Instance Segmentation of Fibers from Low Resolution CT Scans via 3D Deep Embedding Learning

Tomasz Konopczyński*[13]
Tomasz.Konopczynski@tooploox.com

Thorben Kröger[2]
Kroeger@volumegraphics.com

Lei Zhengr[1]
Lei.Zheng@medma.uni-heidelberg.de

Jürgen Hesser[1]
Juergen.Hesser@medma.uni-heidelberg.de

[1] Department of Radiation Oncology
University Medical Center Mannheim
Heidelberg University, Germany

[2] Volume Graphics
Heidelberg, Germany

[3] Tooploox
Wrocław, Poland

Abstract

We propose a novel approach for automatic extraction (instance segmentation) of fibers from low resolution 3D X-ray computed tomography scans of short glass fiber reinforced polymers. We have designed a 3D instance segmentation architecture built upon a deep fully convolutional network for semantic segmentation with an extra output for embedding learning. We show that the embedding learning is capable of learning a mapping of voxels to an embedded space in which a standard clustering algorithm can be used to distinguish between different instances of an object in a volume. In addition, we discuss a merging post-processing method which makes it possible to process volumes of any size. The proposed 3D instance segmentation network together with our merging algorithm is the first known to authors knowledge procedure that produces results good enough, that they can be used for further analysis of low resolution fiber composites CT scans.

1 Introduction

Reliable information about fiber characteristics in short-fiber reinforced polymers (SFRP) is much needed for process optimization during the product development phase. The influence of fiber characteristics on the mechanical properties of SFRP composites is of particular interest and significance for manufacturers [9]. The recent development of X-ray computed tomography (CT) for nondestructive quality control enabled the possibility to scan the materials and retrieve the 3D spatial information of SFRPs. Fiber extraction is the first step towards any further analysis of a SFRP material. However, the spatial resolution of a scan is a limiting factor which makes fiber extraction a difficult problem.

Acquiring scans in high resolution is time consuming and costly. Therefore, in this work we consider only scans acquired by a CT system with low (3.9 μm) resolution. The methods
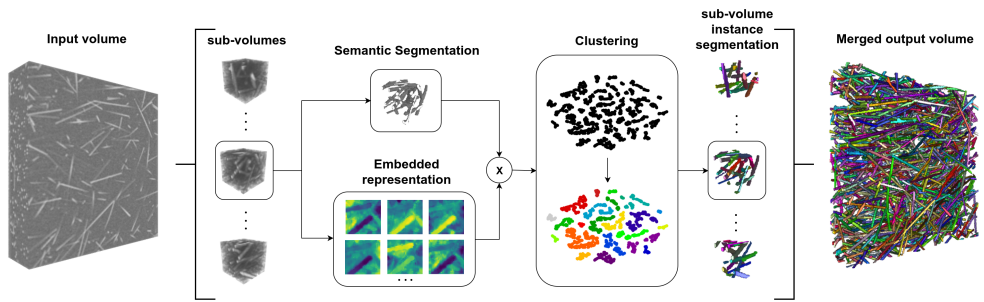
* work done while at Volume Graphics.

Figure 1:   Sketch of the proposed method. The network is processing overlapping sub-volumes of the input volume. For each sub-volume a semantic segmentation mask and an embedding is produced by a deep network. A clustering method is then applied on the segmented regions of the embedding representation producing clusters corresponding to individual fibers. Fibers are then mapped back to the spatial domain. The overlapping instance sub-volumes are then merged into an output volume.

currently in use are usually based on hand designed features. Since fibers can be described as long cylindrically shaped objects, the most widely used family of fully-automatic methods is based on Hessian eigenvalues. Using a set of Hessian based filters at a number of scales, a confidence map of fiber occurrence can be produced [8]. To extract individual fibers, a template matching [21] [5] or a watershed splitting and skeletonisation technique [24] [26] is then applied. However, the performance of these methods degrades severely if the resolution is too low and fails to produce meaningful results [15]. A deep learning method has already shown its superiority over Hessian based techniques to produce more accurate results for semantic segmentation of fibers at low CT resolution [16].

Deep learning architectures have been successfully applied to semantic segmentation problems for both natural 2D images and 3D CT volumes [18] [3]. Similar solutions have been found for the problem of 2D instance segmentation. Faster R-CNN [22] and the Mask R-CNN [11] architectures are examples of region-proposal-based techniques which are the state-of-the-art for common scene-understanding datasets like COCO [17] or ImageNet [4]. However, it is not clear how this approach can be extended to 3D volumetric data with densely packed objects like fibers in SFRP. This is why for our 3D problem, we have opted for alternative deep learning methods for instance segmentation. There are numerous works in which authors try to come up with different ideas for 2D datasets. An interesting idea that could be extended to 3D volumes has been proposed by [1] to reformulate the problem of instance segmentation into learning a mapping to watershed energy. Then, for the final output, a Watershed transform is applied to get the instances. Unfortunately, this method is not applicable to our problem, because fibers are usually too thin to find a border. Another promising idea proposed by [23] is to combine convolutional neural networks (CNN) with recurrent neural networks (RNN). The recurrent structures are used to keep track of objects that have already been found, and excludes these regions from further analysis by the algorithm.

In this work we propose a novel deep learning architecture for automatic extraction (instance segmentation) of fibers from low resolution 3D X-ray computed tomography scans of short glass fiber reinforced polymers. The sketch of the method is presented in Fig. 1.

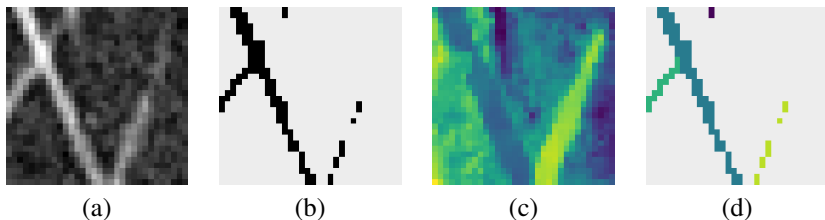|			
(a)	(b)	(c)	(d)

Figure 2: Visualization of the network outputs. (a) Slice of an input volume patch. (b) Corresponding slice from the semantic segmentation output. (c) Corresponding slice of one (out of many) feature map of the embedding output. The network learns to assign different, unique values (colors) to individual fibers. In this particular feature map all fibers are well separated. (d) The masked embedding by a semantic prediction for which a loss for embedding learning is computed. In the prediction phase it is also the input for the clustering step.

We explore and discuss the performance of the presented method achieved by training on a low resolution SFRP CT scan and compare it to a standard watershed splitting and skeletonisation technique. We test the importance of the semantic segmentation branch by replacing it with a ground truth semantic segmentation. To the best of our knowledge, this is the first attempt of using deep embedding learning for the task of instance segmentation on a 3D volumetric data. The proposed method is also the first to successfully retrieve single-fiber segmentation from a low resolution SFRP CT scan, while the outcome of the standard methods is producing unacceptable results. We base our method on an embedding learning approach [25] [2]. The idea is to use a special embedding layer which is placed at the end of a given deep network. The network is then trained by using a special loss function on the final embedding layer which encourages special structure in the embedding space: pixels belonging to the same class should be close, whereas pixels belonging to different classes should be far apart (in the Euclidean metric of the embedding space). The method as a learnable loss function has been first mentioned by [25], and was then used with some modifications in the deep learning architecture of [2] and [7]. These methods achieved competitive performance on 2D datasets compared with the R-CNN based state-of-the-art.

In the problem of fiber segmentation, the network will learn a mapping of each voxel and its surrounding from the input to an embedding space in which voxels belonging to one fiber are separated from voxels belonging to another. Unfortunately, there is one drawback to this method. Such a network is capable of processing only one small sub-volume of a volume at a time because of memory limitation. Each time the network processes a sub-volume it assigns an *arbitrary* index to a fiber region. Because of that, we can not do a simple merge as it is usually done for a semantic segmentation problem, where the output is a probability of being an object of a certain class. For a semantic segmentation mask one can take a simple average over overlapping regions in order to merge sub-volumes into a full volume. Therefore, in order to produce an instance segmentation for a full CT scan, we propose a post-processing algorithm, which merges the overlapping predictions of small blocks into a consistent prediction for the entire CT scan during the prediction phase.

2 Method

Similar to the work of [2], we have extended the Fully Convolutional Network (FCN) architecture [18] designed for semantic segmentation tasks to produce embeddings by using an
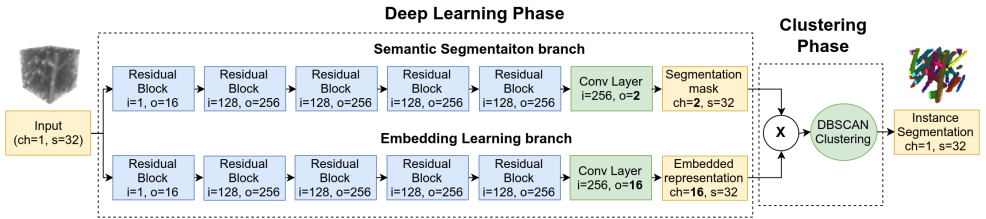
Figure 3: Detailed architecture of our network. *ch* is the number of channels of the sub-volume, *s* is the size of the sub-volume (here $s = 32$ means a sub-volume is of size $32 \times 32 \times 32$), *i* is the number of input channels, whereas *o* is the number of output channels from a convolutional layer or residual block. Both residual blocks and convolutional layers use 3D kernels. The kernel size is set to 3 for residual blocks and 1 for the final convolutional layer. Training is performed only on the deep learning phase. During prediction, the output from the embedding branch is masked by the output of the semantic segmentation branch and is processed by the DBSCAN algorithm producing the instance segmentation. The cross sign indicates the masking operation.

extra output. The extra output could be attached at the very end of the backbone of the semantic segmentation network as in [20], but in our setup we decided to use two sub-networks. One is responsible for computing the semantic segmentation mask, and the other for computing the embedding of voxels. The network can be trained separately for embedding and semantic segmentation using corresponding outputs or trained together for both tasks at the same time. We will refer to the two sub-networks as *semantic segmentation branch* and *embedding learning branch*. The semantic segmentation branch outputs a confidence map that a given voxel belongs to any fiber or not. The embedding learning branch outputs voxels coded in the embedding space. During the training phase, the architecture is trained only based on outputs from the semantic segmentation branch and the embedding learning branch using specified loss functions.

During the prediction phase a clustering step generates clusters corresponding to individual fibers. The clustering method is applied to the embedded voxels which have a high confidence of being a fiber based on the output from the semantic segmentation branch. The outputs from the two branches and the region on which the clustering is computed are presented in Fig 2. The clusters are then mapped back to the spatial domain creating a label volume, where each voxel is assigned an integer label corresponding to the fiber instance it is a part of. To make it possible to use on volumes of any size, we have proposed a greedy merging algorithm. The network produces outputs for overlapping sub-volumes of the input volume, which are then merged to a full volume. The detailed architecture of the network is presented in Fig. 3. In the following sections we will describe the above steps in more detail.

2.1   Semantic Segmentation

The semantic segmentation branch is a standard FCN for semantic segmentation. We have used an architecture that has been designed for the task of semantic fiber segmentation [16]. The output of the branch is penalized by the standard voxel-wise binary cross entropy loss
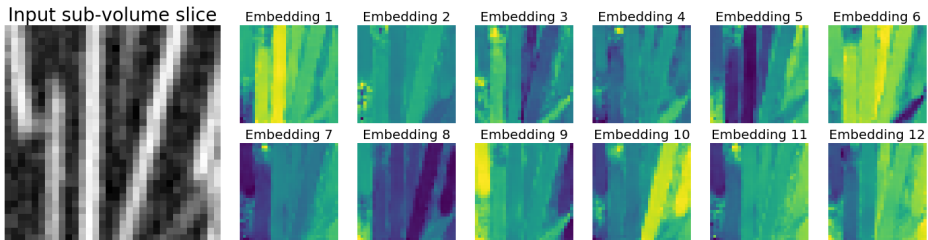
Figure 4: The input volume is represented by a number of embedded volumes at the embedding branch output. Here, slices of the first 12 embedding volumes corresponding to the input sub-volume slice are visualized. Note, that a good embedding will assign different set of colors in each embedding volume so that the clustering in the embedding space will be easy.

L_{CE}, as is common for semantic segmentation tasks. It is defined as:

$$L_{CE} = -[y \cdot log(\hat{y}) + (1 - y) \cdot log(1 - \hat{y})] \tag{1}$$

where y are the true binary labels, and $\hat{y}$ are the predicted labels. During the prediction phase, the output is thresholded at value 0.5 in order to produce binary masks. An example slice of an output of the branch is shown in Fig 2 (b).

2.2 Embedding Learning Loss

The output of the embedding branch is a representation of the sub-volume in an embedding space. The architecture of the branch is identical to the semantic segmentation branch. The only difference is the number of output channels in the final convolutional layer and the loss function. In the semantic segmentation task, the output is producing a volume with two channels, where one is reasoning on the foreground and the other on the background. In the embedding learning, the output has as many channels as the dimensionality of the embedding space (a hyperparameter in the algorithm). An example visualizing feature maps of the embeddings is shown in Fig. 4.

The loss function penalizes voxels of different instances that are too close to each other in the embedding space and encourages voxels of the same instance to be close. As a result, the network maps the voxels into the embedding space, such that voxels that belong to the same fiber should be placed next to each other and form easily separable clusters.

We find that the loss function introduced by [2] inspired by work of [25] and extended to 3D by us works best for our problem. Even though we have extended the problem to 3D, and have used data that contains a high number of objects compared to common scene-understanding problems, the method does not seem to be affected by that. The loss consists of three terms: L_v keeps voxels belonging to the same object close to each other, L_d which forces a minimal distance between clusters of different objects, and L_r which regularizes the

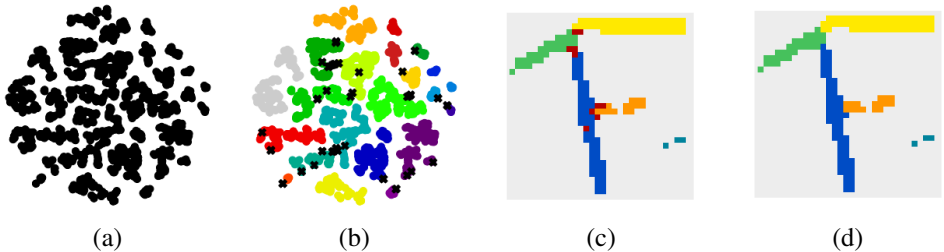| (a) | (b) | (c) | (d) |

Figure 5: 1. Visualization of the clustering steps of the method. (a) Masked embeddings form clusters in a multi-dimensional embedding space (visualized by t-SNE). (b) DBSCAN clusters the embedding representation and assign a different index (color) to each fiber (cluster) with black crosses for outliers. (c) Clusters are then mapped back to the spatial domain. Here a corresponding example slice of the mapping with red pixels for outliers. (d) The watershed algorithm is then applied as a post-processing step to fill the outliers.

cluster centers to be close to the origin. The terms are defined as:

$$L_v = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} [||\mu_c - x_i|| - \delta_v]_+^2 \tag{2}$$

$$L_d = \frac{1}{C(C-1)} \sum_{c_A=1}^{C} \sum_{c_B=1, c_A \neq c_B}^{C} [\delta_d - ||\mu_{c_A} - \mu_{c_B}||]_+^2 \tag{3}$$

$$L_r = \frac{1}{C} \sum_{c=1}^{C} ||\mu_c|| \tag{4}$$

where C is the number of objects in the ground truth patch (clusters), N_c is the number of voxels that corresponds to the object c, x_i is the embedding in the final embedding layer, μ_c is the mean of the embedding of object c, $||\cdot||$ is the L_2 norm, and $[x]_+ = max(0, x)$. The parameters δ_v and δ_d are used to control the desired positions of the clusters. The final loss for the embedding learning L_{embd} is a sum of the previous components.

$$L_{embd} = \alpha L_v + \beta L_d + \gamma L_r \tag{5}$$

where α, β and γ control the strength of the corresponding term. An example slice of an output of the branch is visualized in Fig 2 (c). Note, that the loss is computed only based on the voxels that belong to the foreground fibers. It is the task of the semantic segmentation branch to find the correct position of the fibers.

2.3   Clustering

As discussed in the previous section, the semantic segmentation output creates a confidence map that a given voxel belongs to any fiber or not. A clustering is then applied to the embedded voxels with a high confidence of being fibers. An example input slice of one of the feature maps of the embedding is shown in Fig 2 (d). In this work, we found DBSCAN [6] to work best on the SFRP dataset. In contrast to Mean Shift used in [2], DBSCAN does not make assumptions about the shape of the clusters. We apply clustering only in the prediction

phase because the instance segmentation loss function does not require the instance segmentation map. Note, that DBSCAN does not necessarily assign a label to all voxels. Voxels that were not assigned to any label are assigned as outliers. The clusters are then mapped back to the spatial domain creating an instance segmentation map. Outliers are extrapolated based on their neighborhood in the spatial domain by use of the watershed algorithm, using the clustering labels as seeds. An example visualization of the described steps with help of the t-SNE [19] is shown in Fig 5.

2.4 Merging

Finally, the inference is produced on small overlapping sub-volumes of the entire volume. Each sub-volume contains different label IDs for fibers, making it not clear which fiber is which. To overcome this problem we have designed a merging algorithm, which joins label IDs among the sub-volumes based on a spatial distance of fibers in the overlapped regions. The algorithm is applied at each sub-volume and processes recursively one fiber at a time, looking at neighboring sub-volumes with overlapping regions with objects being close to the fiber of interest. The merging procedure is described in more details in algorithm 1.

Algorithm 1 Merging algorithm

1: **procedure** MERGE(f, p) ▷ for a fiber f in a sub-volume p
2:     $N \leftarrow$ neighbour sub-volume of p ▷ N is a set of sub-volumes neighbouring with p
3:     **for** sub-volume n in N **do**
4:         $G \leftarrow$ fibers in n ▷ G is a set of fibers in a patch n
5:         **for** fiber g in G **do**
6:             $d \leftarrow D(f, g)$ ▷ Spatial distance between f and g
7:             **if** $d > \alpha$ **then**
8:                 $g_{id} \leftarrow f_{id}$
9:                 MERGE(g, n)

---

3 Experiments

3.1 Data

We have evaluated the proposed setup on two hand-annotated regions of low resolution CT scans of SFRP composites acquired by a Nikon MCT225 X-ray CT system from [15]. Scans exhibit typical artifacts and have low, but isotropic resolution. The parts from which the scans were acquired were manufactured by micro injection molding using PBT-10% GF, a commercial polybutylene terephthalate PBT (BASF, Ultradur B4300 G2) reinforced with short glass fibers (10% in weight). The volumes have been hand annotated with center lines and processed by a watershed algorithm to create the instance segmentation ground truth. Both volumes are cubes of dimension $62 \times 260 \times 260$ with approx. 6,500 fibers each. Fibers have a diameter of 10-14 μm (2-3 voxels) and are approx. 1.1 mm long. One scan is used for training, while the other is only used for testing.

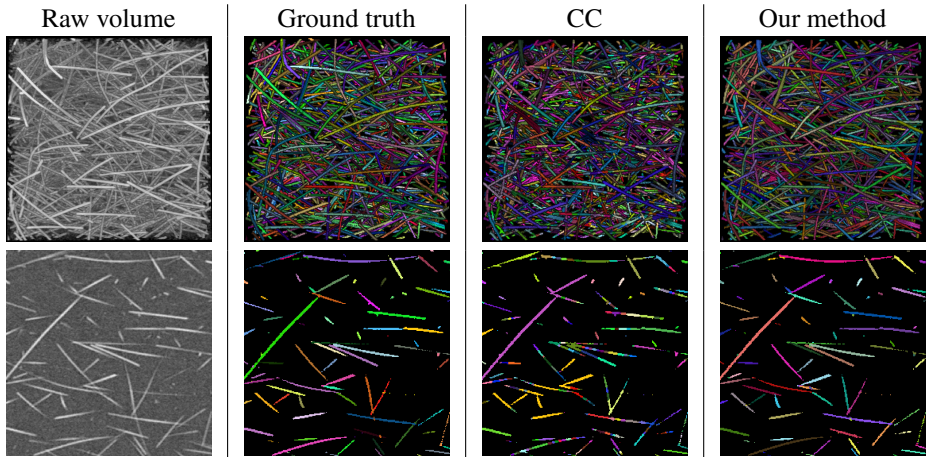| Raw volume	Ground truth	CC	Our method

Figure 6: Visualization of the testing volume and corresponding results. First row shows a 3D rendering of a volume. Second row shows one example slice of the same volume. First column is the input test volume (with a certain threshold to remove the epoxy background in the 3D rendering). Second column is the corresponding ground truth. Third column is the output of a standard connected component (CC) analysis. Fourth column is the output of our method. Fibers are colored semi-randomly based on the fiber ID.

3.2   Training details

The volumes have been normalized to have unit variance and zero mean. Additionally, most of the air voxels surrounding the specimen have been removed by a simple thresholding method. We have trained and evaluated the network on sub-volumes of $32 \times 32 \times 32$ from the training volume. The sub-volumes are randomly flipped and rotated (by 90, 180 or 270 degrees) during the training phase. As mentioned in the introduction, and shown in Figure 3, for backbones of both the semantic segmentation and embedding learning branch we have used the architecture proposed in [16] designed for semantic fiber segmentation. It is a 3D FCN with standard residual units [10] and batch normalization [13] but with no max-pooling to keep the resolution of the already very thin fibers.

The embedding learning is not stable, when trained from noise. Therefore, first we have trained the semantic segmentation branch for 20,000 iterations and saved the weights. Then, we have used the weights as an initialization for the embedding learning branch and trained it for another 20,000 iterations. The loss used for training the embedding learning uses the semantic ground truth masks.

It would also have been an option to share the embeddings and weights for both tasks. Such setup is reported to slightly increase the performance of both semantic and instance segmentation [21]. However, in our setup, we have found the above two-stage training to work better. We use 16 feature embedding maps and set α and β to 1 and γ to 0.001. Optimization has been done by using the Adam optimizer [14] with an initial learning rate set to 0.001. During the prediction phase, the algorithm processes overlapping $32 \times 32 \times 32$ sub-volumes of the test volume with an overlap of 16 in each direction. The post-processing merging algorithm merges the overlapping sub-volumes and produces the final instance segmentation volume.

Setup	Mean ARI	Merged ARI
Embedding Learning	0.9048	0.6529
Embedding Learning + true semantic	**0.9129**	**0.7817**
Connected Components	0.3537	0.2112
Connected Components + true semantic	0.3614	0.2534

Table 1: Comparison of our method with traditional connected components with and without provided ground truth semantic segmentation mask. Mean ARI are mean values over overlapping sub-volumes of the validation volume, while Merged ARI is the score computed over the entire volume after the post-processing merging step. The Dice score of the semantic segmentation mask from the semantic segmentation branch is 0.9784.

For a metric we have use the Adjusted Rand Index [12] to measure the performance of instance segmentation. We find it more informative in the context of SFRP data over the mAP. Defining the ground truth labels as clusters $C = \{C_1, ..., C_k\}$ and the corresponding predicted labels as clusters $\hat{C} = \{\hat{C}_1, ..., \hat{C}_l\}$, the Adjusted Rand Index R_a is:

$$R_a(C, \hat{C}) = \frac{\sum_{i=1}^{k} \sum_{j=1}^{l} \binom{m_{ij}}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3} \tag{6}$$

where $m_{ij} = |C_i \cap \hat{C}_j|$, $t_1 = \sum_{i=1}^{k} \binom{|C_i|}{2}$, $t_2 = \sum_{j=1}^{l} \binom{|\hat{C}_j|}{2}$, $t_3 = \frac{2t_1 t_2}{n(n-1)}$, and n is the number of voxels in the volume. The Rand Index varies from 0 to 1, where 1 means a perfect match between the algorithm output and the ground truth mask.

3.3　Results

We have compared our method to a standard skeletonization followed by connected component analysis and the Watershed method [26]. In the method, a binary erosion is first applied on the semantic mask, which serves as seeds after connected component analysis for a watershed segmentation algorithm. See Fig. 6 for a visual comparison. We have also evaluated the importance of a good semantic segmentation mask. We provide results for both our method and connected components given the semantic segmentation computed by the semantic segmentation branch as well as using the ground truth semantic segmentation.

Therefore we compare four different setups. Our *Embedding Learning* method using the final instance segmentation produced given the semantic segmentation mask from the semantic segmentation branch. *Embedding Learning + true semantic* which is our method but using ground truth semantic segmentation mask instead of the one produced by the methods branch (which is not ideal). *Connected Components* and *Connected Components + true semantic* is the connected component method used either on the output of the semantic segmentation branch or the ground truth semantic mask.

We provide two results in Table 1 for each setup. In the first column, the mean ARI is the mean ARI of all the sub-volumes in the test volume without the merging step. In the second column one can see the score computed over the entire volume after the post-processing merging step which we call a merged ARI. We report the ARI score only for the voxels that belong to the ground truth instance segmentation mask. Including the background voxels would artificially increase the score.

While the standard method clearly fails even when using the true semantic segmentation mask, the proposed method produces meaningful results in all cases. When reasoning on small overlapping patches the proposed method achieves 0.9048 average ARI score. The merging algorithm has trouble with ambiguity of two neighboring outputs and favors merging over splitting. This results in merging two fibers into one, when they are too close to each other. After the merging post-processing step the ARI score decreases to 0.6529.

4   Conclusions

In this work, we proposed a deep 3D fully convolutional architecture together with a set of post-processing steps for a problem of single fiber segmentation from CT scans of SFRP. We extend a less common approach of embedding learning for the task of 3D instance segmentation. We explain in detail the steps of the method together with a post-processing and a merging procedure. We show that we are better than the traditional skeletonization - watershed method. We expect our findings to be applicable to a wide variety of volumetric data and not only to fiber composites.

References

[1] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. *Conference on Computer Vision and Pattern Recognition*, pages 2858–2866, 2017.

[2] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551.*, 2017.

[3] P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, and W. H. Sommer. Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 415–423, 2016.

[4] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. Density-based spatial clustering of applications with noise. *Int. Conf. Knowledge Discovery and Data Mining*, 1996.

[6] T. Fast, A. E. Scott, H. A. Bale, and B. N. Cox. Topological and euclidean metrics reveal spatially nonuniform structure in the entanglement of stochastic fiber bundles. *Journal of materials science*, 50(6):2370–2398, 2015.

[7] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277.*, 2017.

[8] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *In International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1998.

[9] S. Y. Fu and B. Lauke. Effects of fiber length and fiber orientation distributions on the tensile strength of short-fiber-reinforced polymers. *Composites Science and Technology*, 56(10):1179–1190, 1996.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *International Conference on Computer Vision*, pages 2980–2988, 2017.

[12] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] T. Konopczyński, J. Rathore, T. Kröger, L. Zheng, C. S. Garbe, S. Carmignato, and J. Hesser. Reference setup for quantitative comparison of segmentation techniques for short glass fiber ct data. *Conference on Industrial Computed Tomography*, 2017.

[16] T. Konopczyński, D. Rathore, J. Rathore, T. Kröger, L. Zheng, C. S. Garbe, S. Carmignato, and J. Hesser. Fully convolutional deep network architectures for automatic short glass fiber semantic segmentation from ct scans. *Conference on Industrial Computed Tomography*, 2018.

[17] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, and C. L. Zitnick. Microsoft coco: Common objects in context. *European conference on computer vision*, pages 740–755, 2014.

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[19] L. V. D. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9:2579–2605, 2008.

[20] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool. Fast scene understanding for autonomous driving. *arXiv preprint arXiv:1708.02550*, 2017.

[21] P. Pinter, B. Bertram, and K. A. Weidenmann. A novel method for the determination of fibre length distributions from uct-data. *Conference on Industrial Computed Tomography*, 2016.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, pages 91–99, 2015.

[23] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. *European Conference on Computer Vision*, pages 312–329, 2016.

[24] R. M. Sencu, Z. Yang, Y. C. Wang, P. J. Withers, C. Rau, A. Parson, and C. Soutis. Generation of micro-scale finite element models from synchrotron x-ray ct images for multidirectional carbon fibre reinforced composites. *Composites Part A: Applied Science and Manufacturing*, 91:85–95, 2016.

[25] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[26] X. Zhang, D. Li, W. Yang, J. Wang, and S. Liu. A fast segmentation method for high-resolution color images of foreign fibers in cotton. *Composites Part A: Applied Science and Manufacturing*, 78(1):71–79, 2011.