



HAL
open science

Instance segmentation of 3D woven fabric from tomography images by Deep Learning and morphological pseudo-labeling

Samy Blusseau, Yanneck Wielhorski, Zyad Haddad, Santiago Velasco-Forero

► **To cite this version:**

Samy Blusseau, Yanneck Wielhorski, Zyad Haddad, Santiago Velasco-Forero. Instance segmentation of 3D woven fabric from tomography images by Deep Learning and morphological pseudo-labeling. 2022. hal-03345132v2

HAL Id: hal-03345132

<https://hal.science/hal-03345132v2>

Preprint submitted on 13 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Instance segmentation of 3D woven fabric from tomography images by Deep Learning and morphological pseudo-labeling

Samy Blusseau^{a,*}, Yanneck Wielhorski^b, Ziad Haddad^a, Santiago Velasco-Forero^a

^a*Mines Paris, PSL University, Centre for Mathematical Morphology
35, rue Saint Honoré - F-77305 Fontainebleau Cedex, France*

^b*Safran Aircraft Engines, Rond-point René Ravaud - Réau, 77550 Moissy-Cramayel, France*

Abstract

In the field of composite materials, mesoscale modelings based on X-ray computed tomography are becoming ever more widespread. This tool, aiming to increase the fidelity of the descriptive modeling of textile geometry for Finite Elements Analysis (FEA), requires image processing to identify the different objects within the material. In the present study, we propose a novel Deep Learning based segmentation of yarns from tomographic images aiming to provide a complete descriptive modeling of fabrics. The instance segmentation is achieved through an original two-step approach: (i) the determination of labeled yarn paths, by a tracking algorithm on detected 2D points, based on custom neighbour rules (distance and slope), and regression of parametric curves onto selected points, and (ii) a semantic segmentation of the yarn sections. For the second step, in absence of manual labeling of the yarn envelopes, we propose the use of morphological pseudo-labeling for training a Deep Convolutional Neural Network (DCNN), in which the yarn sections are represented by their distance transform. This approach is applied on two samples of a dry 3D woven ply-to-ply angle-interlock (at low and high compaction levels).

Keywords: Textile composites, Material modeling, Deep Learning, CT analysis

1. Introduction

The growing interest of composite materials in many industrial fields is mainly motivated by their relevant mechanical properties combined with a quite low density. There exists a great variety of composite materials according to the kind of fibrous reinforcement and polymer employed for the impregnation which is then cured until final consolidation. Before the impregnation and the curing steps, the weaving and forming processes represent also crucial steps of the global composite material manufacturing. Indeed, while the weaving could lead to miscellaneous topological mistakes, forming could also lead to undesired phenomena such as waviness, wrinkles or locking [1, 2, 3, 4]. Such phenomena are very linked to the type of reinforcement and could have

*Corresponding author.

Email address: samy.blusseau@minesparis.psl.eu (Samy Blusseau)

a great influence on the mechanical properties and on the lifetime (service time). It is important to add that such composite reinforcements could be described at three different scales for predicting textile deformation during forming [5]: at the microscale, relative to the fibers; at the mesoscale, corresponding to the yarn scale; or at the macroscale, in which textile is described as an homogenized material with continuous finite elements.

Thus, the development of effective computational tools enabling to model textiles and predict their physical properties (*e.g.*, mechanical and thermal behaviors) is required. Moreover, these computational tools provide a framework allowing to study sensitive parameters such as woven patterns, yarn morphologies and fiber volume fractions. Overall, two main families of textile modelings could be distinguished [6]:

1. the *predictive approach*, which consists in purely numerical methods aiming to predict the morphology of the textile in a given state, and
2. the *descriptive approach*, which is composed of μ -CT image-based modeling methods where the virtual model is deduced from a volume of images of existing material by segmentation techniques.

In such approaches, the micro and sub-mesoscale (“intermediate scale”) modeling could be achieved in limited proportion compared to the whole material to avoid high computational costs. Hence, the common scale is the mesoscale for which the main issue is to determine accurately the yarn shape (its trajectory and envelope) during the forming.

In the *predictive approach*, the modeling of woven fabric at the fiber scale seems unrealistic because of the great quantity of entities entailing a large number of degrees of freedom. For this reason, a method based on virtual fibers (1 virtual fiber = a bundle of real fibers) produced a good description of micro-mechanisms by performing sub-mesoscale simulations [7, 8, 9, 10, 11, 12, 13, 14, 15]. Thus, the wide literature on woven fabric modeling leads to meso and sub-mesoscale textile generating softwares developed for two decades such as WiseTex [16, 17], DFMA [7, 18], TexGen [8] and Multifil [11, 15]. Hence, the “as-molded” (as-manufactured) state could be computed through simulations of the mesoscale forming process with the yarn mechanical behaviour [19].

The *descriptive approaches* mainly rely on high-resolution X-ray computed tomography (μ -CT) which is widely used nowadays in non-destructive control fields. This great tool was used for design and process modeling of aerospace composites [20] as it provides spatial information allowing to follow the yarn (fiber) life from its manufacturing to its forming process. For instance, yarn deformations induced by the manufacturing process were measured thanks to the Digital Image Correlation (DIC) [21]. Similarly, Digital Volume Correlation (DVC) has proven to be a suitable technique for establishing a novel differentiating tool based on full field measurements of 3D woven composites. This approach provided a quantifiable description of woven by assessing metric differences (yarn deformations) and topological differences (missing yarns) with respect to a reference one [22].

The segmentation method leading to a “digital material twin” can be classified as a region-based segmen-

tation since the main goal is to distinctly separate different classes of objects inside the images. However, nowadays, automated procedures for computer tomography data transfer into finite element models are missing in the literature [23]. Different mesoscale segmentation approaches are described in the literature, depending on the labeling of images. They range from a simple bimodal segmentation [24] (single global thresholding value for the whole volume to separate only resin/porous media from yarns), to multimodal segmentation using clustering [25, 26] or optimization algorithms [27]. Moreover, many studies have shown that segmentation requires a trade-off between quite high enough resolution (*e.g.*, from $1\mu\text{m}$ [28, 29] to $25\mu\text{m}$ [24, 27, 30]) and contrast [31] to highlight the orientation and/or shape of yarns.

Regarding the analysis of μ -CT tomographies, we shall distinguish between three categories of image processing tasks. First, the **object detection** consists in locating the yarns in images, without finding their boundaries. In this case, possible outputs may be the yarn centers or their bounding boxes. Second, the **semantic segmentation** produces a partition of the pixels (or voxels) of an image into several classes (*e.g.*, yarn, resin, porosity). This is a more complex and more accurate task, as it requires to find the object boundaries. Finally, the **instance segmentation** goes a step further, as different objects of the same class are labelled differently (*e.g.*, $yarn_1$, $yarn_2$, etc.). To achieve these tasks, many suitable methods are widely presented in the literature, as reviewed next.

The principal directions and the degree of microstructural anisotropy of yarns can be measured through image local descriptors like the structure tensor [25, 26, 28, 32, 33, 34]). This was successfully applied to separate the warp and weft yarns. However, this approach is efficient for suitable contrast and resolution. In some cases, for binder identification, additional information related to the voxel orientation is required [34].

A post-processing using mathematical morphology operations (closing, convex hull) is then necessary to close and fill the yarn cross-sections as the results obtained with these methods remain a binary mask. Overall, the degree of anisotropy and the average grey value have proven to be suitable to separate entities like the weft and warp yarn groups as well as the resin/air voxels. However, they do not directly give a complete identification of each tow (shape and label) requiring a full separation. Indeed, some issues remain when the yarn is in contact with its neighbour, especially for high compacted states leading to high fiber volume fraction. Even at high resolution, yarns compressed against each other look merged together. In such a case, defining proper contours of yarns accurately remains a complex task.

The idea is then to apply clustering algorithms to identify all the subdomains. The two-parameters K-means clustering algorithm has been studied and shown issues to emphasize yarn clusters [32]. Furthermore, a supervised scheme seems to be the most efficient for clustering classification [35]. So, many works have applied a method based on a machine learning clustering Gaussian Mixture Model (GMM) followed by a 3D mapping-smoothing algorithm ([25, 26, 34]). These studies showed that the complete yarn separations still require additional final steps (*e.g.*, alphaShape function and element orientation [26, 34] or manual user intervention on first binary image [25]).

Variational segmentation methods, deforming initial yarn envelopes built from the yarn paths, manually annotated, are also proposed [27, 36]. As the shape of the yarn cross section could be a priori roughly determined, an elliptical shape is chosen. Bénézech and Couégnat [27] proposed an image-based variational segmentation relying on an iterative global-local approach driven by a heuristic algorithm. Overall, the idea is an expansion method since they initialize the model with small ellipses which grow up just after a few number of iterations. The goal is to optimize the voxelized geometric model with respect to the μ -CT, through three terms: a) the gray levels, b) the local directions and, c) the interpenetrations between yarns (regularization term). Lately, Pidou-Brion and Le Guilloux [36] also proposed a variational mesh active method which fits the mesh of the yarn envelope by an iterative process. The forces to minimize are derived directly from the surface mesh of the yarn envelope all along the process. Sinchuk et al. [37] developed another variational approach consisting in finding an approximation of the initial image by minimizing the Mumford–Shah functional, also defined by three terms: a) the distance between the initial image and its approximation, b) an approximation of the smoothness within subregions and c) the total length of the edges.

Over the past few years, Deep Learning approaches, mainly Deep Neural Networks (DNNs) have yielded a new generation of image segmentation models with remarkable performance improvements on popular benchmarks resulting in what many regard as a paradigm shift in the field [38, 39]. It is noteworthy that most of these benchmarks include thousands or even millions of training data, composed of images with their respective ground-truth annotation [40]. So the creation of initial training datasets is an essential point since the main drawback is that it could be a very time-consuming procedure and operator dependent when they are manually generated.

Thus, Deep Learning was successfully used to semantic segmentation separating weft, warp, resin and void/pore phases inside 2D woven and 3D woven reinforcements [37, 41, 42]. The authors explored convolutional neural networks like U-Net [37] and Residual architectures [41, 42]. Moreover, U-net was also applied [43] to perform a semantic segmentation of a rod-shaped SiC–SiC composite, identifying four material phases: fibers, matrix, pores, and environmental barrier coating (EBC).

However, since the semantic segmentation is devoted to separate the different phases with the composites, it does not allow to label individually each sub-component of these phases. This more accurate segmentation is known as **instance segmentation**, since it does not only seek which pixels belong to a yarn, but also to identify that there are several *instances* of yarns in each slice of the tomography. This kind of segmentation could be performed either directly from the neural network [44] or by combining semantic segmentation with a suitable post-processing [42, 45].

Recently, Mendoza et al. [44] performed instance segmentation on a 3D woven by training a Mask R-CNN network on synthetic, but realistic, images. The idea was to train the Mask R-CNN, devoted to identifying yarn outline with few key-points, with images from FE-models converted into “pseudo-CT” images thanks

to another neural network (U-Net). Indeed, the U-Net was trained on simple labels (yarn path manually annotated and elliptical section hooked up to it) to perform an “inverse segmentation” since the simple images are converted into “pseudo-CT” images. Note that this two-steps approach alleviates the issue of the training dataset usually manually generated [37, 41, 42, 46]. Another kind of data augmentation was performed [42] in applying a random pixel rotation and translation on the training images.

Finally, very lately, Ali et al. [42] and Sinchuk et al. [45] proposed instance segmentation frameworks combining different DCNN for a first step of semantic segmentation followed by the watershed technique [47, 48] during the post-processing, separating connected yarns.

Note that although some aforementioned works are devoted to perform instance segmentation only on dry fabrics [42, 44, 45], they are very useful and suitable to a better analysis of textile composites. Indeed, the composite reinforcement modeling at the mesoscale (in both predictive and descriptive approaches) improves the overall knowledge of textile composite manufacturing in assessing the mechanical response of the textile under various external loads [14, 19, 49, 50] as well as the preform behavior during the forming [15] process. Furthermore, a better description of the textile geometry could also enhance the knowledge of the injection process by determining for instance the permeability of the preform [51].

In this paper, we address instance segmentation on dry 3D woven composed of carbon fiber yarns, especially on layer-to-layer angle interlock as described in section 2.1. This image processing is performed by combining semantic segmentation, realised thanks to a trained U-Net (see section 2.3), with a suitable post-processing explained in section 2.4. This study deals with two distinct goals: on the one hand, the identification of yarn paths and, on the other hand, the yarn cross sections where the results are respectively shown in section 3 and section 4. Finally, some discussions are brought in section 5. Concerning the contour detection, it is noteworthy that to predict the yarn envelopes without any prior manual annotation, the pseudo-labels provided by a morphological approach are used as a training dataset. Regarding the quantitative evaluation of our methods, the detection of the yarn centers is assessed by classical metrics, namely: precision, recall and F_1 score; and the reconstruction of yarn paths by the Hausdorff distance. For the instance segmentation of the yarn envelopes, only a qualitative assessment is proposed, in absence of a reference manual annotation.

2. Materials and methods

2.1. Tomographic volume of 3D woven fabrics

The 3D woven fabric studied in this work is a ply-to-ply angle-interlock, composed of 75 carbon fiber yarns: 39 warp and 36 weft. All yarns are of the same type and size. The warp yarns are distributed alternately in a sequence of 4 and 3 yarns in two consecutive columns. Similarly, two consecutive weft columns are composed of 5 and 4 yarns respectively. In this sample, there are 11 warp planes and 8 weft

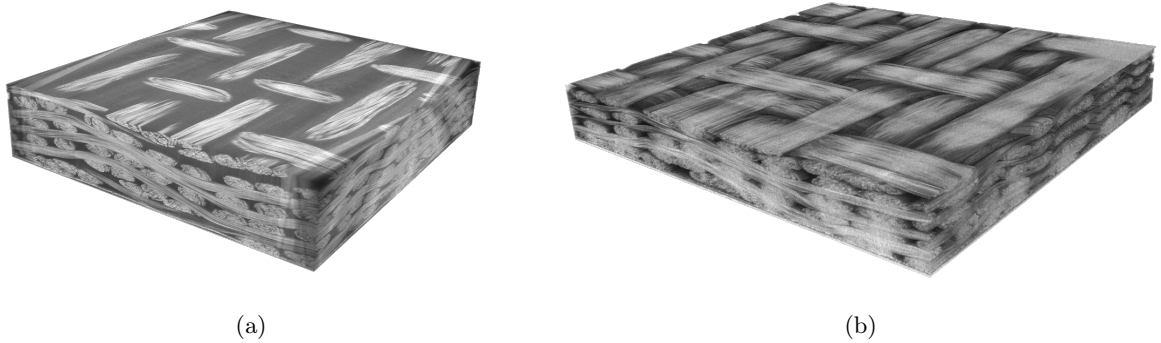


Figure 1: Images from high-resolution X-ray computed tomography of a 3D woven at two different compaction levels: (a) Initial non-compacted and (b) compacted states.

columns. The sample was scanned with a GE Phoenix-Xray tomograph (GE v|tome|x L300) at a resolution of $20\mu\text{m}$ (voxel size). Two samples from this woven pattern were studied at two different compaction steps: a *non-compacted state* (see Figure 1a), and a half-thickness sample (*i.e.*, 50% thickness of the initial non-compacted state), called *compacted state* (see Figure 1b). The image sizes are $1798 \times 1940 \times 447$ voxels and $1725 \times 1545 \times 255$ voxels respectively for the non-compacted and the compacted samples. Note that in these dry preforms the background is significantly darker than warp and weft yarn gray levels.

2.2. Data labeling

For both tomographies described in the previous section, the **yarn path** (*i.e.*, the centers of the yarns) were available thanks to a manual labeling. Indeed, each warp and weft yarn path had been created by first clicking the center point of each yarn every 30 slices. Then, the complete path was interpolated by B-spline functions, as shown in Figure 3. For the non-compacted sample, 39 warp and 36 weft yarns were annotated. Concerning the compacted sample annotations are available for 32 warp and 36 weft yarns.

Concerning the **contours of the yarn sections**, no labeling was available. Indeed, producing such labeling is much harder and time consuming by the manual way. Hence, we produced a pseudo ground-truth segmentation of the non-compacted sample by applying a learning-free baseline algorithm. This pseudo-labeling was then used to train a deep learning model, as described in section 2.3. The algorithm used to produce the pseudo-labeling for the segmentation of the yarn cross sections is based on Mathematical Morphology (MM). Detailing the whole pipeline is beyond the scope of the present paper but its main steps, illustrated in Figure 2, are as follows. First an adaptive morphological opening, as introduced in [52, 53], allows to identify the horizontal yarns in each slice, and then to remove them from the binary mask obtained beforehand by Otsu’s threshold [54]. Then simple morphological filters (fill holes, closings) can recover compact shapes. Finally, connected components composed of more than one yarn are identified based on a size criterion. They are split thanks to the watershed algorithm [47, 48] applied to the inverse of their

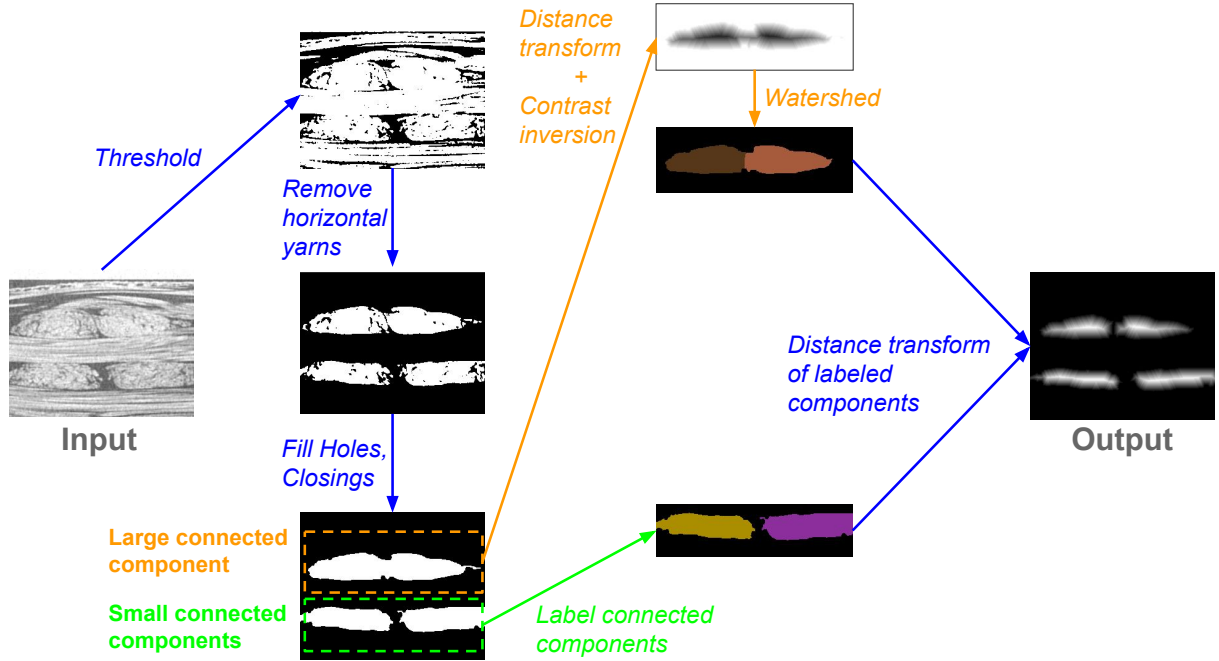


Figure 2: Morphological pipeline producing the pseudo-labeling of the yarns cross sections by their distance transforms.

distance transform (or distance function) [55], which is a very classical process in binary segmentation. Each single yarn is then represented by its distance transform. Recall that, given a metric d , the distance transform $D(I)$ (or function) of a binary image I associates to each pixel its distance to the nearest black pixel:

$$[D(I)](p) = \min \{d(p, q), I(q) = 0\}. \quad (1)$$

In our case, we used as metric the length of shortest path in the graph defined by a hexagonal lattice.

Note that this method produced satisfactory labeling on the non-compacted sample but not on the compacted one. For this reason, we restrict the segmentation of yarn envelopes by supervised learning to the non-compacted sample, as a proof of concept of the yarn representation that can be efficiently learned by a U-Net.

2.3. Processing tomographies with Deep Learning

2.3.1. Basics on deep learning

The first main step in the Deep Learning method is to train a dataset for the learning phase of the algorithm. Let $\{(x_k, y_k)\}_{1 \leq k \leq n}$ be a training dataset of pairs, where each x_k is a data-sample from an input space \mathcal{X} and y_k its corresponding label from the output space \mathcal{Y} . In our case, the x_k s are the slices of a

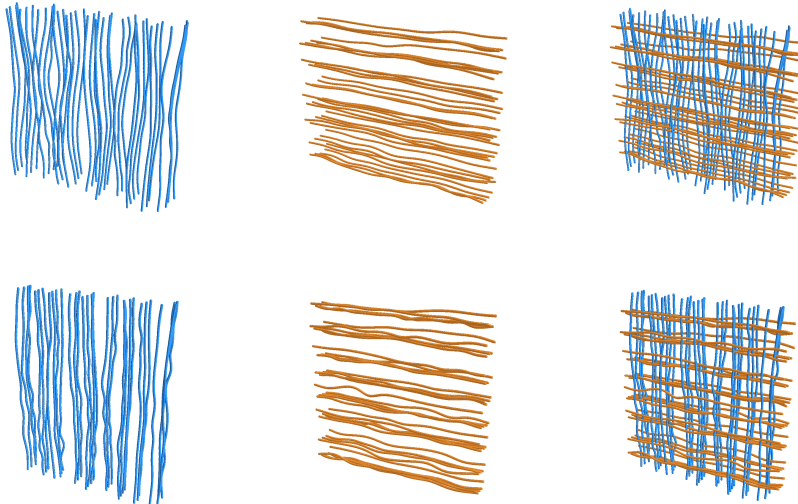


Figure 3: Ground truth yarn paths of the non-compacted dry preform (first row) and the compacted one (second row). Blue stands for warp paths and orange for weft.

tomography, like those of Figure 1, and the y_k s are images of the same size where the yarn paths or the yarn sections are highlighted. Learning from these data consists in using the examples $\{(x_k, y_k)\}_{1 \leq k \leq n}$ to build a parametric map $\phi : \mathcal{X} \mapsto \mathcal{Y}$ that accurately predicts the label y_{n+1} of any new data sample x_{n+1} , that is: $y_{n+1} \approx \phi(x_{n+1})$. A DNN, in its simplest form, is a *compositional* map that may be written as $\phi_{f;\theta} := f^{(d)}(g^{(d)} \dots f^{(1)}(g^{(1)}(\mathbf{X})))$ where each $f^{(l)}$ is a nonlinear function called *activation function* and each $g^{(l)}$ is usually an affine application defined by its weight matrix $\mathbf{W}^{(l)}$ and bias vector $b^{(l)}$: $g^{(l)}(\mathbf{Y}) = \mathbf{W}^{(l)}\mathbf{Y} + b^{(l)}$. Here θ denotes the set of affine parameters $[\mathbf{W}^{(1)}, b^{(1)}, \dots, \mathbf{W}^{(d)}, b^{(d)}]$ and $f = [f^{(1)} \dots f^{(d)}]$ the set of activation functions. The composed application $f^{(l)} \circ g^{(l)}$ is what we call a *layer* of the network. The term *deep* in DNN refers to neural networks with many layers, usually $d > 2$. As we can see from the equation above, the output of a layer becomes the input of the following one, except for the input and output layers, which are merely the input and output of the whole network. When the input of a layer is an image (or a stack of images) the linear part of the layer's affine transformation is actually a *convolution*, and therefore the matrix $\mathbf{W}^{(l)}$ is fully determined by a small number of parameters, called the *convolution kernel*. Such a layer is called a *convolutional layer*, and neural networks including such layers are referred to as *convolutional neural networks* (CNNs) [56].

CNNs can model complex non-linear relationships and have shown their goodness in different kinds of problems such as automatic speech recognition, image recognition, natural language processing, among others [57]. However, one of the fundamental points is the selection of an adequate *architecture* [58], *i.e.* the structure of composition between layers: number of layers, dimension of each layer's output, size of

the convolution kernels, type of activation functions. In the training process, we would usually tune the parameters θ so as to minimise the difference between the labels (ideal maps y_k) and the CNN outputs (estimated maps $\hat{y}_k = \phi_\theta(x_k)$) at any training instance x_k , such that the difference goes to zero as the number of samples N increases. For that, we define a loss function $l(y_k, \hat{y}_k)$ that represents the difference between the labels and the CNN output. In an ideal case, we would like to find θ^* such that:

$$\theta^* = \arg \min_{\theta} E_{\forall x_k \in \mathcal{X}, y_k \in \mathcal{Y}} [loss(y_k, \hat{y}(x_k))], \quad (2)$$

where $E_{\forall x_k \in \mathcal{X}, y_k \in \mathcal{Y}}$ denotes the expected value over all possible pairs of (x, y) , which is impossible to calculate in most of scenarios. However, we can compute an approximation, called *empirical risk* [59], by averaging the loss function on a large set of training examples $(x_k, y_k)_{1 \leq k \leq N}$,

$$\hat{\theta} = \arg \min_{\theta} \sum_{i=0}^N loss(y_k, \hat{y}(x_k)). \quad (3)$$

This minimization is usually done via *stochastic gradient descent* (SGD) [60]. SGD starts from a certain initial θ and then iteratively updates each parameter by moving it in the direction of the negative gradient with respect to the loss function. The computation of gradient with respect to the loss function is done via a direct application of the chain rule in networks, called *back-propagation* [61]. The term *stochastic* in SGD indicates that a random small number of training samples, called a *batch* is used in the gradient calculation. This reduces considerably the computational cost of the gradient evaluation. Additionally, by the law of large numbers, this stochastic gradient should be close to the full sample one, though with some random fluctuations. A pass of the whole training set is called an *epoch*. Usually, after each epoch, the error on a validation dataset is evaluated and when it stabilizes the training is complete.

2.3.2. Architecture

For the different tasks presented here, a Deep Convolutional Neural Network (DCNN) architecture, called U-Net [38], was used. The U-Net architecture was originally designed for biomedical image segmentation and has become very popular for its impressive performances. In the present work, the U-Net architecture is chosen as both addressed tasks (the reconstruction of yarn paths and of their envelopes) are viewed as image segmentation problems. A U-Net maps an input image to an output image expected to show a segmentation of the input. As shown in Figure 4, the computations performed by a U-Net follow two main steps. In the first step, it keeps extracting and combining features from the input image while reducing the spatial resolution by pooling filters at each layer. This is called the *contracting path*. In the second step, called *expansive path*, it increases back the resolution using the accumulated features to produce a new image of the same size as the input, where only the desired objects should be highlighted. More precisely, for the yarn paths detection task, the contracting path consists of the repeated application of two 3×3 convolutions

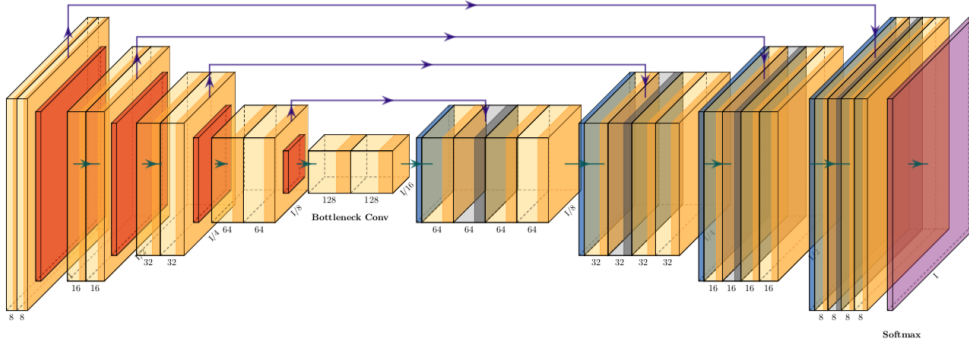


Figure 4: Architecture of the U-Net used for the yarn paths detection. The number of convolutional filters per layer is indicated.

with a rectified linear unit (ReLU) activation and a 2×2 max pooling operation for downsampling. At each downsampling step, the number of feature channels doubles. Every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. In total our architecture has 490,993 parameters for the detection of yarn centers. For the segmentation of the cross sections, the architecture was almost the same except that the first layer contained four filters instead of eight, and the last layer was a 1×1 convolution instead of a softmax layer as in Figure 4, totaling 122,953 parameters.

2.3.3. Training protocol

One of the challenges we faced to apply the deep learning approach is that we only dispose of two 3D images with their corresponding ground-truth, the yarn centers (see section 2.2). Hence, our training protocol is designed to avoid model over-fitting. First, we propose to extract 2D patches of size 256×256 pixels, with a random center position (see some examples in Figure 5b). The goal of random crops is to both prevent the selection bias and the redundancy of the data. We argue that allowing the overlap is essential for the network to learn features shared between patches. Secondly, we use one direction (warp or weft) during training and leave the other one for testing, as illustrated in Figure 5a. The direction used for training is itself split into training and validation slices. More specifically, the *training set* is composed of 10000 patches randomly selected from two thirds of the slices in one direction, and the *validation set* is composed of 1000 patches also randomly selected from the remaining third in the same direction. When not specified otherwise, training and test are run on images of the same level of compaction, but a mix can be done as it will be seen in section 3.3. During training, at the end of each epoch, the current state of the model is evaluated on the training set to give an idea of how well the model is *learning*, then the evaluation on the validation set tells how well the model is *generalizing*. As it has been explained in section 2.3.1, the

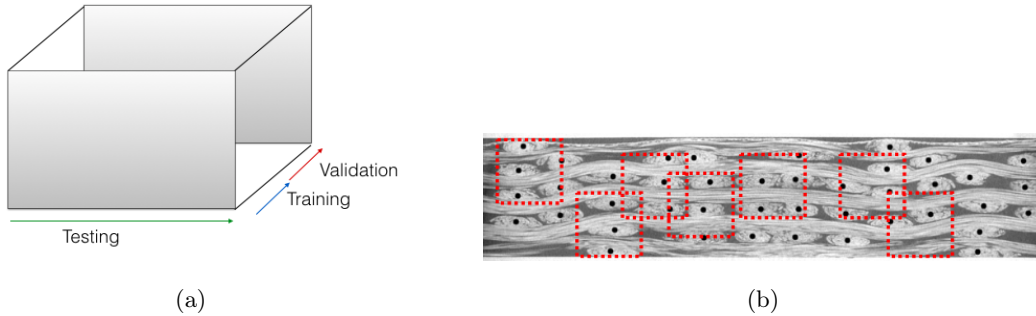


Figure 5: Illustration of the training protocol in the Deep Learning approach. (a) Partition of a tomography into training, validation and test slices. (b) Set of random patches superimposed on a slice.

training is stopped when the error on the validation set does not reduce after a given number of epochs [62], called *patience* parameter. In our experiments, the training process was performed on a single NVIDIA Titan XP GPU using ADAM optimizer [63]. For the detection of yarn centers, the patience parameter was fixed to ten epochs, the learning rate to 0.001 and the mini-batch size to 64. For the segmentation of yarn sections, these parameters were set to 20, 10^{-5} and one respectively. In the sequel, this procedure has been applied on both compacted and non-compacted samples for the yarn path detection (see section 2.3.4), and on the non-compacted sample only for the segmentation of sections (see section 2.3.5). Obviously, it could be noted that after training, the CNN can be applied to any tomography slice.

2.3.4. Detection of yarn centers

The reconstruction of the yarn paths boils down to detecting the location of each yarn in each slice and, from these locations, inferring as many paths as there are yarns in the tomography.

Labels: For this task, we had the necessary ground truth for supervised learning, namely the coordinates of the yarn paths in each slice. The training dataset used here is the yarn paths annotated manually as described in section 2.2. These coordinates are translated into binary images where the yarns' center is marked by white squares, as shown in Figure 6. The size of each square is taken equal to 25×25 pixels. Note that the pixel size of the ground truth has been chosen as a trade-off. Indeed, this size has been set not too small in order to avoid a bias of the model towards trivial predictions, *i.e.*, almost completely dark images, due to a strong area imbalance between objects to detect (white pixels) and background (dark pixels). Moreover, this size should be not too big either, so as to well approximate the yarn center and remain included in the smallest yarn section. The neural network was thus trained to map a 2D input image showing yarns, to an output image where the yarn paths were marked in white on a black background.

Loss function: A performance measure commonly used for evaluating segmentation masks is the Jaccard index, also called the intersection-over-union (IoU) score [64]. In the case of a binary ground truth

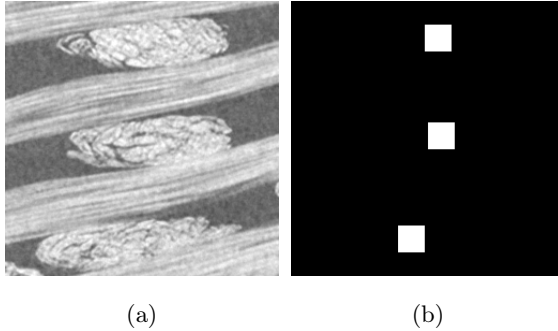


Figure 6: Pair of images (patches) for the training: (a) raw image and (b) binary image with the ground truth markers.

image y and its gray-scale estimation \hat{y} , this can be written as follows:

$$J(y, \hat{y}) = \frac{\epsilon + \sum_i y[i] \cdot \hat{y}[i]}{\epsilon + \sum_i^N (y[i] + \hat{y}[i] - y[i] \cdot \hat{y}[i])}, \quad (4)$$

where $y[i]$ and $\hat{y}[i]$ denote the values of images y and \hat{y} at pixel i , N is the number of pixels in each image, and ϵ is a very small number preventing division by zero. Accordingly, the corresponding loss function to be employed in empirical risk minimization (3) is $loss(y, \hat{y}) := 1 - J(y, \hat{y})$.

2.3.5. Segmentation of yarn sections

The protocol followed here is very similar to the one in the detection of yarn paths. The training and validation sets are generated the same way.

Labels: As specified in section 2.2, no manual labeling was available for the cross-sections of yarns. Therefore, in order to assess the potential of the supervised training of a CNN on the segmentation of yarns, we used as ground-truth the pseudo-labels obtained by the morphological algorithm briefly described in section 2.2. Hence, the label images used to train the network here were the distance transforms of yarns, as shown in Figure 2.

Loss function: Here the loss function is not the Jaccard index anymore but the mean square error (MSE):

$$loss(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y[i] - \hat{y}[i])^2 \quad (5)$$

where N is the number of pixels in the input y and output \hat{y} . Here y is an input slice and \hat{y} the distance transform used as a label (see Figure 2).

2.4. Post-processing: From 2D semantic segmentation to 3D instance segmentation

2.4.1. Recovering yarn paths

The aim of this section is to determine the path of each yarn from the output of the U-Net trained to detect yarn centers (see section 2.3.4). This trained U-Net maps input tomographic slices to output images where the centers of yarns are highlighted more or less accurately by bright shapes. The first post-processing consists in extracting from this output image a set of coordinates representing the detecting yarn centres. This is done by applying Otsu’s adaptive threshold method [54], removing very small components thanks to an area opening [65], and finally compute the barycentre of the remaining connected components. Applying this to each slice in one direction results in a 3D point cloud, which we call the predicted yarn centers. As it is shown in section 3 (see for example Figure 9), most detected points are close to the actual yarn paths but some false positive and negative detections also occur. Hence, the second post-processing consists in recovering the 3D path of each yarn from the noisy cloud of independently detected points.

The procedure relies on two main steps. First, a **tracking** step identifies candidate clusters of points to be yarn paths. Second, a **parametric modeling** fits a parametric function to each cluster.

1. Tracking: This first step consists in building *tracks*, that is to say sequences of detected points that are likely to belong to the same yarn. A track is initialized by one detected point in one slice, and extended forward and backward by points from subsequent and previous slices, complying with a smoothness criterion. To illustrate this step, let’s detail one forward extension. Consider an incomplete track $[p_0, \dots, p_n]$, where each point p_i is identified by its coordinates (x_i, y_i, z_i) , the last coordinate z_i denoting the slice to which the point belongs and (x_i, y_i) its 2D coordinates within the slice. Suppose we want to extend the track to a point from a subsequent slice s indexed by $z^* > z_n$. We denote by $p^* = (x^*, y^*, z^*)$ the closest point to p_n in s and define $dX = |x_n - x^*|$, $dY = |y_n - y^*|$, $dZ = |z_n - z^*|$. Then p^* is appended to the track (that is, $p_{n+1} \leftarrow p^*$) if $\sqrt{dX^2 + dY^2} \leq \min(\alpha dZ, \rho_{\max})$ for some parameters $\alpha > 0$ and $\rho_{\max} > 0$. This means p^* needs to fall within the intersection of a cone pointing on p_n and of slope α , and a cylinder of central axis z_n and radius ρ_{\max} . If this condition is not fulfilled by any point in s , then we look further for one in a subsequent slice. The range of subsequent slices to search (step, furthest slice) may vary. If no extension is found in this range, then the forward tracking is over. The backward tracking works in the same way. By changing the initial point and the range of subsequent (or previous) slices, we manage to build many tracks and with high probability the desired yarn paths are among them.

2. Parametric modeling: Among the set of tracks built in the previous step, some may be redundant (*i.e.*, there might be several tracks for the same yarn) and others may be wrong, in the sense that they might not follow a yarn path. The goal of this second step is to keep only accurate, mutually different tracks, using as criteria their fit to a parametric model which is expected to suit yarns paths. Visualizing the tomography should tell what model seems realistic. In the images considered here, the yarn paths vary like sinusoids

in one dimension (height) and are almost constant in the other (width). Therefore we solve for each track $[p_0, \dots, p_n]$ two least squares problems:

$$(a_0, \dots, a_K, b_1, \dots, b_K)^* = \arg \min_{(a_0, \dots, a_K, b_1, \dots, b_K)} \sum_{i=0}^n \left(y_i - a_0 - \sum_{k=1}^K [a_k \cos(\frac{2k\pi}{T} z_i) + b_k \sin(\frac{2k\pi}{T} z_i)] \right)^2 \quad (6)$$

and

$$(c_0, \dots, c_N)^* = \arg \min_{(c_0, \dots, c_N)} \sum_{i=0}^n \left(x_i - \sum_{k=0}^N c_k z_i^k \right)^2, \quad (7)$$

where the period T , the number K of harmonics and the degree N of the polynomial are parameters to set. Only tracks achieving a small mean squared error for these two problems are kept. Besides, comparing the best fits of two tracks allows us to tell whether they correspond to the same yarn path or not. For instance, [Figure 7](#) shows two examples of yarn tracks (in blue) and their parametric fits (in orange). The polynomial's degree is $N = 3$, the number of harmonics is $K = 2$ and the period is $T = 1600$. Note that the left hand track (see [Figure 7a](#)) is discarded because of its poor fit to the sinusoidal model in the height dimension, whereas the right hand track is kept (see [Figure 7b](#)). In the latter, the parametric model is somehow a smoothed version of the track built on the initial noisy output. If one iteration of this process does not provide an adequate parameterization of all the yarn paths, it is possible to remove from the initial point cloud the points associated with the found fibers, and repeat the tracking and parametric modeling on the residual point cloud. As it is shown in [section 3](#), the initial point clouds output by our method are more or less noisy. When the point cloud is accurate enough, the model based post-processing may converge in one or two iterations. When it is noisier, this may take a few more iterations, including a user interaction to select the correctly found yarn paths before computing the residual point cloud.

Note that the proposed approach to recover yarn paths is fairly general and can be adapted to any kind of preform, as long as the x and y coordinates of the path can be expressed as functions of the z coordinate. Indeed, different parametric models can be used based on prior knowledge on the yarn paths, and their complexity can be increased by increasing the order of these models (that is, increasing the number of harmonics or the degree of polynomials). Furthermore, although the yarns inside the analysed woven patterns are with limited crimps, the method can be adapted to cases where a yarn cross-sections vanish in some slices due to high crimp. First, by increasing the values of parameters α and ρ_{max} , tracks with arbitrary large slopes can be built. Second, note that if a yarn center vanishes at some point, it also reappears later on, and is therefore likely to be appended to the current track with our strategy. Indeed, recall that if no candidate is found in one slice, it is searched in subsequent slices. Finally, instead of choosing a candidate as the closest point to the last point of the current track, we can relax the constraint and consider the k nearest neighbours, with k and integer larger than 1.

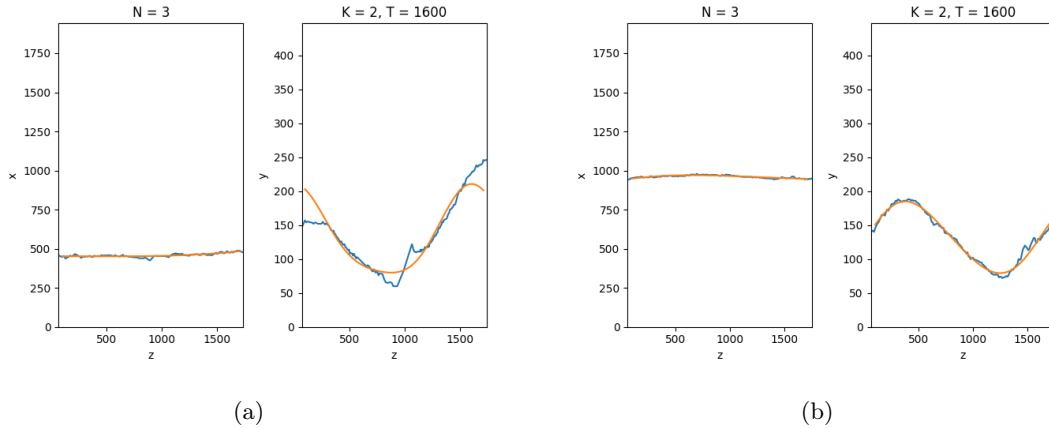


Figure 7: Two examples of tracks (in blue) and their parametric fits (in orange).

2.4.2. Recovering yarn envelopes

Here we describe how we achieve a 3D instance segmentation of the yarns from the 2D output images produced by the U-Net trained as explained in one direction in [section 2.3.5](#), and applied to the orthogonal direction. This U-Net maps an input slice to an image showing approximations of distance transforms at each yarn location. As these are not exact distance transforms, the outline of yarns cannot be obtained by selecting pixels with value one. However, selecting the pixels with positive values is sufficient to recover the connected components associated to the yarns, and therefore most contours except those separating touching yarns. Excessively large connected components are identified as touching yarns. Then, their contours are computed thanks to a watershed transform with markers, applied to the inverse of the output image. The intersection points between the parameterized yarn paths (described in [section 2.4.1](#)) and the connected components are used as markers for the watershed. This completes the step of semantic segmentation of each yarn in 2D slices. To turn this into a 3D instance segmentation, each segmented region in each slice is assigned the label of the yarn path intersecting it. If there is no such path, the region is discarded. This simple step produces 3D regions following the previously found paths and corresponding to the yarn envelopes. This result is then regularized by a morphological filter (oriented opening) to remove obvious segmentation errors. Finally, to obtain a 3D surface mesh from the labeled 3D image, the Marching Cubes algorithm [66] is applied, as implemented in Scikit Image [67]. The marching cubes algorithm provides a means to compute a triangular mesh approximation to the isosurface produced by each one of the segmented regions.

2.5. Overview of the method

The main steps of the method are summarized in [Figure 8](#). As illustrated, we address two tasks: first, find the path of each yarn and, second, their envelopes. In both cases, a U-Net is trained to process 2D

slices. On the one hand, for the first task, a manual labeling of the yarn centers is available. On the other hand, for the second task, a pseudo-labeling of the yarn sections computed from the morphological approach is used. Then, a post-processing of the 2D outputs of both U-Nets allows it to move from a 2D semantic segmentation to a 3D instance segmentation. The 3D yarn paths are recovered first, and then used to label the sections in each slice, producing a final 3D instance segmentation of envelopes.

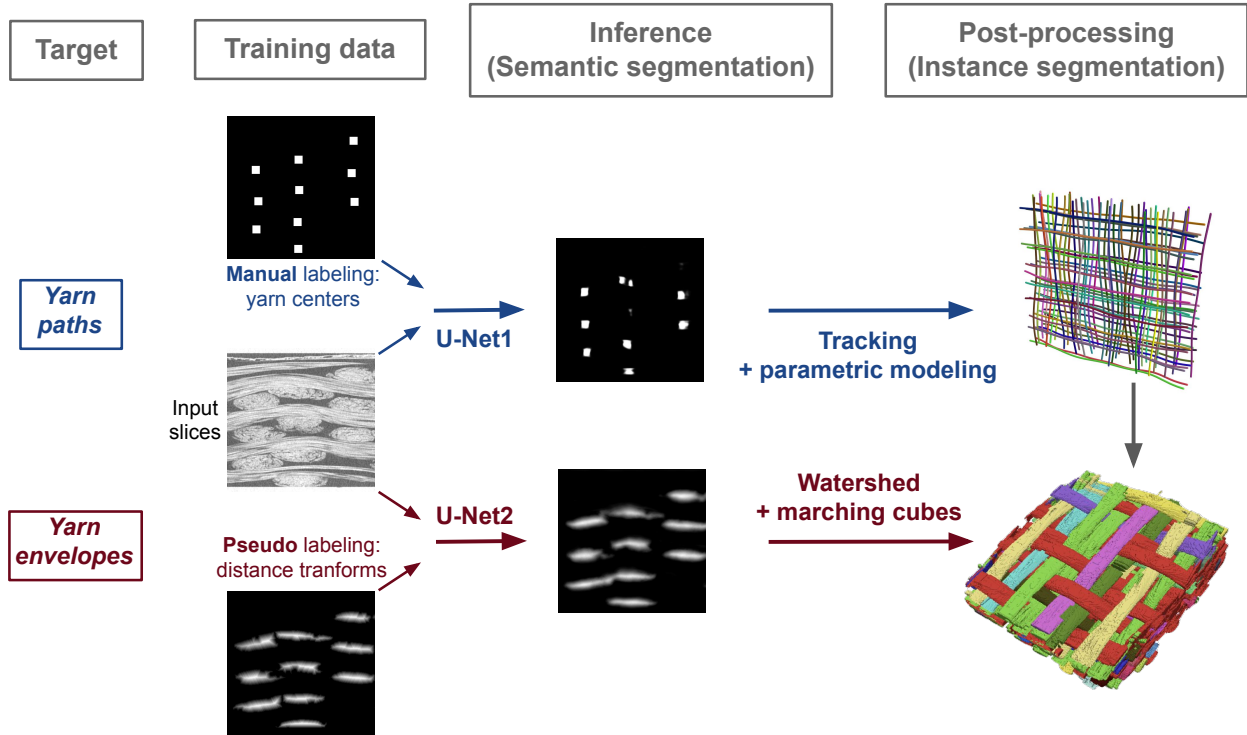


Figure 8: Summary of the method.

3. Results on the identification of yarn paths

3.1. Quality assessment

The assessment of our method is performed through four different metrics. First, *precision*, *recall* and F_1 *score* are used to compare the set of true yarn section centers to those predicted, before and after the tracking-modeling step. These metrics aim at measuring how noisy the detected point clouds are, and how much the noise is reduced after tracking and modeling. Second, in order to assess the quality of the 3D paths reconstructions, the Hausdorff distance is used, as suggested by [Mendoza et al. \[44\]](#)

The method concerning the detection of yarn section centers presented in [section 2.3.4](#) outputs a set of coordinates representing the points detected as part of the yarn paths in the tomography. This set will be called the *output point cloud*. The latter is compared to the ground truth section center marked as another

set of points, referred to as the *ground truth point cloud*. A ground truth point is considered as detected if, at least, one output point is close enough to it. Proximity is measured by the Euclidean distance in three dimensions. The threshold defining detection is 25 pixels. Finally, as the performance of a method is given by its ability to detect just the yarn paths and no other object, two quantities are considered, *precision* and *recall*. The recall is the proportion of ground truth points (*i.e.*, true positives + false negatives) that are correctly detected, while precision is the proportion of true positives among the output point cloud (*i.e.*, true positives + false positives). Both are numbers between zero and one, and the closer to one the better.

$$recall = \frac{\#true\ positives}{\#ground\ truth} \quad \text{and} \quad precision = \frac{\#true\ positives}{\#output} \quad (8)$$

From these two measures a unique score called F_1 score is computed as follows:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (9)$$

Again, the closer F_1 to one, the better.

Concerning the yarn paths assessment, the symmetric Hausdorff distance is used. Given two sets A and B , this distance is defined as

$$D_H(A, B) = \max(\delta_H(A, B), \delta_H(B, A)) \quad (10)$$

with

$$\delta_H(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (11)$$

and $\|\cdot\|$ the Euclidean norm. Every ground truth yarn path A_i is matched to the predicted path B_i that minimizes the Hausdorff distance:

$$B_i = \arg \min_{B \in \text{predicted paths}} D_H(A_i, B). \quad (12)$$

If this predicted path is already matched to another true path with lower Hausdorff distance, then the second closest is chosen, and so on. It may happen that no predicted path matches a ground truth one, in which case we consider that the path has not been reconstructed by our method.

3.2. Results on the non-compact dry preform

Table 1 sums up the results of the approach in terms of recall and precision. Figure 9 is a 3D visualization of the detected points and their filtered version after post-processing (tracking and parameterization). Although the raw results are already very satisfactory, the post-processing allows the removal of most false positives, as the improvement of precision shows. Additionally, it offers an identification of entire yarn paths along the volume, instead of independent points. The accuracy of the path reconstruction, measured by the Hausdorff distances between predicted and true paths, is reported in Figure 10b and Figure 10c. They may be compared to the average minimal distances between neighbour yarns in the slice plane, which is 1.81mm

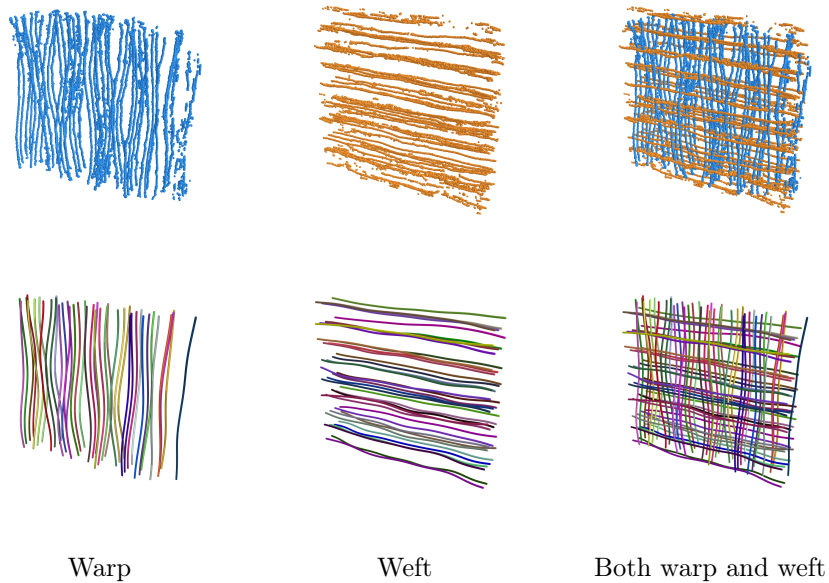


Figure 9: Points detected on the non-compacted dry preform (first row: semantic segmentation), and result after post-processing (second row: instance segmentation).

in warp and $1.83mm$ in weft. Note that in both warp and weft directions, some yarn paths were accurately reconstructed near the edges of the tomography although they were not labeled in the ground truth. They were discarded in the computation of precision, not to undermine it unfairly.

Table 1: Results of the yarn center segmentation on the non-compacted dry preform, before and after post-processing.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.980	.996	.979	.987
Precision	.959	.996	.962	.987
F_1 score	.969	.996	.971	.987

3.3. Results on the compacted dry preform

In a first attempt to apply our approach to compacted dry preform tomography, we used as training data images from the same tomography but orthogonal direction with respect to the test direction (that is, we sampled patches from the weft direction and tested on warp or the other way around). The results in

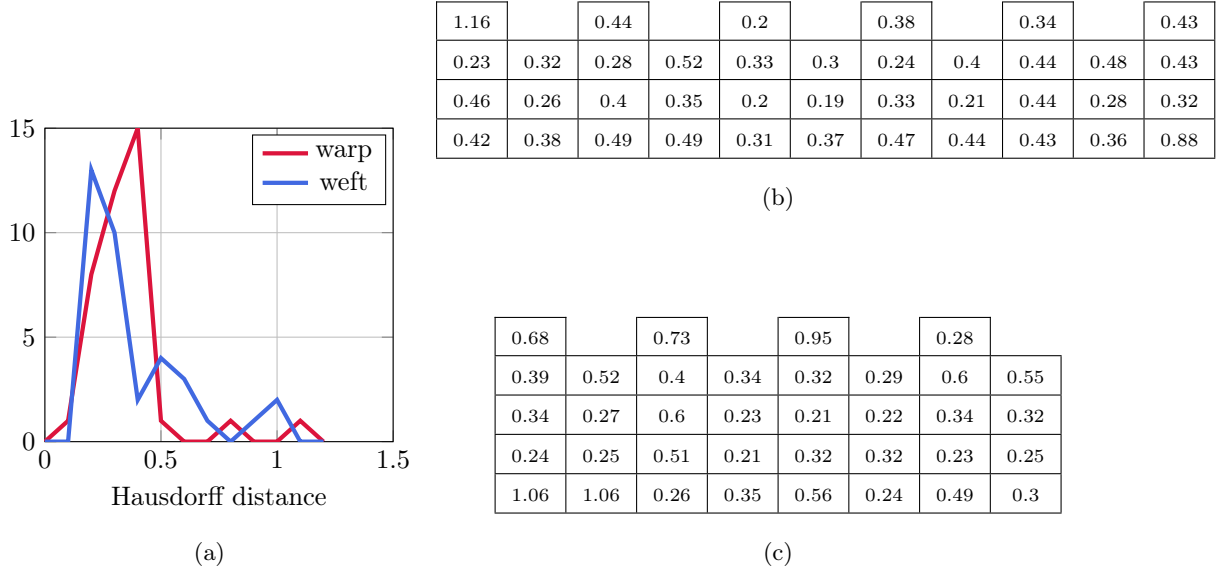


Figure 10: Yarn path assessment for the non-compacted dry preform: (a) Histogram of Hausdorff distances (in mm) for both warp (red line) and weft (blue line) orientation and listing of all distances for each (b) warp and (c) weft yarn.

terms of precision and recall are presented in Table 2. They are not good enough to recover all the yarns paths after post-processing. In particular, the drop in recall is explained by the fact that the post-processing removes correctly detected points if they are not sufficient to recover the whole yarn path.

Therefore, we tried to add images from the non-compacted preform tomography, but still the orthogonal direction, to the training data. This improved significantly the quality of the raw output point clouds, as shown in Table 3. From the visualization of Figure 11, top row, it appears that it is still quite noisier than for the initial non-compacted state. The post-processing step needed therefore a few more iterations and user interactions than with the non-compacted sample, but the process still converged quickly. The final results after pre-processing, shown in Table 3 in terms of precision and recall, and in Figure 12 in terms of Hausdorff distances, still need to be improved in the weft direction, where we still get a drop in recall. This is mainly due to the fact that one yarn could not be reconstructed. To help interpret the Hausdorff distance results compiled in Figure 12, we shall precise that the average minimal distance between two neighbour yarns in a slice plane, is about $1.10mm$ in warp and $1.24mm$ in weft.

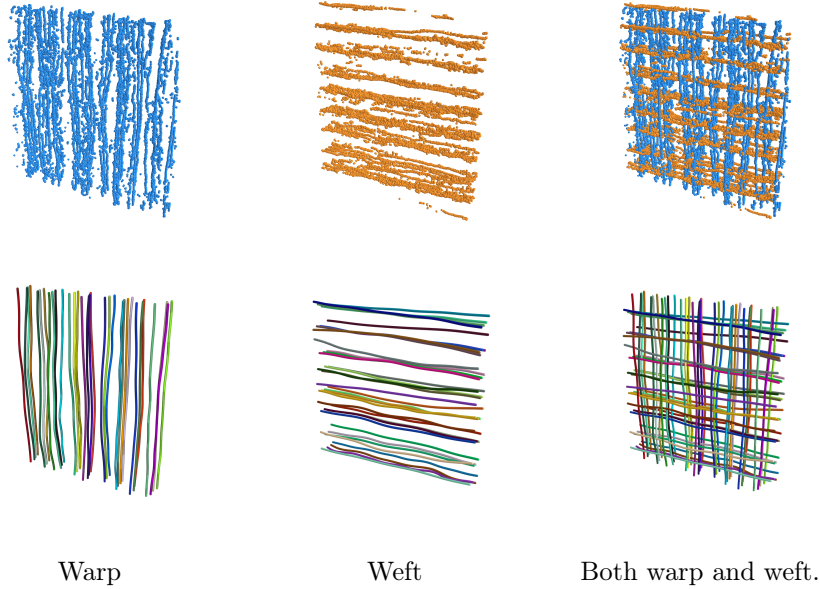


Figure 11: Points detected on the compacted dry preform by the Deep Learning method (first row: semantic segmentation), and result after post-processing (second row: instance segmentation).

Table 2: Results of the yarn center segmentation, before and after post-processing, on the compacted dry preform. Training on the **compacted** sample, orthogonal direction.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.949	.839	.854	.710
Precision	.851	.927	.782	.857
F_1 score	.897	.881	.817	.777

Table 3: Results of the yarn center segmentation, before and after post-processing, on the compacted dry preform. Training on **both compacted and non-compacted** samples, orthogonal direction.

	Warp		Weft	
	Raw	Post-proc.	Raw	Post-proc.
Recall	.980	.983	.895	.799
Precision	.901	.985	.824	.829
F_1 score	.939	.984	.858	.814

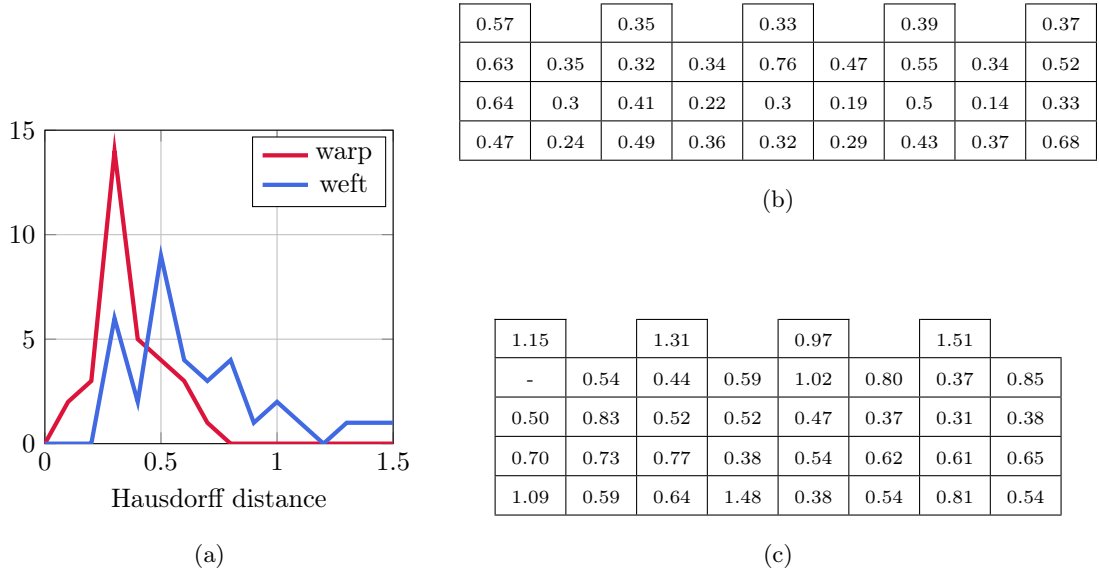


Figure 12: Yarn path assessment for the compacted dry preform: (a) Histogram of Hausdorff distances (in mm) for both warp (red line) and weft (blue line) orientation and listing of all distances for each (b) warp and (c) weft yarn. The hyphen indicates a missing yarn path in the reconstruction.

4. Results on the segmentation of yarn cross sections

In this section we present the results obtained by our approach in the instance segmentation of yarns for the non-compacted sample since the pseudo-labeling method was not satisfactory on the compacted sample. Regarding their quality assessment, it can only be made based on visual inspection, as we lack a reference ground truth for this task. All the segmentation results are made available online, as indicated hereafter. The whole results are presented in two videos available online¹. For a slice by slice examination, the corresponding sequences are also made available².

As described in section 2.3.5, the labels used to train a U-Net to segment yarns were the distance maps produced by a morphological algorithm. We observe that the model does learn to compute an output similar to a distance transform, as shown on the top row of Figure 13. Most importantly, the contours of yarns can be recovered accurately from the network’s output, following the method described in section 2.4.2. This advocates in favour of the representation of yarns by their distance transforms and training a deep CNN to learn this representation.

Compared to the morphological algorithm used to produce pseudo-labels for the supervised learning, it seems that the DNN shows better robustness to artifacts, as shown on the bottom row of Figure 13.

¹https://bit.ly/video_warp and https://bit.ly/video_weft

²https://bit.ly/imseq_compB

Regarding the surface mesh which can be produced from our 3D instance segmentation explained in [section 2.4.2](#), an example is shown in [Figure 14](#).

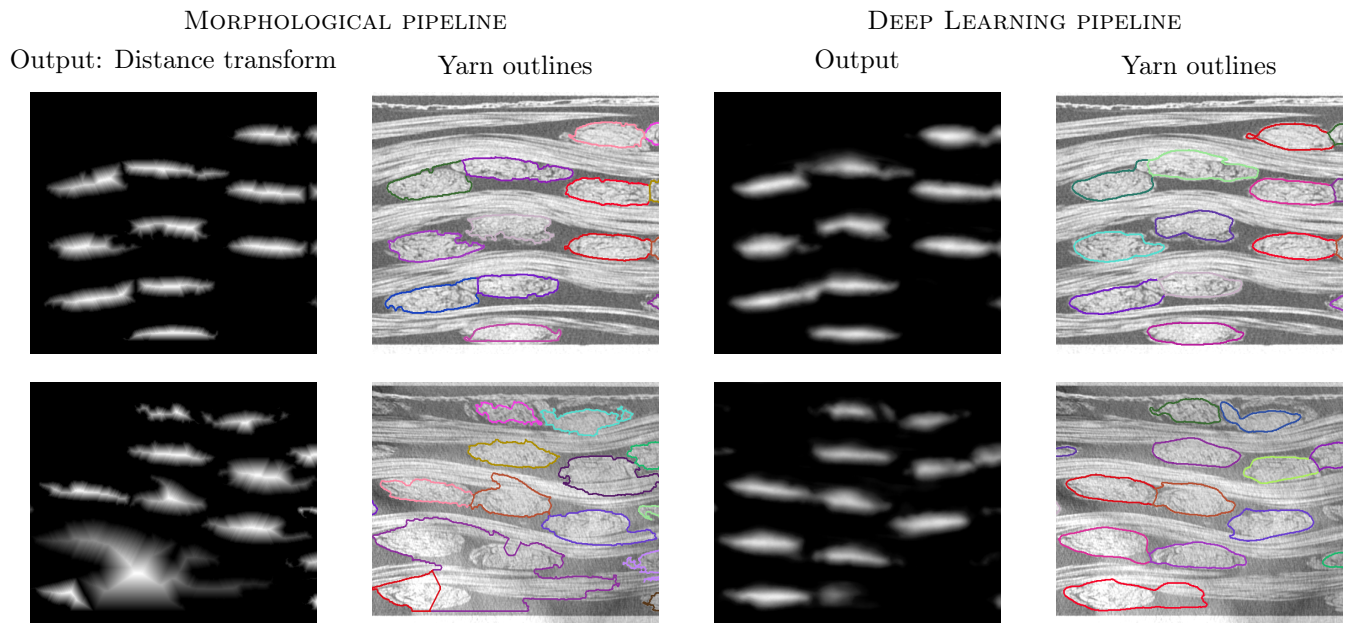


Figure 13: Comparison between the morphological pseudo-labeling and the deep-learning-based results on two warp slices from the non-compacted preform (training on weft slices).

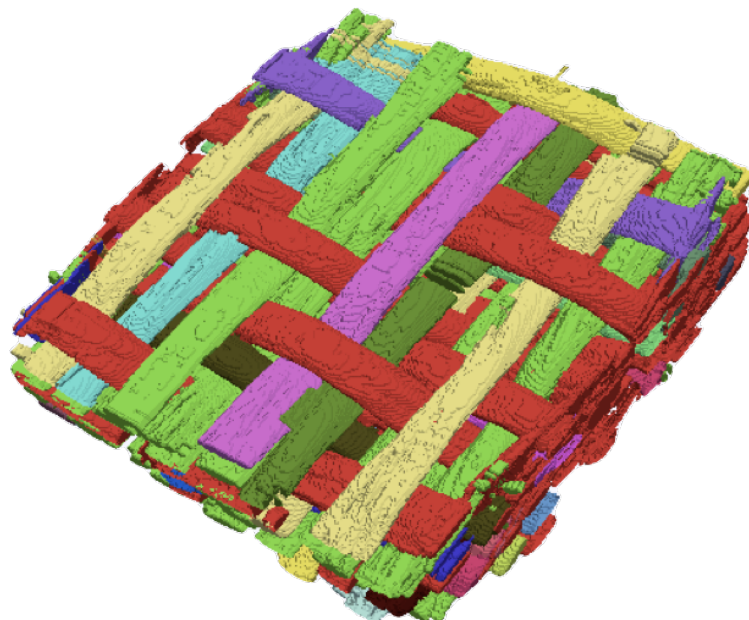


Figure 14: Instance segmentation of yarns, represented as a 3D surface mesh.

5. Discussion

The emergence of Deep Learning applied to segmentation domain offers a great perspective to solve complex image processing issues in composite field. Indeed, it provides a great tool allowing us to extract information of the internal structure of the composite materials with a strong efficiency and robustness never reached by classical morphological methods. The trained models DCNN using learnt features associated with several extra-parameters from the images (*e.g.*, resolution, grayscale distribution) as well as from the textile geometry (*e.g.*, yarn shape and aspect ratio, distance between yarns). However, the main difficulty in supervised approaches is to create annotated data, also called ground-truth. Generally, these annotations are generated manually, therefore starkly time-consuming and human-dependent. Our approach showed notably that it is possible to replace the manual ground-truth by a flawed automatised mathematical-morphology-based segmentation.

The proposed method separates the extraction of the yarn paths from the yarn envelopes but uses yarn trajectories to perform a complete yarn instance segmentation with their global shapes. This enables to select which level of details the user needs. Indeed, on the one hand, when information only on potential pattern anomalies (*e.g.*, wrong topology, strong transverse undulation, missing yarns) is required, only yarn path detections could be extracted. On the other hand, if more complete details are looked for, the yarn envelopes have to be reconstructed. Hence, for example, the envelopes computation can provide the intra-yarn fiber volume fraction and so, the overall fiber volume fraction of the composites. This complete description allows also to optimize the manufacturing of the composites materials by achieving forming process or testing simulations, in order to characterize the mechanical behaviour of the fibrous preform, as well as the injection process simulations (*e.g.*, determination of time filling, permeability). Finally, high fidelity FEA (*e.g.*, homogenization, crack propagation) on the consolidated material could thus be performed.

6. Conclusions and perspectives

In this paper we have presented an image processing framework aiming at extracting the yarn paths and their envelopes from μ -CT tomographies of a dry 3D ply-to-ply angle-interlock fabric, producing an instance segmentation of yarns.

Regarding the extraction of paths, our approach performed very well on the non-compacted sample, and also on the compacted one when training our DCNN on mixed data (slices from both samples) manually annotated. Regarding the segmentation of yarns, we used the output of a morphological algorithm to train a Deep Learning model. From this experiment we can draw an important conclusion: a deep architecture such as U-Net can learn to approximate the distance transform of the yarn sections, which indicates that the distance transform is a suitable labeling for such a supervised learning algorithm on this task.

In future works, we shall focus on the difficult issue of highly compacted fabrics, for which the segmentation of yarn sections is not satisfactory yet. To achieve a good quality pseudo-labeling for this sample, several strategies may be explored, such as pre-processing to regularize the data. More *a priori* information on the shape of yarns and the expected woven pattern, can also be exploited. Furthermore, the deep learning models we trained may be tuned to work on compacted samples thanks to data augmentation strategies.

7. References

- [1] P. Boisse, B. Zouari, A. Gasser, A mesoscopic approach for the simulation of woven fibre composite forming, *Composites Science and Technology* 65 (3-4) (2005) 429–436.
- [2] S. Mathieu, P. Boisse, N. Hamila, F. Bouillon, Locking and Stability of 3D Woven Composite Reinforcements, *Key Engineering Materials* 611-612 (2014) 292–299.
- [3] S. Mathieu, N. Hamila, F. Dupé, C. Descamps, P. Boisse, Stability of 3D Textile Composite Reinforcement Simulations: Solutions to Spurious Transverse Modes, *Applied Composite Materials* 23 (2016) 739–760.
- [4] A. Iwata, T. Inoue, N. Naouar, P. Boisse, S. V. Lomov, Coupled meso-macro simulation of woven fabric local deformation during draping, *Composites Part A* 118 (January) (2019) 267–280.
- [5] X. Sun, J. P.-H. Belnoue, A. Thompson, B. El Said, S. R. Hallett, Dry Textile Forming Simulations: A Benchmarking Exercise, *Frontiers in Materials* 9 (2022) 831820.
- [6] Y. Wielhorski, A. Mendoza, M. Rubino, S. Roux, Numerical modeling of 3D woven composite reinforcements: A review, *Composites Part A: Applied Science and Manufacturing* 154 (March) (2022) 106729.
- [7] G. Zhou, X. Sun, Y. Wang, Multi-chain digital element analysis in textile mechanics, *Composites Science and Technology* 64 (2) (2004) 239–244.
- [8] M. Sherburn, Geometric and Mechanical Modelling of Textiles, Ph.D. thesis, University of Nottingham, 2007.
- [9] Y. Miao, E. Zhou, Y. Wang, B. A. Cheeseman, Mechanics of textile composites: Micro-geometry, *Composites Science and Technology* 68 (7-8) (2008) 1671–1678.
- [10] Y. Mahadik, S. R. Hallett, Finite element modelling of tow geometry in 3D woven fabrics, *Composites Part A: Applied Science and Manufacturing* 41 (9) (2010) 1192–1200.
- [11] D. Durville, Simulation of the mechanical behaviour of woven fabrics at the scale of fibers, *International Journal of Material Forming* 3 (2) (2010) S1241–S1251.
- [12] S. Green, A. Long, B. El Said, S. Hallett, Numerical modelling of 3D woven preform deformations, *Composite Structures* 108 (2014) 747–756.
- [13] A. Drach, B. Drach, I. Tsukrov, Processing of fiber architecture data for finite element modeling of 3D woven composites, *Advances in Engineering Software* 72 (2014) 18–27.
- [14] L. Daelemans, J. Faes, S. Allaoui, G. Hivet, M. Dierick, L. Van Hoorebeke, W. Van Paepegem, Finite element simulation of the woven geometry and mechanical behaviour of a 3D woven dry fabric under tensile and shear loading using the digital element method, *Composites Science and Technology* 137 (2016) 177–187.
- [15] D. Durville, I. Baydoun, H. Moustacas, G. Périé, Y. Wielhorski, Determining the initial configuration and characterizing the mechanical properties of 3d angle-interlock fabrics using finite element simulation, *International Journal of Solids and Structures* 154 (December) (2018) 97–103.
- [16] S. Lomov, A. Gusakov, G. Huysmans, A. Prodromou, I. Verpoest, Textile geometry preprocessor for meso-mechanical models of woven composites, *Composites Science and Technology* 60 (11) (2000) 2083–2095.
- [17] I. Verpoest, S. Lomov, Virtual textile composites software WiseTex: integration with micro-mechanical, permeability and structural analysis, *Composites Science and Technology* 65 (15-16) (2005) 2563–2574.

- [18] Y. Wang, X. Sun, Digital-element simulation of textile processes, *Composites Science and Technology* 61 (2) (2001) 311–319.
- [19] A. Charmetant, E. Vidal-Sallé, P. Boisse, Hyperelastic modelling for mesoscopic analyses of composite reinforcements, *Composites Science and Technology* 71 (2011) 1623–1631.
- [20] K. Naresh, K. A. Khan, R. Umer, W. J. Cantwell, The use of X-ray computed tomography for design and process modeling of aerospace composites: A review, *Materials and Design* 190 (2020) 108553.
- [21] A. Mendoza, J. Schneider, E. Parra, S. Roux, Measuring yarn deformations induced by the manufacturing process of woven composites, *Composites Part-A: Applied Sciences and Manufacturing* 120 (2019) 127–139.
- [22] A. Mendoza, J. Schneider, E. Parra, E. Obert, S. Roux, Differentiating 3D textile composites: A novel field of application for Digital Volume Correlation, *Composite Structures* 208 (2019) 735–743.
- [23] R. M. Auenhammer, L. P. Mikkelsen, L. E. Asp, B. J. Blinzler, Automated X-ray computer tomography segmentation method for finite element analysis of non-crimp fabric reinforced composites, *Composite Structures* 256 (2021) 113136.
- [24] M. Ali, R. Umer, K. Khan, S. Bickerton, W. J. Cantwell, Non-destructive evaluation of through-thickness permeability in 3D woven fabrics for composite fan blade applications, *Aerospace Science and Technology* 82 (2018) 520–533.
- [25] W. Wijaya, M. A. Ali, R. Umer, K. A. Khan, P. A. Kelly, S. Bickerton, An automatic methodology to CT-scans of 2D woven textile fabrics to structured finite element and voxel meshes., *Composites Part-A: Applied Sciences and Manufacturing* 125 (2019) 105561.
- [26] Y. Liu, I. Straumit, D. Vasiukov, S. V. Lomov, S. Panier, Prediction of linear and non-linear behavior of 3D woven composite using mesoscopic voxel models reconstructed from X-ray micro-tomography, *Composite Structures* 179 (2017) 568–579.
- [27] J. Bénézech, G. Couégnat, Variational segmentation of textile composite preforms from X-ray computed tomography, *Composite Structures* 230 (2019) 111496.
- [28] N. Naouar, E. Vidal-Sallé, J. Schneider, E. Maire, P. Boisse, Meso-scale FE analyses of textile composite reinforcement deformation based on X-ray computed tomography, *Composite Structures* 116 (2014) 165–176.
- [29] N. Naouar, E. Vidal-Sallé, J. Schneider, E. Maire, P. Boisse, 3D composite reinforcement meso FE analyses based on X-ray computed tomography, *Composite Structures* 132 (2015) 1094–1104.
- [30] M. Ali, R. Umer, K. Khan, W. J. Cantwell, In-plane virtual permeability characterization of 3D woven fabrics using a hybrid experimental and numerical approach, *Composites Science and Technology* 173 (2019) 99–109.
- [31] L. Djukic, I. Herszberg, W. Walsh, G. Schoeppner, B. Gangadhara Prusty, D. Kelly, Contrast enhancement in visualisation of woven composite tow architecture using a MicroCT scanner. Part 1: Fabric coating and resin additives, *Composites Part-A: Applied Sciences and Manufacturing* 40 (2009) 553–565.
- [32] I. Straumit, S. V. Lomov, M. Wevers, Quantification of the internal structure and automatic generation of voxel models of textile composites from X-ray computed tomography data, *Composite Part-A: Applied Sciences and Manufacturing* 69 (2015) 150–158.
- [33] N. Naouar, D. Vasiukov, C.-H. Park, S. V. Lomov, P. Boisse, Meso-FE modelling of textile composites and X-ray tomography, *Journal of Materials Science* 55 (36) (2020) 16969–16989.
- [34] B. Wintiba, D. Vasiukov, S. Panier, S. V. Lomov, K. Ehab, M. Kamel, T. J. Massart, Automated reconstruction and conformal discretization of 3D woven composite CT scans with local fiber volume fraction control, *Composite Structures* 248 (May) (2020) 112438.
- [35] D. W. Aha, D. Kibler, M. K. Albert, Instance-Based Learning Algorithms, *Machine Learning* 6 (1991) 37–66.
- [36] V. Pidou-Brion, Y. Le Guilloux, Active yarn meshes for segmentation on X-ray computed tomography of textile composite materials at the mesoscopic scale, *Composite Structures* .
- [37] Y. Sinchuk, P. Kibleur, J. Aelterman, M. N. Boone, W. V. Paepegem, Variational and Deep Learning Segmentation of

- Very-Low-Contrast X-ray Computed Tomography Images of Carbon/Epoxy Woven Composites, *Materials* 13 (4).
- [38] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241, 2015.
- [39] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-cnn, in: *ICCV*, 2961–2969, 2017.
- [40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: *European conference on computer vision*, Springer, 740–755, 2014.
- [41] M. A. Ali, Q. Guan, R. Umer, W. J. Cantwell, T. Zhang, Deep learning based semantic segmentation of μ CT images for creating digital material twins of fibrous reinforcements, *Composites Part A: Applied Science and Manufacturing* 139 (2020) 106131.
- [42] M. Ali, Q. Guan, R. Umer, W. J. Cantwell, T. Zhang, Efficient processing of μ CT images using deep learning tools for generating digital material twins of woven fabrics, *Composites Science and Technology* 217 (2022) 109091.
- [43] A. Badran, D. Marshall, Z. Legault, R. Makovetsky, B. Provencher, N. Piché, M. Marsh, Automated segmentation of computed tomography images of fiber-reinforced composites by deep learning, *Journal of Materials Science* 55 (34) (2020) 16273–16289.
- [44] A. Mendoza, R. Trullo, Y. Wielhorski, Descriptive Modeling of Textiles using FE Simulations and Deep Learning, *Composites Science and Technology* (2021) 108897.
- [45] Y. Sinchuk, P. Kibleur, J. Aelterman, M. N. Boone, W. V. Paepegem, Geometrical and deep learning approaches for instance segmentation of CFRP fiber bundles in textile composites, *Materials* 227 (2020) 114626.
- [46] W. Huang, P. Causse, V. Brailovski, H. Hu, F. Trochu, Reconstruction of mesostructural material twin models of engineering textiles based on Micro-CT Aided Geometric Modeling, *Composites Part A: Applied Science and Manufacturing* 124 (2019) 105481.
- [47] S. Beucher, F. Meyer, *The morphological approach to segmentation: the watershed transformation*, chap. 12, Marcel Dekker, Inc., 433–481, 1993.
- [48] F. Meyer, Topographic distance and watershed lines, *Signal Process* 38 (1) (1994) 113–125.
- [49] X. Zeng, B. L. P., A. Endrueit, M. Mikhail, L. A. C., Geometrical modelling of 3D woven reinforcements for polymer composites: Prediction of fabric permeability and composite mechanical properties, *Composite Part-A: Applied Sciences and Manufacturing* 56 (2014) 150–160.
- [50] L. Daelemans, B. Tomme, B. Caglar, V. Michaud, J. Van Stappen, V. Cnudde, M. Boone, W. Van Paepegem, Kinematic and mechanical response of dry woven fabrics in through-thickness compression : Virtual fiber modeling with mesh overlay technique and experimental validation, *Composites Science and Technology* 207 (January) (2021) 108706.
- [51] A. Geoffre, Y. Wielhorski, N. Moulin, J. Bruchon, S. Drapier, P.-j. Liotier, *International Journal of Multiphase Flow* Influence of intra-yarn flows on whole 3D woven fabric numerical permeability: from Stokes to Stokes-Darcy simulations, *International Journal of Multiphase Flow* 129.
- [52] S. Blusseau, S. Velasco-Forero, J. Angulo, I. Bloch, Tropical and morphological operators for signal processing on graphs, in: *25th IEEE International Conference on Image Processing (ICIP)*, 1198–1202, 2018.
- [53] S. Blusseau, S. Velasco-Forero, J. Angulo, I. Bloch, Adaptive Anisotropic Morphological Filtering Based on Co-Circularity of Local Orientations, *Image Processing On Line* 12 (2022) 111–141, <https://doi.org/10.5201/ipol.2022.397>.
- [54] N. Otsu, A threshold selection method from gray-level histograms, *IEEE transactions on systems, man, and cybernetics* 9 (1) (1979) 62–66.
- [55] P. Soille, et al., *Morphological image analysis: principles and applications*, vol. 2, Springer, 1999.
- [56] Y. LeCun, Y. Bengio, et al., Convolutional networks for images, speech, and time series, *The handbook of brain theory and neural networks* 3361 (10).
- [57] L. Deng, D. Yu, *Deep learning: methods and applications*, *Foundations and trends in signal processing* 7 (3–4) (2014)

197–387.

- [58] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, *APSIPA Transactions on Signal and Information Processing* 3.
- [59] V. Vapnik, Principles of risk minimization for learning theory, in: *Advances in neural information processing systems*, 831–838, 1992.
- [60] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [61] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, Tech. Rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [62] L. Prechelt, Early stopping-but when?, in: *Neural Networks: Tricks of the trade*, Springer, 55–69, 1998.
- [63] K. Diederik P., B. Jimmy, Adam: A Method for Stochastic Optimization, in: Y. Bengio, Y. LeCun (Eds.), *ICLR*, 2015.
- [64] M. A. Rahman, Y. Wang, Optimizing intersection-over-union in deep neural networks for image segmentation, in: *International symposium on visual computing*, Springer, 234–244, 2016.
- [65] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
- [66] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *ACM siggraph computer graphics* 21 (4) (1987) 163–169.
- [67] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in Python, *PeerJ* 2 (2014) e453.