCNN.

Training an artificial intelligence (AI) model with simulated/synthetic data can be very challenging. There is a well-known performance gap between the AI model-trained employing synthetic training data and tested on real experimental data.^[14,15] When the different microstructural phases within the target dataset carry distinct gray levels, the task becomes less challenging as the AI model can perform the segmentation based on the gray levels. As mentioned earlier though, with similar gray levels the importance of geometry and statistical distributions is elevated for accurate phase recognition. In this case, generating high-resemblance synthetic training data is a task that would require such geometrical and statistical distribution knowledge. This knowledge can be acquired through classic image processing (i.e., watershed, particle analysis, etc.) only if the labels for each pixel/voxel are known as apriori (i.e., already labeled dataset). This would require manual labeling. This approach would become an oxymoron, because such labeled experimental data can be employed for training an AI model in the first place. Generative adversarial networks (GANs) are yet another promising approach for generating high resemblance training data. However, GANs also require manually labeled training data. Lastly, the exact synthetic replication of artifacts and gray-level patterns present within the experimental datasets is another hurdle. Thus, being able to employ low resemblance synthetic training data for successful experimental data segmentation, is compelling.

In ref. [7], Tsamos et al. attempted to automatically segment very challenging XCT datasets of a 6-phase Al–Si metal matrix composite (MMC) reinforced with ceramic fibers and ceramic particles (6-phases: Al₂O₃ fibers, SiC particles, intermetallics

(IMs), eutectic Si, Al matrix, and voids/cracks). The training/validation datasets were synthetically generated with the in-house synthetic materials library *BAM SynthMAT*.^[7] The gray levels were assigned based only on the phase type (fast, low resemblance approach, i.e., no radiographic simulation) on the synthesized microstructures, based on random gray level sampling from the experimental XCT datasets (see **Figure 1**). The main challenge was to achieve good generalization for segmentation from low-resemblance synthetic XCT datasets on experimental XCT datasets (the latter possess more complex gray level and artifacts patterns). This was achieved by introducing a special 3D DCNN: *3D Triple_UNet*,^[7] which is capable of superior generalized training with synthetic training data compared to a standard UNet architecture.^[16] Moreover, special augmentations (brightness, contrast, Gaussian noise, and Gaussian blur) and advanced forwarding strategies (*3D MultiView slicing*) were adopted to further promote generalization.^[7] The automatic segmentation achieved a good overall Dice score, but lower scores were reported for the Al₂O₃ fibers, SiC particles, and IMs, due to their similar gray levels. The latter scores were achieved on experimental Al–Si MMC XCT datasets that were conditioned with a non-local means (NLM) denoising filter^[17,18] before automatic segmentation. For nonconditioned datasets, the achieved Dice scores were even lower.^[7]

In ref. [10], Tsamos et al. proposed an iterative segmentation algorithm (*i.S.Sy.Da.T.A.*) to increase the segmentation Dice scores, employing the same NLM-conditioned experimental XCT datasets and synthetic XCT training data as in ref. [7]. *i.S.Sy.Da.T.A.* utilizes segmentation outputs of additional experimental datasets (statistically equivalent to the target experimental dataset) from previous iterations to improve the segmentation precision.^[10] Compared to the results in ref. [7], this method increased the overall Dice score, as well as the individual scores for Al₂O₃ fibers, SiC particles, and IMs. For nonconditioned experimental XCT datasets, the improvement was inferior but measurable. It was concluded that the number and intensity of artifacts present (i.e., noise, blur, rings) detrimentally affect the final segmentation performance. Thus, it was argued that another more sophisticated XCT data conditioning method should be employed instead of an NLM filter when synthetic

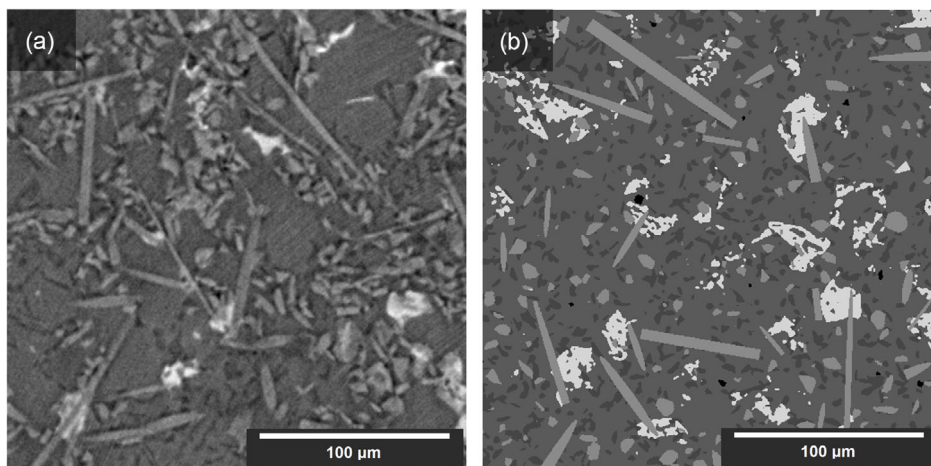


Figure 1. a) Experimental Al–Si MMC XCT reconstruction –versus, b) low resemblance synthetic Al–Si MMC XCT.

training data are used.^[7,10] In this study, we build upon the findings from refs. [7,10] by testing such a conditioning method. More specifically, Tsamos et al.^[19] introduced a multilevel/multiscale XCT data deep conditioning framework: *BAM SynthCOND*, which is capable of denoising, deblurring, and diminishing ring artifacts in experimental XCT datasets. This was accomplished with special in-house DCNNs (*ACEnets*), solely trained with synthetic XCT training data, and with special training strategies.^[19] In this study, we employ the same synthetic Al–Si XCT data from refs. [7,10] but now both for training *BAM SynthCOND* and the *3D Triple_UNet*. In other words, we combine the methods from our three previous publications (*BAM SynthMAT*, *Triple_UNet*, special augmentations, *i.S.Sy.Da.T.A.*, and *BAM SynthCOND*); in doing so, we achieve even higher Dice scores compared to our previous attempts in refs. [7,10] (at least 0.70, regarded as the acceptable threshold for accurate segmentation^[6]). This holds for all microstructural phases of the challenging Al–Si MMC experimental XCT data. A summary of the steps taken is outlined below.

1.2. Material Description

For our investigation, we employed readily available XCT datasets of AlSi12CuMgNi MMCs,^[6,7,10] reinforced with {0% vol Al₂O₃ short fibers, 15% vol SiC particles} and {7% vol Al₂O₃ short fibers, 5% vol SiC particles}. Cast near-eutectic Al–Si alloy materials are very popular in the automotive industry for engine pistons manufacturing.^[20] Also, there is a strong interest growing within the aerospace industry with prospective applications in frames, aerals, and joints, as a promising substitute to the standard unreinforced Al and Ti alloys.^[21]

While transition metals like Cu and Ni promote the creation of dense and stable aluminum IMs, the presence of the Si phase increases the melt fluidity. In the microstructure of the alloy, the IMs and the eutectic Si phase create a 3D interconnected network^[6,22,23] enhancing the overall mechanical characteristics. By adding ceramic particles and/or short ceramic fibers as reinforcements, some mechanical properties (e.g., strength) are further improved.^[6,22,23]

In our study, the MMC's microstructure is composed of six phases: Al matrix, eutectic Si, IMs, short Al₂O₃ (ceramic) fibers, SiC (ceramic) particles, and voids/cracks. Squeeze casting was employed, and the hybrid preform carried a mat of planar-randomly oriented (on xy-plane) reinforcing fibers. The reinforcing ceramic particles were dispersed randomly across the mat. A full description of the manufacturing process, experimental process, and tools utilized for Synchrotron XCT imaging can be found in ref. [6]. **Figure 2** shows a cross-section (xy-plane) of a high-quality XCT reconstruction of the material (image size of 512 × 512 pixels). We note that various phases (particles, fibers, Al matrix, and some IMs) have comparable X-Ray attenuation coefficients.

1.3. Research Approach Outline

1) We used the same synthetic and experimental Al–Si MMC XCT datasets from refs. [7,10] (common for both publications) for consistency, along with the same manually annotated ground

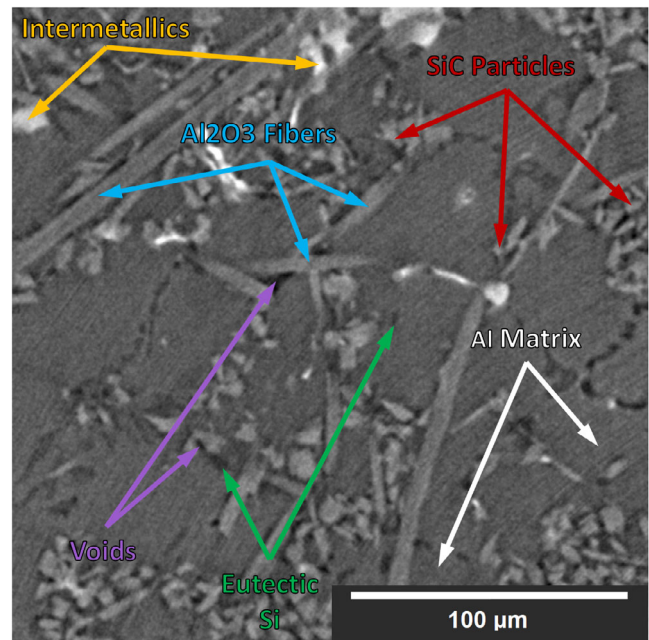


Figure 2. A random xy-plane slice from the reconstructed XCT data of the Al–Si MMC. The various microstructural phases are annotated.

truths for the final Dice score assessment and comparison between this study and our previous studies. Furthermore, we adopted the same training and forwarding strategies including the *i.S.Sy.Da.T.A.* iterative segmentation from ref. [10]. As a new approach, we adapted the *3D Triple_UNet* from ref. [7,10] by increasing the number of trainable parameters (i.e., the number of channels). A justification for this is provided later in the discussion section; 2) Another novelty is that we conditioned (denoised, deblurred, ring artifact reduction) the experimental Al–Si MMC XCT datasets with the deep conditioning framework: *BAM SynthCOND* proposed in ref. [19] (instead of NLM conditioning). The associated DCNNs within the conditioning framework were trained with the same synthetic Al–Si MMC XCT datasets used for training the *Triple_UNet*; and 3) We assessed the improvement of the segmentation Dice scores between our previous studies and the current study, in relation to the more sophisticated conditioning method applied to the experimental datasets (i.e., the *BAM SynthCOND*).

2. Experimental Section

2.1. Synthetic Al–Si MMC XCT Training Data

Since this work builds on the work presented in refs. [7,10] we used the same synthetic Al–Si MMC XCT training/validation datasets; this means, 2 × 7: 512 × 512 × 512 voxel volumes (synthetic volumes and respective microstructural labels). We sliced these volumes into 64 × 64 × 64 voxel volumes (which was the selected input size for the *3D Triple_UNet*), with an omnidirectional Stride = 56. This operation resulted in 5103 64 × 64 × 64 voxel pair volumes, which served as input data and ground truth labels. These were randomly shuffled into training/validation pairs with a ratio of approximately 80/20 (i.e., 4465 pairs for

training, and 638 pairs for validation). Finally, to promote generalized training from synthetic data, we used the data augmentations proposed in refs. [7,10] on the $64 \times 64 \times 64$ volumes during the slicing process. Specifically: $\pm 10\%$ brightness/contrast augmentations (in random order and intensity), 0–1 (random sigma) 3D Gaussian spatial blur, and 0–8 (random standard deviation, based on an 8-bit range: 0–255) Gaussian noise. The augmentations were applied in this specific order.^[7,10]

2.2. Segmentation – *i.S.Sy.Da.T.A.*

The classic approach for performing segmentation with a DCNN is to first train the neural network with the available training data and then perform the forward pass of the target data to be segmented. This approach was successfully followed in ref. [7] with synthetic training data. In ref. [10], to further promote generalized training (when synthetic training data are used for experimental data segmentation), the *i.S.Sy.Da.T.A.* iterative segmentation algorithm was proposed. Such an algorithm, apart from the synthetic training datasets (SDG: Synthetic Dataset Group), employs further experimental datasets, which are separate but statistically equivalent to the target experimental datasets (TEDG: Target Experimental Dataset Group). The latter datasets are used to reinforce the training process (REDG: Reinforcing Experimental Dataset Group). Effectively, with *i.S.Sy.Da.T.A.*, the training process starts ordinarily, employing initially only the SDG to train the DCNN (iteration 0). With this initial fully trained status of the DCNN, the REDGs can then be segmented, sliced, and used as reinforcement training data, combined with the SDG in the next iteration (random shuffling: REDG and SDG combined with 80/20 training/validation data ratio). Ideally, different, or alternate REDGs should be used in consecutive iterations to prevent the same segmentation error transfer in

subsequent iterations ref. [10]. In our case, we employed two distinct REDGs (REDG1 and REDG2, the same from ref. [10]), alternating them between iterations (i.e., iteration 0: <SDG> only, iteration 1: <SDG and REDG1>, iteration 2: <SDG and REDG2>, iteration 3: <SDG and REDG1> again, iteration 4: <SDG and REDG2> again, etc.) Each REDG consisted of three different experimental Al-Si MMC XCT datasets statistically equivalent to the TEDG. Training augmentations were applied only to the SDG and not to the REDGs as there is no need to promote further generalization through them.^[10] In total, we run *i.S.Sy.Da.T.A.* for three iterations as suggested in ref. [10]. The training and validation data are summarized in **Table 1**.

2.3. Data Conditioning, 3D DCNN Architecture, Forwarding Strategy, and Training Parameters

In our previous studies, we employed the popular NLM filter (smoothing factor = 1, sigma = 8) to condition the experimental XCT datasets (we attempted segmenting unconditioned data as well with less satisfactory results). Here, we employed the deep conditioning framework: *BAM SynthCOND*, introduced in ref. [19], which takes advantage of purely synthetic data to train the conditioning DCNNs in a supervised manner. We believe that conditioning experimental data with *BAM SynthCOND*, can yield data that are better perceivable by a segmenting DCNN trained with analogous synthetic data. Therefore, the relevant conditioning training data were essentially the same SDG, specifically adapted for denoising, deblurring, and ring artifact minimization. The exact methodology is given in detail in ref. [19]. As such, we conditioned REDG1, REDG2, and TEDG before proceeding with the *i.S.Sy.Da.T.A.* segmentation. A comparison between the conditioning methods is illustrated in **Figure 3**.

Table 1. Training/validation data summary employed in *i.S.Sy.Da.T.A.*

	Synthetic Al-Si MMC XCT volumes (SDG)	Reinforcing experimental Al-Si MMC XCT volumes (REDG1 or REDG2)	Training/validation volumes slicing stride	Total sub-volume pairs	Training pairs	Validation pairs
Augmentations	YES	NO	–	–	–	–
Iteration 0	7	0	56	5103	4465	638
Iteration n	7	3	56	7290	6379	911

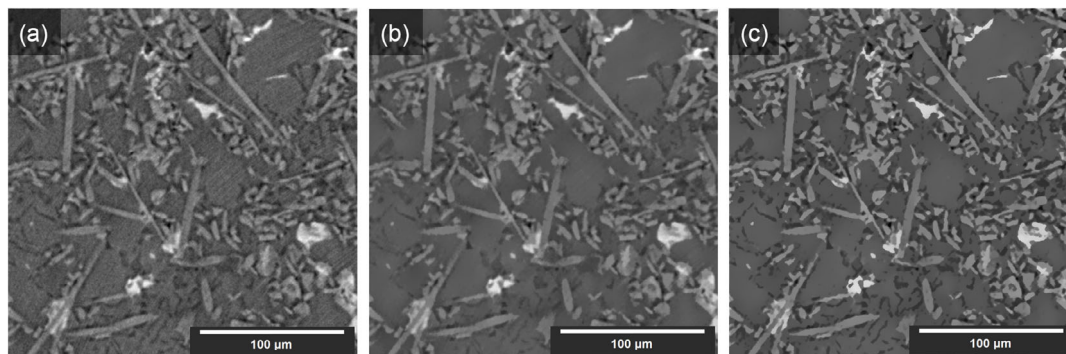


Figure 3. Comparison between conditioning methods for the experimental XCT data (a: Not conditioned, b: NLM-filter conditioned, c: Conditioned with *BAM SynthCOND* from ref. [19]).

For the segmenting DCNN, we adapted the 3D *Triple_UNet* from refs. [7,10] increasing the total number of channels (the output layer remained the same, i.e., 6 output channels as the number of microstructural phases present.) The *Triple_UNet* was proven in ref. [7] to output better segmentations (i.e., superior generalization) with synthetic training data compared to a standard UNet architecture.

For forwarding, we used the same 3D *MultiView* Forwarding Strategy as in refs. [7,10]. With this approach, the TEDG and REDGs were rotated four times with respect to the z-axis (0°, 90°, 180°, 270°) and then omnidirectionally sliced (in 3D) before being forwarded through the trained *Triple_UNet*. The selected size was 64 × 64 × 64 voxels (mini volumes) with a slicing stride = 28 (not to be confused with the training data slicing stride = 56).

Since the deep analysis is performed in 3D, there is no need to optimize the view angles. In fact, such an approach would be a good practice in 2D (i.e., analyzing 3D volumes with a 2D deep learning model). With a 3D deep learning model, the geometry of the microstructure can be fully captured. Thus, geometrical misinterpretations of microstructural particles with similar attenuation coefficients are minimized (e.g., a 2D deep learning model mislabeling a fiber vertical to the plane of view as a particle). However, the 4 rotations (views) can still be beneficial with a 3D model as we demonstrated in a previous publication ref. [7] (1% improvement in the Dice scores for every microstructural phase). That is because the increased number of samples reduces the segmentation error arising from the stochastic nature of the AI model.

After the forward pass, the resulting probability mini-volumes were assembled (added/overlapped) back into probability volumes according to the slicing stride. The four views were rotated back to 0° and added together according to their respective microstructural phase (class), before the final classification (i.e., segmentation result). A schematic of the adapted 3D *Triple_UNet* and 3D *MultiView* forwarding strategy is shown in **Figure 4**.

The 3D *Triple_UNet* was built and trained with *i.S.Sy.Da.T.A.* segmentation, utilizing Sony's Neural Network libraries,^[24] on a workstation with specs: Intel Pentium i7 processor, Nvidia GeForce RTX 3090 graphics card, and 32 GB of RAM. The ADAM algorithm^[25] was chosen as the optimizer (with parameters: initial learning rate/alpha = 1e−4, beta1 = 0.9, beta2 = 0.999, updated at every iteration), with input batch_size = 48, and exponential learning rate multiplier (i.e., decay) = 0.9, for a total of 50 epochs. Additionally, every epoch involved a random shuffling of the training dataset to prevent stagnation. Both the training and validation errors were recorded, but much as in our previous studies, the final learnable parameters were picked from the epoch that reduced the validation error (to prevent overfitting). The whole pipeline of the training/segmentation procedure in our study is summarized in **Figure 5**.

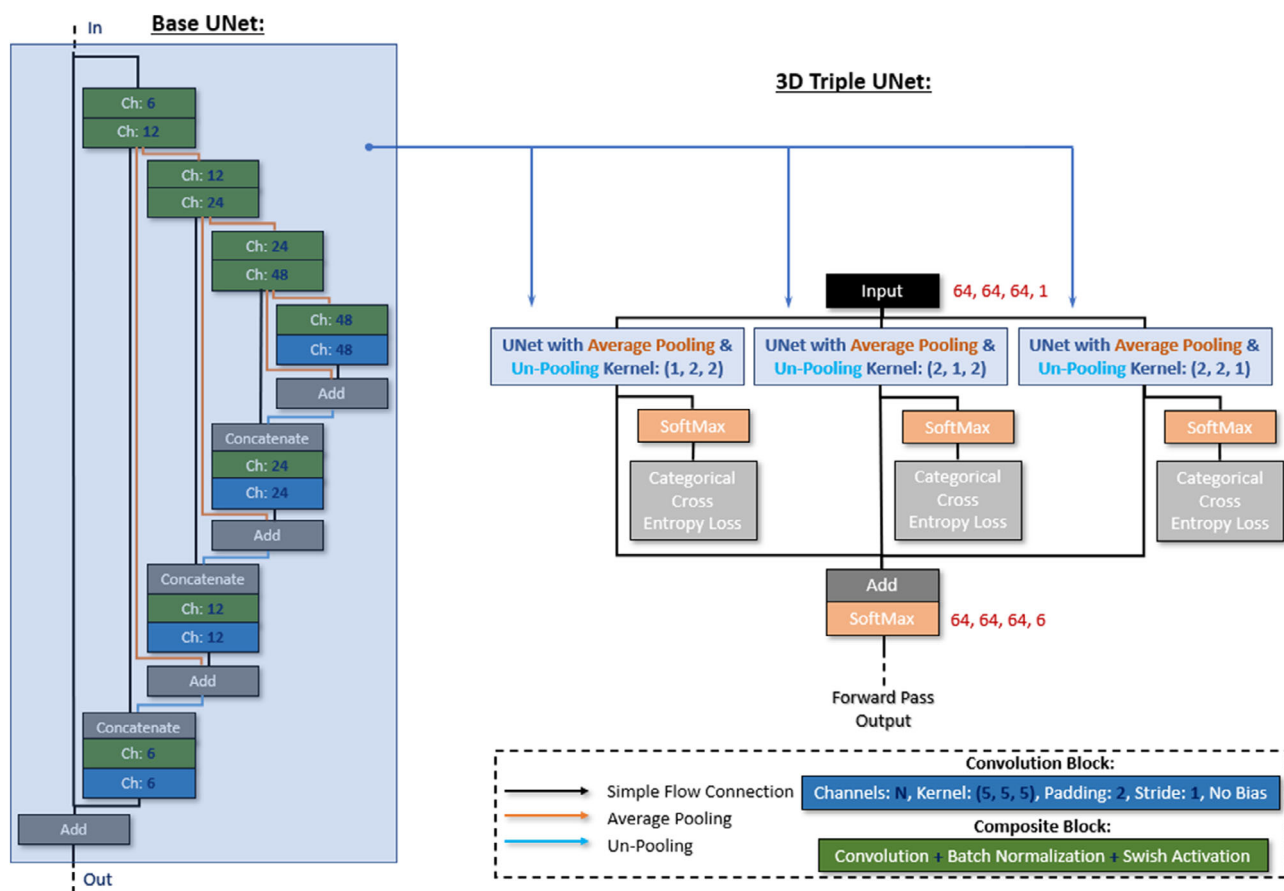
3. Results and Discussion

The TEDG consisted of four statistically equivalent experimental Al–Si MMC XCT 512 × 512 × 512 voxel volumes. From each volume, a slice was randomly extracted and manually labeled to be used as ground truth data for the resulting segmentation

precision (Dice score) calculation (i.e., four slices in total). Both the TEDG and extracted slices are in line with the previous studies^[7,10] for scientific consistency. However, as mentioned in ref. [10], the eutectic Si phase ground truth labels were underlabeled (due to human error). Thus, in this work, we have revised the ground truth slices and addressed the problem of underlabeling. In this section, we show how each strategy substantially improved generalization and hence segmentation precision (i.e., average Dice scores of the four slices for each microstructural phase). Furthermore, we report (in **Table 2**) the standard deviation (STD) of the Dice scores obtained in this study. For the calculation of the STD, the four ground truth slices and their respective segmentations were sliced in quadrants, increasing the number of samples to 16 image pairs. All the results from our previous studies have been revised to reflect the updated ground truth slices (i.e., updated eutectic Si labels), and only the updated results (tables and figures) are reported. These updated Dice scores are compared with those obtained in the current study. Thus, we lay out a complete strategy/roadmap to achieve high automatic segmentation precision on experimental XCT data from low resemblance synthetic training data.

The effects of each strategy (from our previous studies toward this study) using low resemblance synthetic training data, are reported in **Table 2**. We observe from the first line in **Table 2** that the Dice scores for the most challenging microstructural phases (Fibers, IMs, and Particles: 0.34, 0.53, and 0.48, respectively), are quite low when a standard segmentation approach (no augmentations employed, standard single view slicing) and a standard UNet architecture are used. The performance of standard approaches is relatively poor, even though the experimental XCT data were NLM conditioned (result from ref. [7]). The introduction of augmentations (brightness, contrast, Gaussian noise, Gaussian blur), 3D *MultiView* forwarding strategy, and 3D *Triple_UNet* architecture improve substantially the Dice scores (fibers, IMs, and particles: 0.49, 0.55, and 0.66, respectively, revised result from ref. [7]). The introduction of the *i.S.Sy.Da.T.A.* segmentation strategy improves even further the Dice scores (fibers, IMs, and Particles: 0.54, 0.66, and 0.72, respectively, best-revised result from ref. [10]). Yet, the Dice scores for the fibers and IMs remain below 0.70, which is considered as the threshold for satisfactory segmentations. The replacement of NLM conditioning filter with *BAM SynthCOND* from ref. [19], trained with the same synthetic XCT data used to train the *Triple UNet*, has a large impact: not only the challenging microstructural phases were better segmented, but also all Dice scores improved (fibers, IMs, and particles: 0.71, 0.70, and 0.74, respectively, Si, Al Matrix, and overall Dice score: 0.88, 0.89, and 0.87 respectively). The use of *BAM SynthCOND* from ref. [19] yields Dice scores of at least 0.70 for every category. The low STDs obtained, indicate consistent segmentation precision across the TEDG. In **Figure 6**, the segmentation maps for all the above cases are given (same order as above: **Figure 6a–d**, from worst to best segmentation). Certain regions of interest, where significant improvements were observed, are highlighted with white boxes in the figure.

It is apparent that to achieve good automatic segmentations on experimental XCT data with low resemblance to synthetic training data, the conditioning step of the experimental data is crucial. This is reasonable since the exact artifact and gray-level patterns



MultiView Forwarding Strategy:

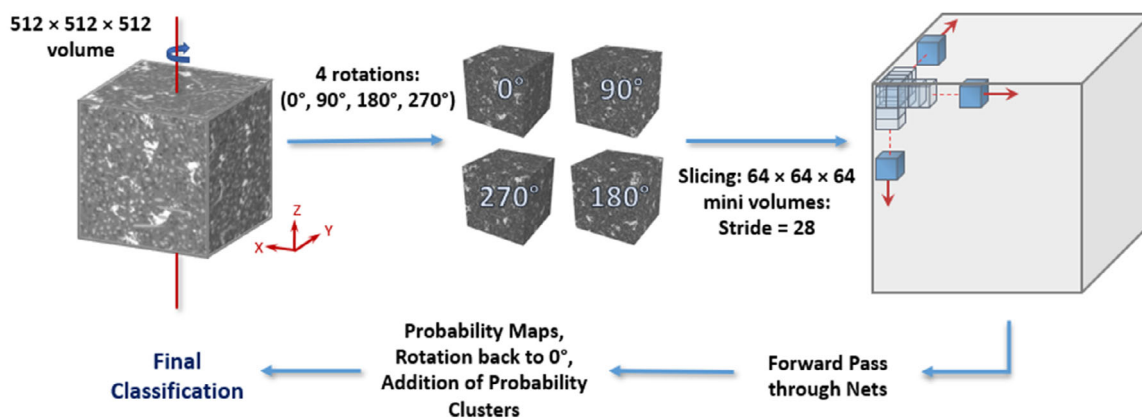


Figure 4. The 3D Triple_UNet architecture and 3D MultiView forwarding strategy. Adapted and reproduced figure under terms of the CC-BY 4.0 license [terms: CC BY 4.0 Legal Code | Attribution 4.0 International | Creative Commons].^[7] Copyright 2023, A. Tsamos et al. MDPI Journal of Imaging.

within the experimental datasets are not easily replicable in synthetic datasets. As reported in Table 3, the Dice scores for all microstructural phases increase significantly with better conditioning (unconditioned XCT data, NLM conditioned XCT data, BAM SynthCOND conditioned XCT data).

In our previous studies,^[7,10] we deliberately kept the number of parameters low, to avoid overfitting on the (artifact-free)

synthetic training data. In fact, in refs. [7,10] a Triple_UNet with more channels results in segmentations with lower segmentation precision, which is the case with the NLM filter where some artifacts remain after conditioning (Figure 3b). In other words, the Triple_UNet attempts to segment data with some artifacts even though it is trained to accurately segment artifact-free data. The introduction of BAM SynthCOND,^[19] allows us to overcome

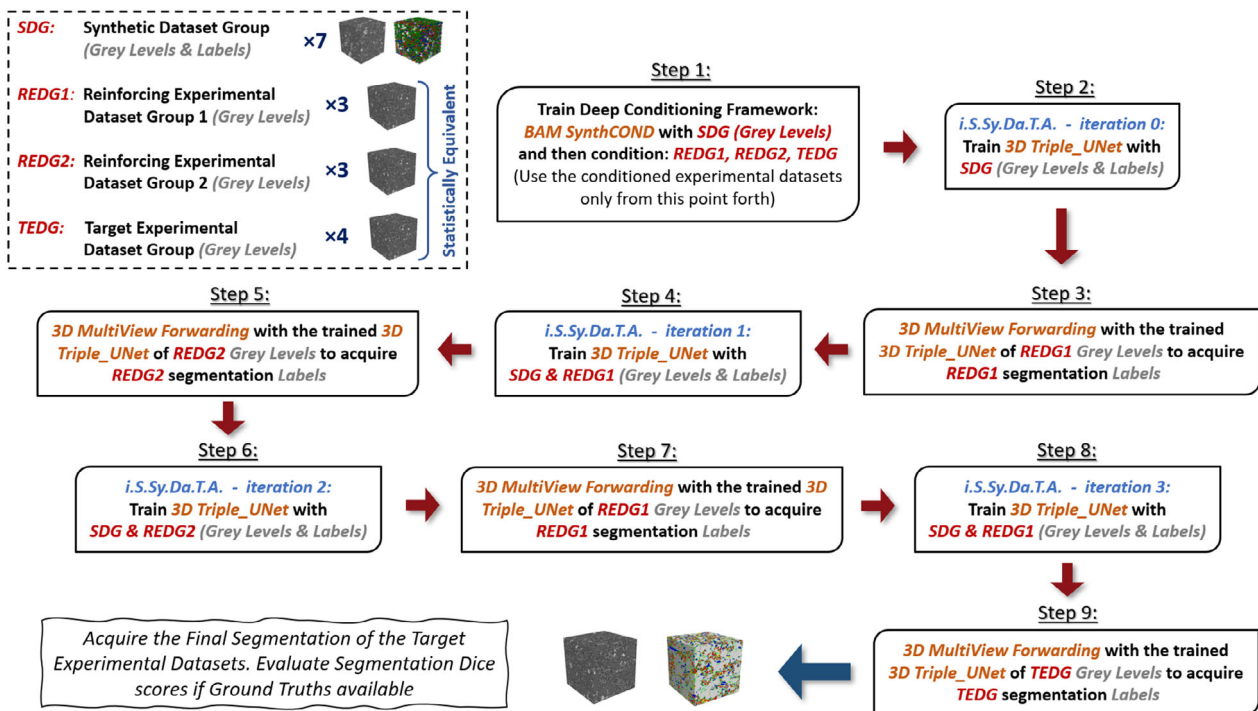


Figure 5. Pipeline of the data handling and training (*i.Sy.Da.T.A.*) employed in the current study.

Table 2. Resulting segmentation dice scores from our previous studies and current study with the revised ground truth (i.e., revised eutectic si labels). STDs are reported only for the newest segmentation result obtained in the current study.

Experimental data segmentation - DICE	Fibers	IMs	Si	Particles	Al matrix	Overall
With low resemblance synthetic training data						
[STD]						
No augmentations, standard UNet, single view, NLM8 conditioning	0.34	0.53	0.66	0.48	0.76	0.65
Augmentations, Triple UNet, MultiView, NLM8 conditioning	0.49	0.55	0.77	0.66	0.87	0.80
Augmentations, Triple UNet, MultiView, NLM8 conditioning, <i>i.Sy.Da.T.A.</i> (3 iter.)	0.54	0.66	0.76	0.72	0.88	0.82
Augmentations, Triple UNet, MultiView, BAM SynthCOND, <i>i.Sy.Da.T.A.</i> (3 iter.)	0.71	0.70	0.88	0.74	0.89	0.87
	(0.092)	(0.109)	(0.013)	(0.060)	(0.018)	(0.028)

this limitation since the artifacts within the experimental XCT datasets are minimized after conditioning (practically artifact-free, Figure 3c). Consequently, we could increase the number of channels (i.e., parameters) in the current *Triple_UNet* implementation, and therefore improve performance, without incurring overfitting. Furthermore, since the deep conditioning framework and the *Triple_UNet* are trained with the same synthetic XCT data (i.e., the SDG), the conditioned TEDG, REDG1, and REDG2 experimental datasets are compelled (in a supervised manner) to be entropically analogous (i.e., similar gray-level patterns). This allows the segmentation network to focus more on learning/identifying geometric features instead of gray-level patterns. This is another reason which allowed us to increase the total number of channels and still avoid overfitting.

4. Conclusion

In this study, we prove that to achieve high automatic segmentation precision on challenging experimental XCT data from low resemblance synthetic training data, preconditioning of experimental datasets is imperative. The deep conditioning framework: *BAM SynthCOND* can well tackle the problem. In fact, it can outperform the already well-established NLM filtering in terms of conditioning performance (i.e., fewer remaining artifacts: noise, blur, ring artifacts) and can prepare the experimental datasets to be entropically analogous to the synthetic training datasets. This is because *BAM SynthCOND* employs the same synthetic training data as the segmenting DCNN. Thus, significant improvement in the resulting segmentation precision is observed compared to

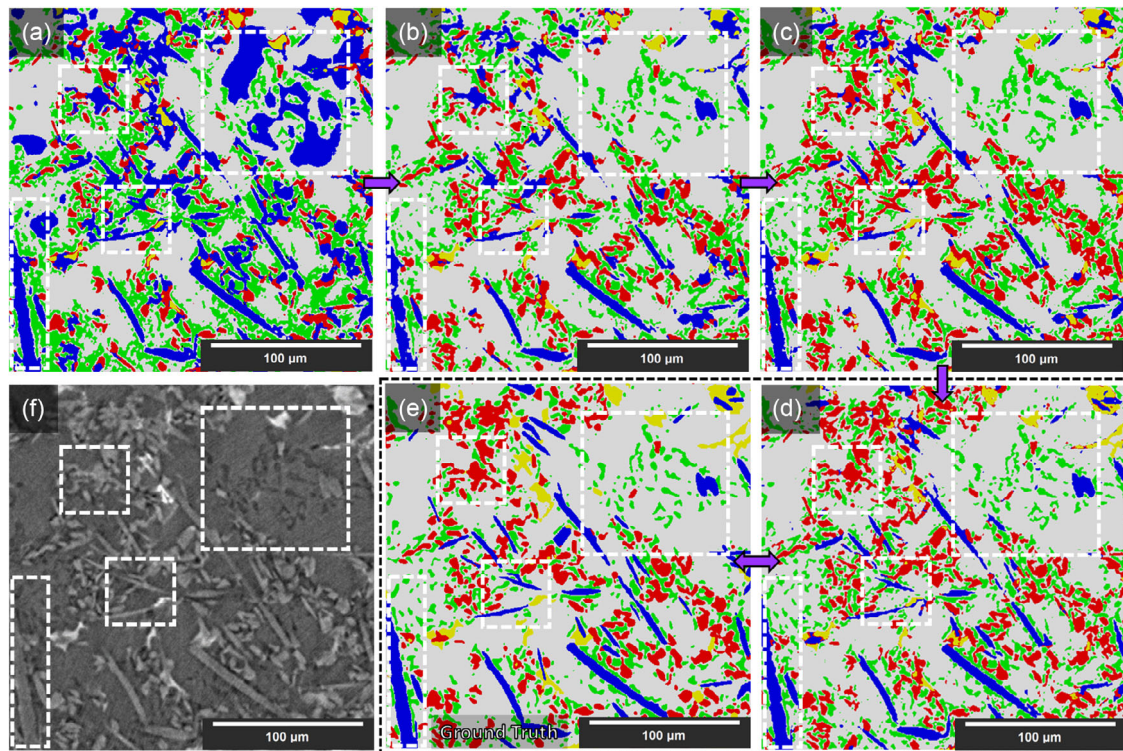


Figure 6. Comparison of resulting segmentation maps between methods from our previous studies and the current study. (a: No augmentations, standard UNet, single-view slicing, NLM8 conditioning, b: Augmentations, *Triple_UNet*, *MultiView*, NLM8 conditioning, c: Augmentations, *Triple_UNet*, *MultiView*, NLM8 conditioning, *i.S.Sy.Da.T.A.* - 3 iterations, d: Augmentations, *Triple_UNet*, *MultiView*, *BAM SynthCOND* conditioning, *i.S.Sy.Da.T.A.* - 3 iterations, e: Revised ground truth labels for Si phase, f: Reconstructed not conditioned XCT data of the Al-Si MMC). Labels color: {blue: Al_2O_3 fibers, red: SiC particles, green: eutectic Si, yellow: IMs, gray: Al matrix}. The white boxes indicate regions with significant segmentation precision improvements.

Table 3. Comparison between the resulting dice scores and the conditioning method used for the experimental XCT data (with the *i.S.Sy.Da.T.A.* Segmentation, *Triple_UNet*, *MultiView* forwarding, and special augmentations).

Experimental data segmentation - DICE	Fibers	IMs	Si	Particles	Al matrix	Overall
Augmentations, Triple Unet, MultiView, No conditioning, <i>i.S.Sy.Da.T.A.</i> (3 iter.)	0.49	0.44	0.68	0.61	0.84	0.73
Augmentations, Triple Unet, MultiView, NLM8 conditioning, <i>i.S.Sy.Da.T.A.</i> (3 iter.)	0.54	0.66	0.76	0.72	0.88	0.82
Augmentations, Triple Unet, MultiView, <i>BAM SynthCOND</i> conditioning, <i>i.S.Sy.Da.T.A.</i> (3 iter.)	0.71	0.70	0.88	0.74	0.89	0.87

NLM conditioning. The entropic similarity between synthetic and experimental datasets promoted the use of a larger segmentation DCNN, further improving the performance. This allows higher segmentation precision, contrary to our previous studies.

Based on our findings, we propose the following strategy to attain good automatic segmentation from low resemblance synthetic XCT training data: 1) Specific data augmentations on the synthetic training data: brightness, contrast, random Gaussian noise, and random Gaussian blur; and 2) The 3D *Triple_UNet*, the 3D *MultiView* forwarding strategy, the *i.S.Sy.Da.T.A.* segmentation, and finally the deep conditioning framework: *BAM SynthCOND*, for treating all the experimental XCT datasets.

Since the proposed strategy does not depend on the specific details of the images considered, we expect it to improve segmentation for other imaging techniques as well.

Acknowledgements

This work has been funded by the BAM-internal Project "Ideen_2019-028". Open Access funding enabled and organized by Projekt DEAL.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

BAM SynthCOND, BAM SynthMAT, i.S.Sy.Da.T.A. segmentations, synthetic training data, Triple_UNet

Received: July 7, 2023

Revised: November 3, 2023

Published online:

-
- [1] S. M. Azimi, D. Britz, M. Engstler, M. Fritz, F. Mücklich, *Sci. Rep.* **2018**, *8*, 2128.
- [2] P. Acharya, T. P. Chu, K. R. Ahmed, S. Kharel, *Int. J. Artif. Intell. Appl.* **2022**, *13*, 1.
- [3] W. Du, H. Shen, J. Fu, *IEEE Trans. Ind. Electron.* **2021**, *68*, 12912.
- [4] T. Konopczynski, D. Rathore, J. Rathore, T. Kröger, L. Zheng, C. S. Garbe, S. Carmignato, J. Hesser, (Preprint) arXiv:1901.01211, v1, Submitted: Jan 2019.
- [5] S. Bellens, P. Vandewalle, W. Dewulf, *Procedia CIRP* **2021**, *96*, 336.
- [6] S. Evsevlev, S. Paciornik, G. Bruno, *Adv. Eng. Mater.* **2020**, *22*, 1901197.
- [7] A. Tsamos, S. Evsevlev, R. Fioresi, F. Faglioni, G. Bruno, *J. Imaging* **2023**, *9*, 22.
- [8] N. Otsu, *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62.
- [9] T. Strohmam, K. Bugelnig, E. Breitbarth, F. Wilde, T. Steffens, H. Germann, G. Requena, *Sci. Rep.* **2019**, *9*, 19611.
- [10] A. Tsamos, S. Evsevlev, R. Fioresi, F. Faglioni, G. Bruno, *Comput. Mater. Sci.* **2023**, *223*, 112112.
- [11] B. Lin, N. Emami, D. A. Santos, Y. Luo, S. Banerjee, B.-X. Xu, (Preprint) arXiv:2109.04429, v1, Submitted: Sep 2021.
- [12] B. Ma, X. Wei, C. Liu, X. Ban, H. Huang, H. Wang, W. Xue, S. Wu, M. Gao, Q. Shen, M. Mukeshimana, A. O. Abuassba, H. Shen, Y. Su, *npj Comput. Mater.* **2020**, *6*, 125.
- [13] A. Boikov, V. Payor, R. Savelev, A. Kolesnikov, *Symmetry* **2021**, *13*, 1176.
- [14] E. Friedman, A. Lehr, A. Gruzdev, V. Loginov, M. Kogan, M. Rubin, O. Zvitia, (Preprint) arXiv:2303.15219, v1, Submitted: Mar 2023.
- [15] T. Alkhalifah, H. Wang, O. Ovcharenko, *Artif. Intell. Geosci.* **2022**, *3*, 101.
- [16] O. Ronneberger, P. Fischer, T. Brox, in *Lecture Notes in Computer Science* **2015**, pp. 234–241.
- [17] A. Buades, B. Coll, J.-M. Morel, in *2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05)* **2005**.
- [18] H. Zhang, D. Zeng, H. Zhang, J. Wang, Z. Liang, J. Ma, *Med. Phys.* **2017**, *44*, 1168.
- [19] A. Tsamos, S. Evsevlev, G. Bruno, *Tomogr. Mater. Struct.* **2023**, *2*, 100011.
- [20] G. Requena, H. P. Degischer, *Mater. Sci. Eng., A* **2006**, *420*, 265.
- [21] K. U. Kainer, *Metal Matrix Composites: Custom-Made Materials For Automotive And Aerospace Engineering*, Wiley-Vch, Weinheim **2006**.
- [22] S. Evsevlev, T. Mishurova, S. Cabeza, R. Koos, I. Sevostianov, G. Garcés, G. Requena, R. Fernández, G. Bruno, *Mater. Sci. Eng., A* **2018**, *736*, 453.
- [23] S. Evsevlev, S. Cabeza, T. Mishurova, G. Garcés, I. Sevostianov, G. Requena, M. Boin, M. Hofmann, G. Bruno, *J. Mater. Sci.* **2020**, *55*, 1049.
- [24] Neural Network Libraries. An open-source software to make research, development, and implementation of neural network more efficient. Sony corp. <https://nnabla.org/>.
- [25] D. P. Kingma, J. Ba, (Preprint) arXiv:1412.6980, v1, Submitted: Dec 2014.