

Research

Geometric deep learning for enhanced quantitative analysis of microstructures in X-ray computed tomography data

M. Lapenna¹ · A. Tsamos^{2,3} · F. Faglioni⁴ · R. Fiorese⁵ · F. Zanchetta⁵ · G. Bruno^{2,3}

Received: 2 December 2023 / Accepted: 22 May 2024

Published online: 07 June 2024

© The Author(s) 2024 [OPEN](#)

Abstract

Quantitative microstructural analysis of XCT 3D images is key for quality assurance of materials and components. In this paper we implement a Graph Convolutional Neural Network (GCNN) architecture to segment a complex Al-Si Metal Matrix composite XCT volume (3D image). We train the model on a synthetic dataset and we assess its performance on both synthetic and experimental, manually-labeled, datasets. Our simple GCNN shows a comparable performance, measured via the Dice score, to more standard machine learning methods, but uses a greatly reduced number of parameters (less than 1/10 of parameters), features low training time, and needs little hardware resources. Our GCNN thus achieves a cost-effective reliable segmentation.

Article Highlights

- We present a novel Graph Convolutional Neural Network approach to segment a 3D image of a complex Al-Si Metal Matrix composite.
- Training on synthetic datasets shows a semantic understanding of the ground truth labels of manually-labeled experimental images.
- Our simple Graph Neural Network shows a comparable performance to standard methods, using a reduced number of parameters, low training time, towards cost-effective reliable segmentation.

Keywords Geometric deep learning · Segmentation · Microstructure · X-ray computed tomography · Al–Si metal matrix composites

1 Introduction

The field of material science is experiencing a revolution: the new and successful deep learning inspired algorithms are changing the landscape of XCT (X-ray computed tomography) quantitative analysis (see [1], and refs therein, for a comprehensive account of recent developments).

✉ A. Tsamos, athanasios.tsamos@bam.de; ✉ R. Fiorese, rita.fiorese@unibo.it; M. Lapenna, michela.lapenna4@unibo.it; F. Faglioni, francesco.faglioni@unimore.it; F. Zanchetta, ferdinando.zanchett2@unibo.it; G. Bruno, giovanni.bruno@bam.de | ¹DIFA, Unibo, Via Irnerio 46, Bologna 40126, Italy. ²Bundesanstalt für Materialforschung und -Prüfung, BAM, Unter den Eichen 87, Berlin 12205, Germany. ³Institute of Physics and Astronomy, University of Potsdam, Karl-Liebknecht-Straße 24/25, Potsdam 14476, Germany. ⁴DGSC, Unimore, Via Campi 103, Modena 41100, Italy. ⁵Fabit, Unibo, Via San Donato 15, Bologna 40127, Italy.



More specifically, quantitative microstructural analysis of XCT images, via an accurate statistical assessment of individual features, can greatly improve the assurance checks necessary for the qualification of materials and components. The use of XCT allows reducing the time and cost involved with materials development and/or qualification. Such analysis, via machine learning methods, is best performed through automatic segmentation, where each voxel of a 3D XCT image is analyzed and classified depending on whether the voxel belongs to a certain class.

More specifically, recent works [2–4] provide successful algorithms for the automatic segmentation of XCT reconstructions of additively manufactured materials, respectively with a 3D DCNN (Deep Convolutional Neural Network) [3], with a 2D DCNN [2], and with autoencoders [4], towards reliable quality assurance. These automatic segmentation methods were originally developed for applications in medical imaging, such as detection and segmentation of tumors or other relevant macroscopic or microscopic medically significant objects (see [5] for a recent full review of techniques). The application of DCNNs in material science is now the paradigm and state of the art technique for automatic segmentation.

In the present work, we focused on the automatic segmentation of a six-phase Al-Si alloy composite reinforced with ceramic fibers and particles. Since such microstructures are extremely challenging to be quantitatively analyzed by manual segmentation, neural networks are becoming more and more employed with high accuracies [6–8].

The best way to segment these multiphase composite material XCT 3D images is to directly work on the 3D reconstructed volume with 3D networks. That is because different microstructural phases can share similar X-ray attenuation coefficients (similar densities) and, consequently, similar grayscales in the reconstructed XCT data [8]. It is then important to capture their 3D shape in order to distinguish them. Furthermore, if many microstructural phases are present, geometrical features become more and more important for accurate feature recognition [8]. For this reason, a 2D model may fail in the segmentation task. For instance, in [7], due to the similar grayscales of the reinforcing SiC particles and Al_2O_3 fibers, the in-plane vertical fibers were wrongly classified as particles with a 2D DCNN.

Up to now, the Deep Learning 3D autoencoders as in [4, 6–8] proved more effective for this task, however quite demanding in training time and RAM usage. Therefore, due to the time-consuming manual labelling, together with the demanding computer resources of the present approaches, new deep learning techniques need to be investigated.

2 Novelty of the approach

In this work we propose a novel approach via Geometric Deep Learning (Graph Convolutional Neural Networks, GCNNs [9, 10]), to the problem of segmentation of XCT reconstructions of a six-phase Al-Si alloy composite, reinforced with ceramic fibers and particles. For the training of our GCNN we use the synthetic dataset provided by [8], created to solve the time-consuming problem of manually labeling XCT volumes, while for the test we use both synthetic and experimental manually-labeled data, also provided by [8], to which we compare directly our performances. It is important, however, to remark that our approach to the question, via Graph Neural Networks, is new and different from all the previous approaches mentioned above employing standard DCNNs implementations.

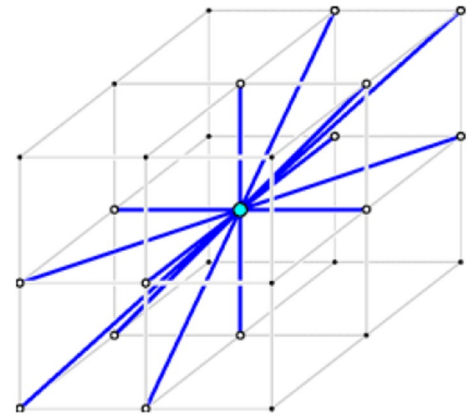
Our aim is to give a proof of concept to assess the advantages of using GCNNs in the segmentation of XCT data of complex microstructures, with respect to the more conventional DCNN architectures mentioned above and based on autoencoders.

The first significant feature of GCNNs is that they take graphs as input data, that is, they take a set of nodes, with associated features together with their links. The volume represented by the XCT dataset is naturally viewed as a graph (see Fig. 1), where the voxels are the nodes and a node is connected by links with the neighboring ones.

The spatial orientation of the graph in a GCNN does not influence its learning, since graph convolutions in GCNNs are invariant with respect to rotations of the graph and permutations of its nodes (see Sect. 4). This precisely fits the setup of our problem: Al-Si alloy composite segmentation, i.e. voxel classification, is independent of its spatial orientation (up, down, left and right). This natural invariance makes GCNN architectures a more natural choice for this kind of segmentation, with respect to the more conventional 2D and 3D DCNNs. In fact, the convolutional kernels of DCNNs on 2D and 3D Euclidean grids intrinsically depend on orientation. This advantage allows GCNNs to reach comparable accuracies with far fewer parameters, since they come with such symmetries already built in.

Moreover, viewing a volume (i.e. 3D XCT reconstruction) as a graph, allows a more flexible approach, since nodes representing voxels can contain information not just according to their spatial position, but, thanks to the links between

Fig. 1 XCT volumes as graphs; the depicted links are just an example illustration of the possibilities



nodes, we can also encode the similarity in features. Thus, we can obtain a more effective node (i.e. voxel) classification (see [11] for a clear illustration of this fact in a toy social network example [12]).

Another advantage of GCNNs is their ability to exploit the graph structure in node classification problems, and then perform the training with just a small subset of available nodes. In other words, GCNNs can typically train much faster, leading to an even further reduced number of parameters, in comparison with DCNNs performing similarly on the same task.

These advantages will translate, once the geometric approach is properly optimized, into a significant reduction of computational costs, both in terms of training time and hardware requirements (e.g. RAM). Thus, our novel approach, which is currently at the proof of concept stage, will not be just cost effective, but will allow researchers to tackle more challenging questions, currently out of reach even for the most powerful DCNNs.

3 Materials and data

We examine an AlSi12CuMgNi matrix metal composite (MMC) reinforced with 7%vol Al₂O₃ short fibers and 15%vol SiC particles, see [8] for a detailed description (Fig. 2 is taken from there). Synchrotron XCT (SXCT) data were acquired at the BAMline beamline at the BESSY II synchrotron in Berlin, Germany. The experimental procedure and equipment used for the SXCT imaging and the analysis of the microstructure are described in detail in [7]. In Fig. 2 we show a 512 × 512 pixel cross-section of the XCT reconstruction of the material. We notice how different components of the material may share similar gray levels.

In supervised Deep Learning models, in order to perform successful training we need large amounts of labelled data. For this reason, we take a synthetic dataset for the training of our GCNN: an exhaustive description of the procedure guiding the generation of such dataset is found in [8]. Here we briefly summarize it for the sake of completeness. Starting from a Metal Matrix Composite (MMC) background as in [7], an in-house MATLAB library (BAM SynthMAT [8]) generates

Fig. 2 XCT reconstruction slice of the AlSi12CuMgNi Metal Matrix Composite (MMC), see [8]

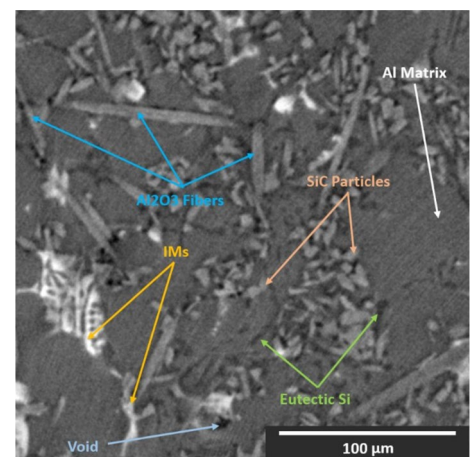
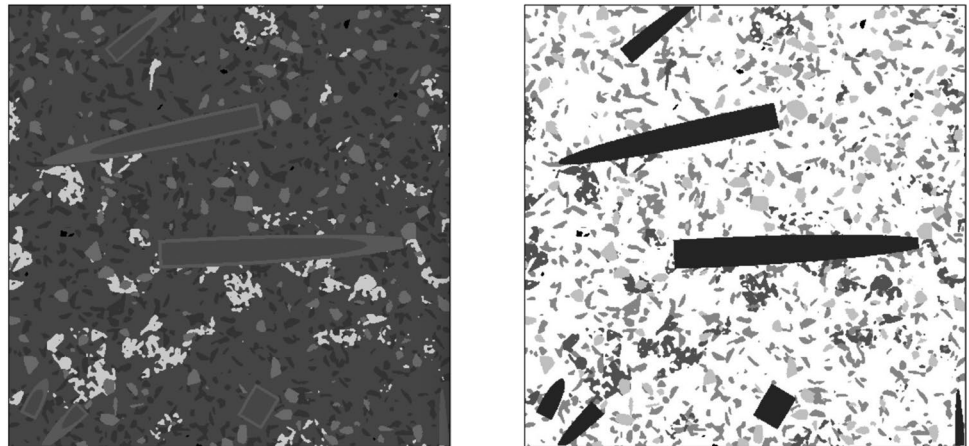


Table 1 Mean occurrence of the 6 phases in the synthetic dataset

| | Voids | Al ₂ O ₃ Fibers | IMs | Si | SiC particles | Al matrix |
|------------|--------|---------------------------------------|--------|---------|---------------|-----------|
| Occurrence | 0.13 % | 4.32 % | 3.90 % | 10.31 % | 4.41 % | 76.92 % |

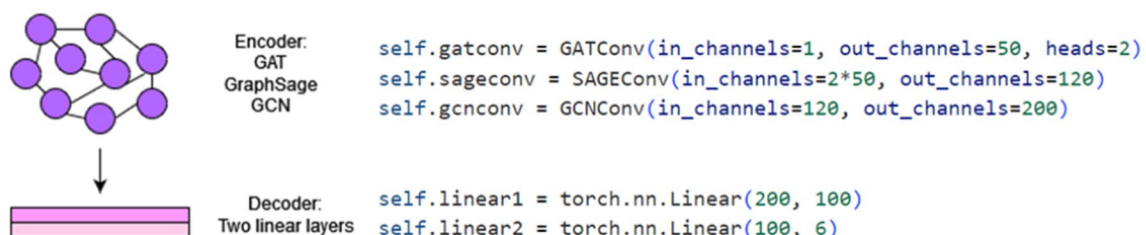
Fig. 3 On the left a slice of the synthetic Al-Si MMC, on the right the corresponding labels

synthetic Al-Si MMCs microstructures similar to those of an XCT reconstruction. The routine is based on structural resemblance and grayscales, and a priority function assembles the generated microstructures into a single volume. Functions assign grayscales to voxels to account for local phase contrast, noise and blur. Since experimental data are inhomogeneous in spatial particle distribution and volume fractions, the synthetic volumes are created accordingly. This allows obtaining a phase distribution mimicking the experimental one.

The resulting synthetic volumes are extracted/saved as raw 8-bit binary data with assigned grayscales (range 0–255) and as raw 8-bit binary data containing specified labels for the various synthetic phases (labels from 0 to # of phases-1). In the case of the Al-Si MMC, there are 6 different phases: voids, Al₂O₃ fibers, Intermetallics (IMs), eutectic Si, SiC particles, and Al matrix (see Table 1). Since the distribution of phases is not uniform, we weigh the contribution of each phase to the loss function according to its occurrence (see Sect. 4). In Table 1 we show the average occurrence of each phase in the synthetic volumes.

The synthetic dataset we use consists of $8512 \times 512 \times 512$ voxel synthetic Al-Si MMCs volumes, each generated with different parameters (different volume fractions, particles, fiber sizes/lengths, orientations, grayscales, etc.). Of these, 7 volumes are used for training/validation and one for testing. In Fig. 3 we show a cross-section from one of the synthetic volumes and the corresponding labels.

The experimental dataset on which we test the trained model consists of four conditioned $512 \times 512 \times 512$ voxel XCT volumes with only one manually-labeled slice for each volume. These slices are used as ground truth when evaluating the trained model on the experimental manually-labeled volumes.

**Fig. 4** The diagram of our GCNN architecture

4 Methods

We employ GCNNs to obtain semantic segmentation of an XCT volume of an AlSi12CuMgNi MMC reinforced with 7% vol Al₂O₃ short fibers and 15%vol SiC particles, see Fig. 2 for a slice of such volume. 3D Image semantic segmentation means the classification of each voxel in a given volume (an XCT volume in this case) so that we know to which class it belongs and we are, therefore, able to reconstruct objects within the given 3D reconstruction (i.e. aluminum fibers, SiC particles, etc.). We view this question as a supervised node classification problem with GCNNs, taking advantage of the success of such methods in their applications to social networks and text databases [11, 13].

Given an XCT volume, we first build a graph where each node is a voxel labelled with a number ranging from 0 to 5, according to the phase it represents: voids, Al₂O₃ fibers, Intermetallics (IMs), eutectic Si, SiC particles, and Al matrix. We also give to each node a feature, which is an integer between 0 and 255, corresponding to its gray level. We then establish links between a node and each of its 6 nearest neighboring ones, similarly to Fig. 1. As it will be clear from our treatment below, we use an *attention mechanism* [14], so that some links are more important than others. Once our algorithm is trained, it gives a label to each voxel in the test XCT volume, achieving the full volume segmentation. We measure the accuracy of predictions via the Dice score (see Sect. 5). We briefly graphically describe our simple GCNN segmentation algorithm in Fig. 4.

As any GCNN, the algorithm is built with an *encoder* and *decoder* architecture. The encoding is obtained by a sequence of the three convolutional layers, each followed by a non-linearity: GAT (Graph Attention Networks), GraphSage, and GCN (Graph Convolutional Network) (Fig. 4). The decoder consists of two linear layers separated by a non-linear activation function (Fig. 4). Before looking at the encoder in more detail, we briefly explain Graph Convolutional Layers (GCL) in general. A GCL can exchange information between nodes by aggregating their features in a process called *Message Passing* and by concatenating node features.

- *Aggregate*: we add information to the feature h_v of a node v , with operations involving the features of its neighboring nodes $\mathcal{N}(v)$.
- *Concatenate*: we concatenate the features of a node with the features of the neighboring ones.

We now go more specifically into the implementation of the GCLs in our GCNN.

- *GAT layer* [14]. It takes the XCT volume as input. The attention mechanism (Graph Attention i.e. GAT layer) is designed to attribute weights to the most relevant parts of the input features. We implement a GAT convolutional layer using the Pytorch GATConv. This layer is proposed in [14] and consists of a single-layer feedforward neural network, parametrized by a weight vector \mathbf{a} in \mathbf{R}^{2p} and a weight matrix $\mathbf{W} \in \mathbf{R}^{p \times n}$ embedding the features from \mathbf{R}^n to \mathbf{R}^p (where n is the initial dimension of the feature vector). To stabilize the learning process of self-attention and take advantage of feature diversity, we introduce multiple attention heads: we perform parallel convolutions with different weight matrices and we concatenate the final outputs. The formula for the attention coefficients is the following:

$$\alpha_{vw} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T W h_v + \mathbf{a}^T W h_w\right)\right)}{\sum_{k \in \mathcal{N}(v) \cup \{v\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T W h_v + \mathbf{a}^T W h_k\right)\right)} \quad (1)$$

where v and w are different nodes and $\mathcal{N}(v)$ is the set of nodes linked with v . We will then use the attention coefficients to aggregate neighboring nodes:

Table 2 Dataset split ratio

| Training | Validation | Test |
|----------|------------|------|
| 52% | 35% | 13% |

Table 3 Dice score of the GCNN model on the synthetic testing volume and on the 4 experimental slices

| Test data | Al ₂ O ₃ fibers | IMs | Si | SiC particles | Al matrix | Overall |
|--------------|---------------------------------------|----------------|----------------|----------------|----------------|--------------------------------|
| Synthetic | 91.54 ± 0.60 % | 90.24 ± 1.25 % | 88.40 ± 0.50 % | 89.85 ± 0.63 % | 95.80 ± 0.15 % | 89.03 ± 0.45 %; 93.90 ± 0.23 % |
| Experimental | 30.50 ± 6.00 % | 61.45 ± 4.50 % | 46.00 ± 0.90 % | 41.23 ± 5.38 % | 72.68 ± 1.00 % | 50.37 ± 3.55 %; 58.00 ± 0.36 % |

The values and their errors are obtained by taking the mean and standard deviation over 10 trainings. We report two figures for the overall score: the first one averaging on classes, the second one for a comparison with [8]

Table 4 Dice scores obtained by the Single U-Net of [8] in the *Plain, Single View* case

| Test data | Al ₂ O ₃ Fibers | IMs | Si | SiC particles | Al matrix | Overall |
|--------------|---------------------------------------|------|------|---------------|-----------|---------|
| Synthetic | 99 % | 99 % | 97 % | 99 % | 99 % | 99 % |
| Experimental | 34 % | 53 % | 50 % | 48 % | 76 % | 62 % |

$$h'_v = \sigma \left(\sum_{w \in \mathcal{N}(v) \cup \{v\}} \alpha_{vw} W h_w \right) \quad (2)$$

giving us for each node v the features h'_v that will be fed to the next GCL (σ denotes a generic non-linearity, in our case a ReLU).

- *GraphSage layer* [15]. It is similar to GCN (see below) but provides a more general framework to customize the steps in the convolution. Specifically, the SAGEConv Pytorch implementation, that we use here, employs the sum operation both for aggregating and concatenating:

$$h'_v = W_1 h_v + W_2 \frac{1}{|\mathcal{N}(v)|} \sum_{w \in \mathcal{N}(v)} h_w \quad (3)$$

where W_1 and W_2 are the weights of the aggregation and concatenation functions. The final feature vector h'_v is then fed to a fully connected layer and normalised to h''_v after going through a non-linear activation function.

- *GCN layer* [11]. This is a spectral method, combining aggregation and concatenation through a single matrix W of weights:

$$h'_v = W^T \sum_{w \in \mathcal{N}(v) \cup v} \frac{\tilde{A}_{vw}}{\sqrt{\tilde{d}_v \tilde{d}_w}} h_w \quad (4)$$

with $\tilde{d}_v = 1 + \sum_w A_{vw}$ and $\tilde{A} = A + I_{|G|}$, where A is the adjacency matrix of the graph G , $I_{|G|}$ denotes the identity, and $|G|$ the number of nodes.

We now briefly describe our training. We first split our synthetic dataset into 7 volumes for training and validation and 1 volume for testing. In order compare performances, we test the model on the same volume used by [8]). Each volume is then split into $64 \times 64 \times 64$ voxel overlapping sub-volumes, and then the graph structure is built into each of them. Such structure is then fed individually into our model as expressed in Fig. 4. The 7 volumes are furtherly split into 60% for training and 40% for validation, after shuffling on the extracted $64 \times 64 \times 64$ sub-volumes. We summarize such splitting into Table 2.

We optimize the model using the Adam optimizer with a starting learning rate of 0.001, decaying exponentially with a rate of 0.96, and weight decay regularization of 0.001. The model is trained for 200 epochs and the batch size is set to 64 graphs. We employ cross-entropy loss and, since the classes are unbalanced, we weigh the contribution of each class by the inverse of its mean percentage occurrence among the 8 dataset volumes. Once the model is trained, to determine the accuracy, we collect the probabilities for all $64 \times 64 \times 64$ voxel sub-volumes extracted from the test volume, and we reconstruct $6512 \times 512 \times 512$ voxel probability volumes, one for each class. In order to better segment the overlapping regions between sub-volumes, we reconstruct the probability volume by summing the probabilities of overlapping voxels. The final $512 \times 512 \times 512$ voxel segmented volume is obtained by assigning to each node the highest probability class among the 6 probability volumes.

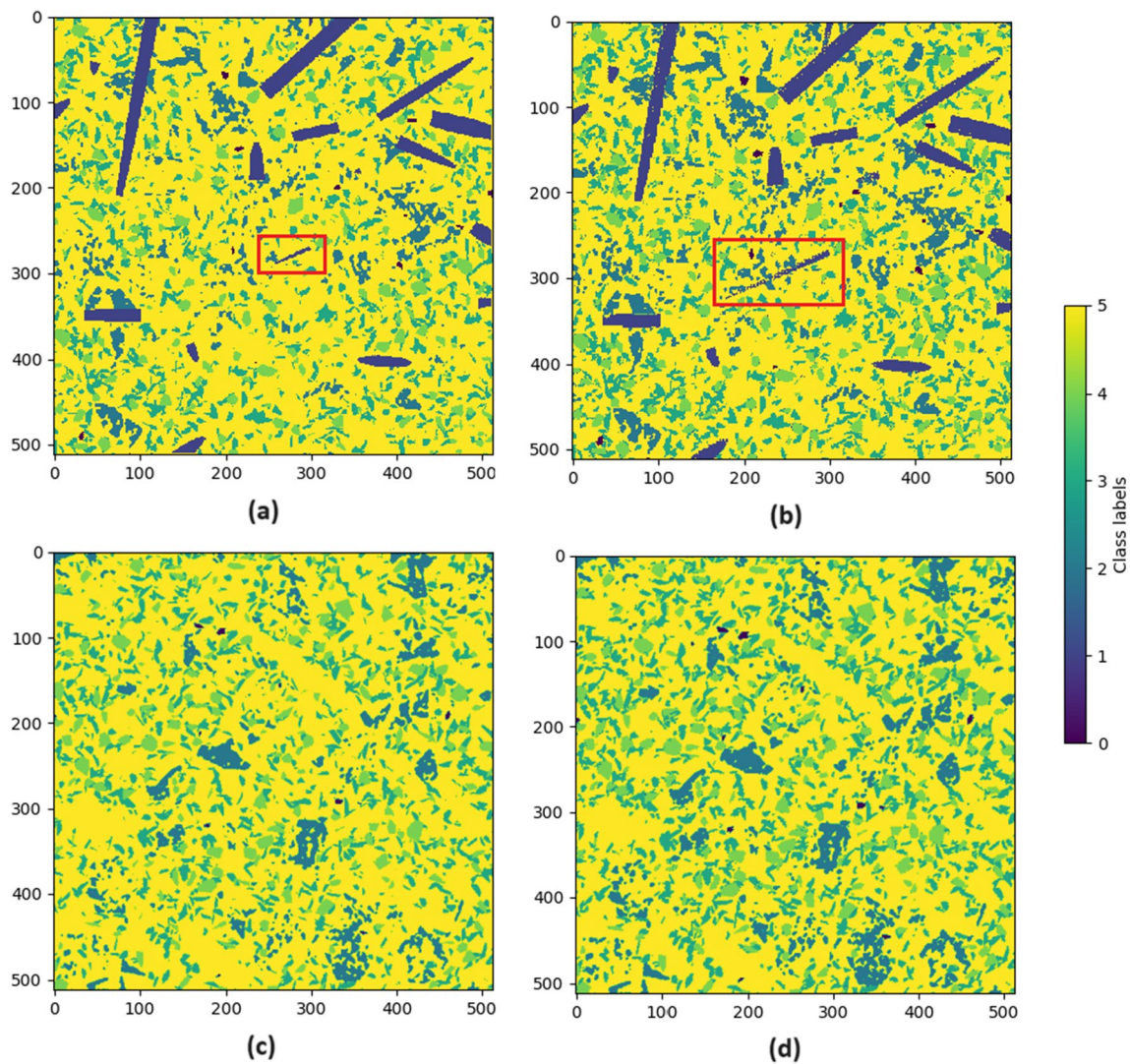


Fig. 5 A comparison between ground truths of slices in the synthetic dataset (left: **a**, **c**) and the predicted labels by our GCNN model (right: **b**, **d**). The color bar shows the corresponding color for each class label, from 0 to 5: voids, Al_2O_3 fibers, Intermetallics (IMs), eutectic Si, SiC particles, and Al matrix

5 Results

In order to evaluate the accuracy on the test set, we run 10 trainings with a fixed random seed and we compute the standard deviation corresponding to 95% confidence interval according to the common practice in the literature (see for example [16]). We measure the accuracy with the Dice score, a commonly used metric for image segmentation (see for example [6, 8]). The Dice coefficient rewards correctly segmented voxels and penalizes incorrect ones; it is defined in terms of True-Positive, False-Positive and False-Negative voxels as such:

$$\text{Dice} = \frac{2TP}{2TP + FP + FN} \quad (5)$$

In Table 3, we report the Dice score resulting from the evaluation of the GCNN model on the synthetic test dataset, and on the 4 manually-labeled slices of experimental data (95% confidence interval). The Dice score on experimental data is obtained by averaging first on the Dice scores of the 4 slices, then on the results of the 10 trained models.

As expected, since the model is trained on synthetic data, it performs worse on experimental data than on synthetic ones (see also [8]). This is because synthetic grayscale distributions are not perfectly capturing the complexity of the

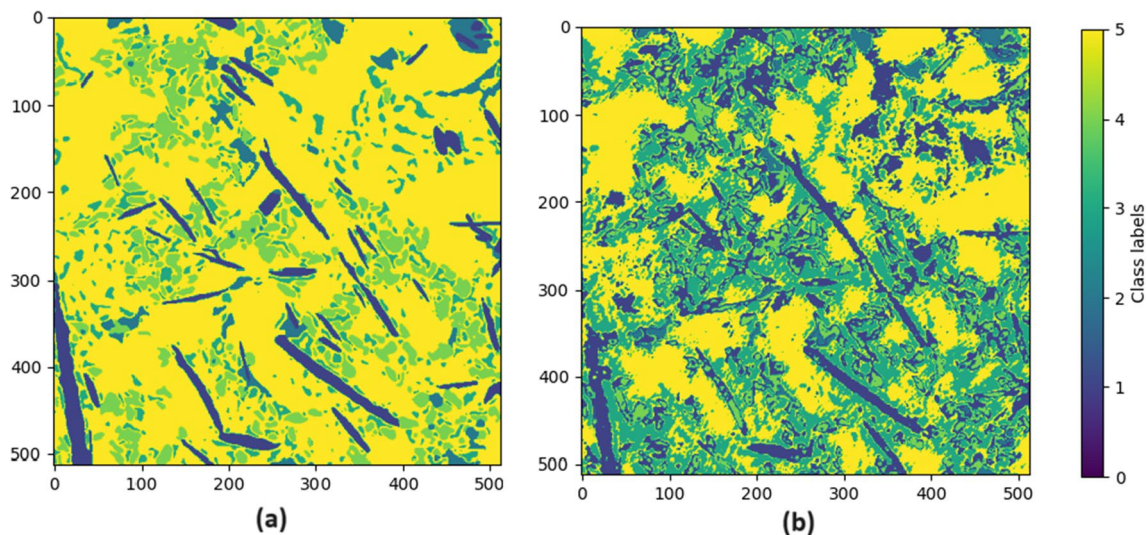


Fig. 6 A comparison between ground truths of a slice in the manually-labeled dataset (left: **a**) and the predicted labels by our GCNN model (right: **b**). The color bar shows the corresponding color for each class label, from 0 to 5: voids, Al_2O_3 fibers, Intermetallics (IMs), eutectic Si, SiC particles, and Al matrix

experimental ones, since they incorporate fewer grayscales. From Table 3, we observe that, on synthetic data, the model segments very reliably the different phases, with small standard deviation. We compare our model with the architecture of [8], whose Dice scores are reported in Table 4. From Table 4, we see a similar performance, but with far larger use of resources, with similar drop from synthetic to experimental data.

The number of trainable parameters of our GCNN model is 61406, while the Single U-Net model has 667446 parameters. We expect this will bring a considerably longer training time and RAM usage for DCNN versus GCNN once libraries are properly optimized.

In Fig. 5, we show synthetic slices in the test set with the ground truth values and the predictions by the GCNN model. As we can see, the segmentation is very good. The differences we can easily highlight concern fibers: in the red rectangle we show how a fiber is extended going from the true slice to the predicted one.

In Fig. 6, we also give an example of a manually-labeled experimental slice in the test set with the ground truth values and the predictions by the GCNN model. We notice that despite the accuracy error, as expressed in our previous Table 3, the GCNN correctly captures the geometry of the objects in the slice, thus giving a correct semantic understanding of the image.

6 Conclusions

We present a simple Graph Convolutional Neural Network, that performs a successful semantic segmentation of XCT volumes of an AlSi12CuMgNi Metal Matrix Composite reinforced with 7%vol Al_2O_3 short fibers and 15%vol SiC particles. Our method borrows a strategy developed for node classifications in different contexts (social networks [11], text analysis [13]). Our approach makes use of very limited computer resources and datasets; therefore it leads to a cost effective approach to semantic segmentation.

We believe our approach could be used in the future, together with the more standard autoencoder based Deep Learning methods, towards a new standard in microstructural analysis, via semantic segmentation in different material science questions.

Acknowledgements We thank our anonymous referees for valuable suggestions. The research ML, RF, FZ was supported by Gnsaga-Indam, by COST Action CaLISTA CA21109, by HORIZON-MSCA-2022-SE-01-101086123 CaLIGOLA and PNRR MNESYS. Funding for M. Lapenna from BAM Project QI Digital is acknowledged. FF acknowledges support from Unimore-DSCG-FAR-DIP-2023.

Author contributions ML, AT, FZ: conceptualization, data curation, formal analysis, investigation, methodology, writing-original draft. FF, RF, GB: supervision, writing-review and editing.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability Samples of the synthetic and experimental datasets discussed and classified in the paper are available together with the full code at: <https://github.com/michelalapenna/GNN-for-segmentation-of-XCT>.

Declarations

Ethics approval and consent to participate Not applicable.

Competing interests The authors declare that no Conflict of interest may have influenced the work in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Choudhary K, et al. Recent advances and applications of deep learning methods in materials science. *npj Comput Mater.* 2021;8:1–26.
2. Du W, et al. Automated detection of defects with low semantic information in x-ray images based on deep learning. *J Intell Manuf.* 2020;32:141–56.
3. Wong VWH, Ferguson M, Law KH, Lee Y-TT, Witherell P. Automatic volumetric segmentation of additive manufacturing defects with 3d u-net; 2021. [arXiv:2101.08993](https://arxiv.org/abs/2101.08993).
4. Fotos G, Campbell A, Murray P, Yakushina E. Deep learning enhanced watershed for microstructural analysis using a boundary class semantic segmentation. *J Mater Sci.* 2023;58:1–21.
5. Sarvamangala DR, Kulkarni RV. Convolutional neural networks in medical image understanding: a survey. *Evol Intel.* 2021;15:1–22.
6. Strohmann T, et al. Semantic segmentation of synchrotron tomography of multiphase al-si alloys using a convolutional neural network with a pixel-wise weighted loss function. *Sci Rep.* 2019;9:19611.
7. Evsevlev S, Paciornik S, Bruno G. Advanced deep learning-based 3d microstructural characterization of multiphase metal matrix composites. *Adv Eng Mater.* 2020;22:1901197.
8. Tsamos A, Evsevlev S, Fioresi R, Faglioni F, Bruno G. Synthetic data generation for automatic segmentation of x-ray computed tomography reconstructions of complex microstructures. *J Imaging.* 2023;9:22.
9. Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Process Mag.* 2017;34:18–42.
10. Bronstein MM, Bruna J, Cohen T, Velicković P. Geometric deep learning: grids, groups, graphs, geodesics, and gauges; 2021. [arXiv:2104.13478](https://arxiv.org/abs/2104.13478).
11. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks; 2017. [arXiv:1609.02907](https://arxiv.org/abs/1609.02907).
12. Zachary WW. An information flow model for conflict and fission in small groups. *J Anthropol Res.* 1977;33:452–73.
13. McCallum A, Nigam K, Rennie J, Seymore K. Automating the construction of internet portals with machine learning. *Inf Retrieval.* 2000;3:127–63.
14. Velicković P, et al. Graph attention networks. *International Conference on Learning Representations*; 2018.
15. Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs; 2018. [arXiv:1706.02216](https://arxiv.org/abs/1706.02216).
16. Genç A, Kovarik L, Fraser HL. A deep learning approach for semantic segmentation of unbalanced data in electron tomography of catalytic materials. *Sci Rep.* 2022;12.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com