Juliane Blarr

DEVELOPMENT OF COMPUTATIONAL, IMAGE PROCESSING AND DEEP LEARNING METHODS FOR THE MICROSTRUCTURE CHARACTERIZATION OF CARBON FIBER REINFORCED POLYAMIDE 6 BASED ON CT IMAGES

SCHRIFTENREIHE DES INSTITUTS
FÜR ANGEWANDTE MATERIALIEN BAND 124

KIT Scientific Publishing

Juliane Blarr

Development of Computational, Image Processing and Deep Learning Methods for the Microstructure Characterization of Carbon Fiber Reinforced Polyamide 6 Based on CT Images

Schriftenreihe
des Instituts für Angewandte Materialien
Band 124

Karlsruher Institut für Technologie (KIT)
Institut für Angewandte Materialien (IAM)

Eine Übersicht aller bisher in dieser Schriftenreihe erschienenen Bände finden Sie am Ende des Buches.

Development of Computational, Image Processing and Deep Learning Methods for the Microstructure Characterization of Carbon Fiber Reinforced Polyamide 6 Based on CT Images

by
Juliane Blarr

KIT Scientific Publishing

Karlsruher Institut für Technologie
Institut für Angewandte Materialien

Development of Computational, Image Processing and Deep Learning Methods for the Microstructure Characterization of Carbon Fiber Reinforced Polyamide 6 Based on CT Images

Zur Erlangung des akademischen Grades einer Doktorin der Ingenieurwissenschaften von der KIT-Fakultät für Maschinenbau des Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Juliane Blarr, M.Sc.

Tag der mündlichen Prüfung: 30. August 2024
Hauptreferent: Prof. Dr.-Ing. Kay André Weidenmann
Korreferentin: Prof. Dr.-Ing. Luise Kärger
Korreferent: Prof. Dr. Kaan Inal

Impressum

KIT Scientific Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

www.bibliothek.kit.edu/ksp.php | E-Mail: info@ksp.kit.edu | Shop: www.ksp.kit.edu

Print on Demand 2025 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 2192-9963
ISBN 978-3-7315-1396-4
DOI 10.5445/KSP/1000176200

“Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.”

Marie Curie

Abstract

Discontinuous fiber reinforced polymers such as long fiber reinforced thermoplastics are valued for their good density-specific mechanical properties, relatively low costs, fast cycle times and possible function integration in lightweight design relevant sectors such as automotive engineering. However, their complex microstructure exacerbates the modeling and design of components. To characterize the micro- and mesostructure, the acquisition of three-dimensional computed tomography scans and subsequent image analysis has established itself as a non-destructive method. Especially carbon fiber reinforced polymers pose challenges for this approach: Their small diameter combined with a low image contrast to the surrounding matrix (due to similar attenuation coefficients) complicates the application of previously developed methods for the evaluation of mechanical parameters such as fiber volume contents. Therefore, two novel methods for determining the fiber volume content from scans of this material are presented in this thesis, one of which is an artificial intelligence-based convolutional neural network approach. Furthermore, even in the case of quantities that can be determined using already implemented methods, such as fiber orientation tensors, computed tomography images pose the problem that a high resolution is associated with a small specimen geometry. In turn, this reduces the cross-scale significance of a quantity determined from that small specimen. Three different methods (two algebraic and one artificial neural network) for the interpolation of fiber orientation tensors were therefore tested in order to obtain orientation information across an entire plate from a few support points of tensors that were determined from small specimens. The orientation distribution of the plastificate and thus of the initial state, on which the compression molding process is based, was also investigated. In addition, the problem arose that the acquisition of CT

scans is energy-intensive and time-consuming. Artificial microstructures could reduce this scanning effort. However, artificial microstructures of carbon long fiber reinforced material generated with conventional packing algorithms do not sufficiently resemble the real ones. As a result, the rather novel approach of generative neural networks was successfully used to generate two-dimensional images that are as similar as possible to the layers of the computer tomography scans. The investigations in this dissertation show the potential of image evaluation and especially the use of approaches based on artificial intelligence for the investigation of computed tomography images of carbon long fiber reinforced polymers despite the physically unfavorable initial situation.

Kurzfassung

Diskontinuierlich faserverstärkte Kunststoffe wie langfaserverstärkte Thermoplaste werden in leichtbaurelevanten Branchen, wie dem Automobilbau, aufgrund ihrer guten dichtespezifischen mechanischen Eigenschaften, verhältnismäßig geringen Kosten, schnellen Zykluszeiten und möglicher Funktionsintegration geschätzt. Deren komplexe Mikrostruktur ist allerdings der Modellierung und Auslegung von Bauteilen hinderlich. Zur Charakterisierung der Mikro- und Mesostruktur hat sich das Anfertigen von dreidimensionalen Computertomographiescans und anschließende Bildauswertung als zerstörungsfreies Prüfverfahren etabliert. Die Verwendung von Kohlenstofffasern sorgt allerdings für Herausforderungen bei dieser Herangehensweise: Ihr geringer Durchmesser verbunden mit niedrigem Kontrast im Bild zur umgebenden Matrix erschweren durch hohes Bildrauschen die Anwendung bereits entwickelter Verfahren zur Auswertung mechanischer Kenngrößen, wie Faservolumenanteilen. Aus diesem Grund werden in dieser Thesis zwei neuartige Methoden zur Bestimmung des Faservolumengehalts aus Scans dieses Materials aufgezeigt, wovon eines ein auf künstlicher Intelligenz basierter Ansatz eines convolutional neural networks ist. Weiterhin ergibt sich selbst bei mit bereits implementierten Verfahren bestimmbaren Größen, wie Faserorientierungstensoren, bei Computertomographieaufnahmen die Problematik, dass eine hohe Auflösung mit einer kleinen Probengeometrie einhergeht. Kleine Probengrößen reduzieren wiederum die skalenübergreifende Aussagekraft einer daraus bestimmten Größe. Um aus wenigen Stützpunkten an Tensoren, die aus kleinen Proben bestimmt wurden, Orientierungsinformationen über eine ganze Platte hinweg zu erhalten, werden daher drei verschiedene Verfahren (zwei algebraische und erneut ein

künstliches neuronales Netzwerk) zur Interpolation von Faserorientierungstensoren implementiert und evaluiert. Auch die Orientierungsverteilung des Plastifikats und damit des dem Pressprozesses zugrundliegenden Ausgangszustands wurde untersucht. Darüberhinaus ergab sich die Problematik, dass die Anfertigung von CT-Scans energieintensiv und zeitaufwändig ist. Künstliche Mikrostrukturen könnten diesen Scanaufwand verringern. Allerdings ähneln die mit konventionellen Packungsalgorithmen erzeugten künstlichen Mikrostrukuren von kohlenstofflangfaserverstärktem Material den realen nicht ausreichend. Infolgedessen wurde der recht neuartige Ansatz von generativen neuronalen Netzen erfolgreich verwendet, um - trainiert auf den Schichten der Computertomographiescans - möglichst ähnliche zweidimensionale Bilder zu erzeugen. Durch die Untersuchungen in dieser Dissertation zeigt sich das Potential der Bildauswertung und speziell auch der Nutzung von Ansätzen basierend auf künstlicher Intelligenz zur Untersuchung von Computertomographieaufnahmen kohlenstofflangfaserverstärkter Polymere trotz der physikalisch ungünstigen Ausgangssituation.

Acknowledgements

First of all, many thanks to Prof. Kay André Weidenmann, who made it possible for me to write this dissertation in the first place by granting me the position and believing in me since I wrote my Bachelor's thesis. I would also like to thank him for his constant support and kind character, incredibly quick responses to emails and inquiries of all kinds, and academic discussions. Thanks to Prof. Peter Elsner, who sadly passed away during my doctoral time. He was my main supervisor and I was able to benefit greatly from his pragmatic and academic advice combined with his calm nature at the beginning. I would also like to take this opportunity to thank Prof. Luise Kärger, who subsequently kindly took me on as a KIT-supervising professor. I am very grateful for her outstanding scientific comments on my work and her modest and humble character. Thanks also to Prof. Kaan Inal, my Canadian supervisor, who gave me great support and made two stays in Canada possible. His mixture of encouragement and trust paired with challenging demands has always helped me to improve. Thanks also to Prof. Thomas Böhlke, who always supported me in the IRTG and advised me in one-on-one meetings when necessary.

Furthermore, I would like to thank Dr.-Ing. Wilfried Liebig, my group leader and direct boss, who was always there for me, for his academic support, humanity, and help in day-to-day business. Thanks to all my colleagues in the IRTG and at the IAM-WK. I would like to take this opportunity to thank my office colleague Benedikt Scheuring in particular for his friendship and help. We went through a lot of fun and exhausting times together, which I would not want to miss and will remember forever. Thanks also to Nicolas Christ, who has been a friend for longer than my PhD. I was able to spend exciting, challenging and

wonderful times with both of them in Canada. I would also like to thank all my other colleagues at IAM-WK, especially from the HWL. Thanks to Anselm for countless conversations - not only about materials science - and to Daniel for making me laugh as well as to Ludwig and Miriam for the great guidance in the beginning and to Markus, Henrik and Christoph. You all made the good times better and the stressful ones bearable. I'm also very grateful for the support of the workshop and the technicians, of whose years of experience we all profit. A special thanks to my students, Steffen Klinder, Philipp Kunze, Noah Kresin and Francisco Xabier Landa, that helped me by either writing their thesis with me and/or being student assistants.

In particular, I would like to acknowledge and thank my boys, with whom I fought my way through my mechanical engineering studies and without whom I would probably never have gotten this far and would not be the person I am today. Many thanks also to my (soccer) girls and my flatmates, who have accompanied and supported me during my studies and doctorate and for whose friendship I am very grateful. Massive thanks especially to my love for your unwavering care and support and fulfilling my life with joy and purpose. You make me a better person every day.

Finally, I would like to thank my family. My sister, who taught me to read even before I started school, my brother, who helped me all too often with my maths and physics homework, and my parents, who have always supported me emotionally and financially. You have played no small part in this dissertation and I love you.

Karlsruhe, June 2024 *Juliane Blarr*

Symbols and acronyms

Acronyms

(µ)CT	(Micro-) Computed Tomography
.ipynb	Jupyter notebook
.py	Native Python file
2D	Two-Dimensional
3D	Three-Dimensional
ABS	Acrylonitrile Butadiene Styrene
ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
AOA	Average Or Above (thresholding)
BCE	Binary Cross Entropy
bwHPC	Baden-Württemberg High Performance Computing
C_i	Label of specific specimen in charge area of plate
CA	Component Averaging
CcGAN	Continuous conditional GAN
CF	Carbon Fiber
CF_i	Label of specific specimen in transition area between charge and flow of plate

CFRP	Carbon Fiber Reinforced Polymer
cGAN	Conditional GAN
CNN	Convolutional Neural Network
Co	Continuous (reinforced polymers)
CoDicoFRP	Continuous-Discontinuous Fiber Reinforced Polymer
CoFRP	Continuous Fiber Reinforced Polymer
CT	Computed Tomography
DCGAN	Deep Convolutional Generative Adversarial Network
DFG	Deutsche Forschungsgemeinschaft
Dico	Discontinuous (reinforced polymers)
DicoFRP	Discontinuous Fiber Reinforced Polymer
DL	Deep Learning
ED	Euclidean Distance
F_i	Label of specific specimen in flow area of plate
FASEP	Brand name - System for determining the fiber length distribution
FE	Finite Element (model)
FID	Fréchet Inception Distance
FIJI	Image processing package (a distribution of ImageJ2, bundling a lot of plugins which facilitate scientific image analysis)
FLD	Fiber Length Distribution
FOD	Fiber Orientation Distribution
FODF	Fiber Orientation Distribution Function
FOT	Fiber Orientation Tensor
FRP	Fiber Reinforced Polymer
fSAM	Fused Sequential Addition and Migration
FVC	Fiber Volume Content

FVF	Fiber Volume Fraction
GAN	Generative Adversarial Network
GF	Glass Fiber
GPa	Gigapascal
GPU	Graphics Processing Unit
HSV	Hue, Saturation and Value
HT	High-tenacity
IBOF	Invariant Based Optimal Fitting
IDW	(Shepard's) Inverse Distance Weighting
InfoGAN	Information maximizing GAN
IRTG	International Research Training Group (in this work: IRTG2078 funded by the German Research Foundation)
LFT	Long Fiber reinforced Thermoplastic
LFT-D	Long Fiber reinforced Thermoplastic Direct process
LFT-G	Long Fiber reinforced Thermoplastic Granulate process
MAE	Mean Absolute Error
MEP	Maximum Entropy Method
ML	Machine Learning
MLP	Multi-Layer Perceptron
MPa	Megapascal
MSE	Mean Squared Error
MSSIM	Mean Structural Similarity Index Measure (SSIM)
ODF	Orientation Distribution Function
PA(6)	Polyamide (6)
PAN	Polyacrylonitrile
PC	Polycarbonate
PET	Polyethylene Terephthalate

Pixel (px)	Picture x Element
PP	Polypropylene
RAM	Random-Access Memory
ReLU	Rectified Linear Unit
ROI	Region(s) Of Interest
ROM	Rule Of Mixture
RSA	Random Sequential Adsorption
RVE	Representative Volume Element
SAM	Sequential Addition and Migration
SEM	Scanning Electron Microscopy
SFT	Short Fiber reinforced Thermoplastic
SGD	Stochastic Gradient Descent
SLP	Single Layer Perceptron
SMC	Sheet Molding Compound
SNR	Signal-to-Noise Ratio
SPD	Symmetric Positive Definite
SR	Super Resolution
SSE	Sum of Squares Error
SSH	Secure Shell protocol
SSIM	Structural Similarity Index Measure
SVE	Statistical/Stochastic Volume Element
TEM	Transmission Electron Microscopy
TGA	Thermogravimetric analysis
Voxel (vx)	Volume x Element
WGAN	Wasserstein GAN
WGAN-GP	Wasserstein GAN with Gradient Penalty

Scalars

α	Absorption coefficient
$\bar{\mu}_{\mathrm{m}}$	Effective mass attenuation coefficient
$\bar{\phi}$	Interpolation scheme
$\bar{E}$	Effective Young's modulus of composite
η	Distance variable
γ	(Shear) Strain
λ_i	Eigenvalues
$\mathbb{E}$	Expectation operator
μ	Attenuation coefficient
$\mu_0, \mu_1, \mu_{\mathrm{T}}$	Class mean levels
μ_{m}	Mass attenuation coefficient
$\mu_{\mathrm{r}}, \mu_{\mathrm{g}}$	Means of real and generated images
μ_{s}	Scatter coefficient
ν	Poisson ratio
ω_0, ω_1	Probabilities of class occurrence
ϕ_i	Discrete values
ψ	Probability distribution function
ρ	Density
ρ_i	Standard deviation (in context of structure tensor)
σ	Width of filter window/standard deviation of Gaussian kernel
$\sigma(u)$	Activation function
σ_0, σ_1	Class variances
σ^2_{B}	Between-class variance
σ^2_{W}	Within-class variance
σ_i	Normal stress in i-direction

τ	Shear stress
θ_{G}	Parameters of the generator
ε	(Normal) Strain
ξ	Empirical geometry factor
A	Area
a_{r}	Aspect ratio
b	Bias
C	Stiffness
c	cos of angle
C_{fiber}	Circumference of fiber
$cov(x, y)$	Covariance of variables x and y
cov_{r}, cov_g	Covariance of real and generated images
D	Discriminator
d	Diameter
$d(\boldsymbol{x}, \boldsymbol{x}_i)$	Distance from known point $\boldsymbol{x}_i$ to unknown point $\boldsymbol{x}$
d_{E}	Euclidean distance
D_x	Definition domain of structuring element
E	Young's modulus
e	Charge of an electron
E_p	Distance function or error/cost function
$F \times F$	Filter dimension
F	Force
f	Mapping
$f(x, y)$	Filter mask
f_{l}	Fiber length probability density function
G	Generator
$G(x, y)$	Gaussian function

$G_\mathrm{f}, G_\mathrm{m}, G_{ij}$	Shear modulus (of the fibers, the matrix or considering the two coordinate axes i and j)
$G_{\boldsymbol{A}}$	Glyph to tensor $\boldsymbol{A}$
$h^{(l)}$	Hidden layer
I	Radiation intensity
i	Input
$I(\boldsymbol{x})$	Image intensity function
I_0	Initial condition of radiation intensity
K_i	Orthogonal K-invariants
l	(Fiber) Length
$l(y, \hat{y})$	Log loss/BCE loss
l_c	Critical fiber length
l_d	Detector pixel size
L_FD	Distance between focus and detector
L_FO	Distance between focus and object
l_t	Length of fiber in order to transfer load
l_v	Voxel size
L_i	Specific fiber length of the i-th fiber
m	Mass
m_e	Mass of an electron
m_g	Geometric magnification
m_i	Moments of i
$N \times N$	Image dimension
n/N	Number (amount of instances of certain parameter/variable)
O	Output size
o	Soft sign activation function
$o(\boldsymbol{x})$	Output image

P	Padding
p_{g}	Distribution of the generator
p_{z}	Distribution of noise variables
p_i	Probabilities
p_j	Fractions of below-threshold and above-threshold pixels
R	Mean separation of the fibers normal to their length
r/r_{f}	Radius / Fiber radius
r_{l}	Learning rate
R_i	Orthogonal R-invariants
S	Stride
s	$\sin$ of angle
s	Isotropic voxel size
t	Threshold
T_{Otsu}	Otsu threshold
u	Activation potential
U_{a}	Acceleration voltage
V	Volume
v	Velocity
$V(D, G)$	Value function
v_{f}	Fiber volume content
w_{f}	Fiber weight content
w_i	Weights (fraction)
x_n	Input values
y_j	Output values
z_i	Class label
z_j	Gray value

Vectors

$\boldsymbol{h}$	Hidden layer vector (output)
$\boldsymbol{p}$	Unit vector describing fiber orientation
$\boldsymbol{v}_i$	Eigenvector
$\boldsymbol{x}$	Location

Tensors (2nd order)

$\bar{\boldsymbol{\sigma}}$	Effective stress
$\bar{\boldsymbol{\varepsilon}}$	Effective strain
$\boldsymbol{\Lambda}$	Diagonal eigenvalue matrix
$\boldsymbol{\sigma}$	Stress
$\boldsymbol{\tau}$	Shear stress
$\boldsymbol{\varepsilon}$	Strain
$\boldsymbol{A}$	(Fiber) orientation tensor of second order
$\boldsymbol{C}$	Second order stiffness tensor
$\boldsymbol{H}$	Hessian matrix
$\boldsymbol{J}$	Final structure tensor
$\boldsymbol{K}_\sigma$	Gaussian Blur of width σ
$\boldsymbol{R}$	Rotation matrix (composed of the eigenvectors $\boldsymbol{v}_i$)
$\boldsymbol{S}$	Structure tensor
$\boldsymbol{W}$	Weight matrix
UL, ..., LR	Measured tensors of second order (respectively denoting “Upper Left", ..., "Lower Right", etc.). All together form the set of measured tensors $\mathcal{T}_\mathrm{m}$

Tensors (4th order)

$\mathbb{A}$	(Fiber) orientation tensor of fourth order
$\mathbb{C}$	Fourth order stiffness tensor
$\mathbb{E}$	Eshelby tensor
$\mathbb{H}^0$	Reference tensor
$\mathbb{I}$	Identity tensor
$\mathbb{I}^S$	Identity on symmetric tensors
$\mathbb{L}$	Localization tensor
$\mathbb{M}$	Localization tensor
$\mathbb{S}$	Compliance tensor (inverse stiffness tensor)

Sets

$\mathbb{R}^3$	Real coordinate space of dimension 3
$\mathcal{C}$	Closure approximation
$\mathcal{P}$	Probability of finding fiber in direction $\boldsymbol{p}$
$\mathcal{S}$	Two-dimensional unit sphere
$\mathcal{T}_\mathrm{m}$	Set of measured tensors of second order
$\mathcal{T}_i = \boldsymbol{T}_{x,y}$	Set of interpolated tensors of second order at specific (x,y) grid position
$SO(3)$	Special orthogonal group in 3 dimensions

Operators

$(\cdot)'$	Deviatoric part
$(\cdot)^T$	Transpose
$(\cdot)^{-1}$	Inverse
:	Double contraction of tensors
$*$	Convolution

$\bar{(\cdot)}$	Averaged/Effective quantity
$\cdot$	Scalar product or single contraction of tensors
$\det$	Determinant
$\dot{(\cdot)}$	Material time derivative
$\boldsymbol{\nabla}(\cdot)$	Gradient
$\langle\cdot\rangle$	Volume average
$\|\|(\cdot)\|\|$	Euclidean tensor norm
∇	Depending on the context, one of the three differential operators gradient, divergence or rotation
$\ominus$	Erosion
$\oplus$	Dilation
$\otimes$	Dyadic product
∂	Partial derivative
$\star$	Rayleigh product
$\mathrm{tr}(\cdot)$	Trace of a matrix
d	Derivative

Quaternions

q	Unit quaternion

Contents

1 Introduction

At the beginning of this thesis, the motivation for the topic under discussion will be addressed, followed by the objectives of this work and a brief outline of the thesis structure. Subsequently, the indication of the usage of texts/graphics from the author's own publications is clarified.

1.1 Motivation

The European Green Deal obliges all EU member states to become climate-neutral by 2050 [1]. To this end, Germany has set an interim target of a 55 % reduction in greenhouse gas emissions by 2030 compared to 1990 levels in a federal climate protection law [2]. The transport sector makes a significant contribution to emissions through automobile traffic and aviation, which makes reducing emissions in these areas correspondingly important. In recent years, research has therefore been carried out in a wide variety of scientific fields, with lightweight design in particular contributing to this end in the field of materials science. This is due to the fact that the vehicle mass enters linearly into four of the five types of driving resistance and therefore plays a decisive role in fuel consumption [3]. This makes material classes that are relatively light, but at the same time offer convincing structural rigidity and strength for the safety of vehicle occupants and functional maintenance, particularly relevant. Fiber reinforced polymers are arguably the best known, relatively new material system in this context of weight-specific excellent mechanical properties [4], which is why a lot of research funding has been invested in their characterization and modelling.

So-called continuous (Co) fiber reinforced components have higher stiffness and strength due to their microstructure (high anisotropy and high fiber volume content), whereas discontinuous (Dico) fiber reinforced components are faster and easier to manufacture and offer a high degree of design freedom. The combination of both Co and Dico to synergistically exploit the advantages of both material classes is the central topic of the International Research Training Group (IRTG) 2078, within the framework of which this doctoral thesis was written. It was funded by the German Research Foundation (DFG). Dico material shows a heterogeneous microstructure, the characterization of which is essential for understanding and simulating the material behavior. Therefore, the project C2 of the IRTG 2078, in the context of which this work was created, aimed to analyze the microstructure of the Dico material used in each generation in more detail; in this third generation of carbon fiber reinforced polyamide 6. In addition to conventional macroscopic and microscopic destructive testing methods, computed tomography (CT) has emerged as probably the most important non-destructive, imaging characterization option. The 3D images, which allow the spatial differentiation of the constituents of the composites through different gray values, can provide important microstructure characterization parameters through image processing algorithms, such as fiber volume contents, fiber length distributions or fiber orientation distributions.

While the development of evaluation methods for CT images of glass fiber reinforced polymers has progressed quite far, new challenges arise when using the stiffer, stronger but also more expensive carbon fibers as reinforcement material. Since polymers and carbon fibers are chemically quite similar and hence their attenuation coefficients differ little, the contrast in the images is comparatively low. In addition, carbon fibers usually have a diameter of 5 µm - 7 µm, which is below the image resolution when scanning reasonable sizes of specimens and also often below the possible image resolution of conventional CT devices in general. This makes it impossible to identify individual fibers and results in significantly higher noise in the images. Several problems arise from these conditions for this work leading to the research objectives described in the next section.

1.2 Objectives of the thesis

First of all, the choice of an appropriate specimen size for the CT investigations is not trivial. A sample that is too large limits the maximum possible image resolution accordingly, while a sample that is too small represents microstructural phenomena that are too localized and difficult to transfer to an entire macroscopic plate of this material. In addition, the image quality is reduced in the latter case. Assuming that a suitable specimen size was chosen, there are procedures such as gradient-based fiber orientation determination that are usable despite the fact that individual fiber segmentation is not feasible. Fiber orientation information is then obtained, for example in the form of fiber orientation tensors at discrete, small areas across such a plate. This provides one of the first major questions of this work: How to obtain a continuous fiber orientation tensor field from these fiber orientation tensors at specific points, which is suitable for comparison with orientation courses resulting from process simulations or for use in stiffness modeling for comparison with experimental results? This tensorial interpolation problem entails different properties and requires alternative methods compared to scalar interpolation. Regarding this issue, three different methods are tested that differ both in their basic concept (linear algebra vs. artifical intelligence (AI)) as well as in their implementation and computational/human effort.

The evaluation of some variables, on the other hand, requires thresholding methods that do not work effectively for the scans of carbon fiber reinforced polymers, such as the determination of the fiber volume content. It is an open question whether thresholding methods can be adapted appropriately in order to work for low-contrast and non single fiber resolving images or whether switching to AI is a sensible option.

Furthermore, the creation of artificial microstructures, intended to mimic these materials and often used in material modeling in the form of representative volume elements (RVE), is challenging with conventional packing algorithms.

Again, the suitability of AI in the form of so-called generative adversarial networks is to be assessed.

In addition, the influence of the microstructure of the initial charge/plastificate as the basis of the pressing process is also to be evaluated. Specifically, the porosity and material orientation will be investigated. Summing up, the following open research questions were assessed in this work:

1. How does information on small specimens relate to large dimensions (cf. scale-bridging, up- and down-scaling), especially in relation to the interpolation of fiber orientation tensors?
2. How can characteristic quantities like the fiber volume content (that require thresholding) be extracted reliably from µCT images of carbon fiber reinforced polymers?
3. Can realistic µCT images be generated with generative adversarial networks in order to save scan resources and do they have potential in the creation of representative volume elements?
4. Underlying all of those points is the question as to whether conventional methods (linear algebra or classical image processing) are superior or inferior to artificial intelligence based solutions.

1.3 Outline of the thesis

In Chapter 2 fundamental basics of fiber reinforced polymer (FRP) material, their modeling, characteristic microstructure quantities, their image-based evaluation, computed tomography and artificial intelligence are covered to allow perspicuity of the following methods. In particular, related research contributions of these fields are studied, classified and partially reproduced. Chapter 3 focuses on the distinct description and derivation of the self-developed methods. Chapter 4 is dedicated to the detailed and objective presentation of the results of the applied methods. The assessment and quality analysis of these results

is subject of the Discussion in Chapter 5. Possible further developments and prospects for future research in these areas are also included. Finally, the work is summarized and concluded in Chapter 6.

1.4 Remark on the use of own publications in this thesis

Parts of this work were previously published in journal papers/conference proceedings by the author. If text of own publications has been reproduced in an identical manner apart from minor linguistic changes that were necessary due to different notation, numbering, etc., the section is marked with a footnote. In it, the word "extracted" is used along with the respective publication. Analogously, if graphics/tables have been published identically before, the same wording is used in their caption. In the case that graphics/tables have been newly created but are based on either own publications or other people's publications, the phrase "based on" along with the respective publication is part of the caption. However, if text has been newly written and was published by the author substantially, this work will be cited and if it affects an entire paragraph a footnote with the expression "based on" is used again. The author's publications are listed at the end of this work.

2 State of the Art

The State of the Art first describes fiber reinforced polymers in general and their mechanical description. Then, the possibilities for the quantitative description of the microstructure of discontinuous fiber reinforced polymers are shown. After a section on computed tomography (CT) and known possibilities for processing CT images, the final section introduces artificial intelligence, and in particular artificial neural networks (ANNs), convolutional neural networks (CNNs) and generative adversarial networks (GANs).

2.1 Fiber reinforced polymers

Composites are a class of materials characterized by the combination of two components that typically differ decisively in their properties. The desired optimum result is a synergy effect resulting in the composite material to perform better than the sum of the individual constituents.[5] For quite some time now, there have been various examples of composite materials that are used on a daily basis, such as ferroconcrete in construction or laminated composites such as plywood. In this context, FRP represent a special group whose further scientific development was mainly sparked by their enormous potential in lightweight design. Owing to their high density-specific strength and stiffness, FRP are used in the transport industry (automotive, aeronautical), sports industry as well as in the energy sector, e.g., in wind turbine blades. They consist of fibers with high strength and stiffness embedded in a typically low-density polymer matrix which gives the fibers a position structurally, transfers the load to them [5] and protects them from environmental influences [6]. Typical fiber materials

include glass with relatively low cost and high strength, carbon, which is even stronger and stiffer, but also significantly more expensive, or Aramid, and now increasingly natural fibers such as flax or hemp [5]. The thin fiber shape is not chosen arbitrarily. According to the weakest link theory, the so-called size effect describes the statistical decrease of strength-reducing defects in a smaller material volume [7, 8], as well as the reduction of the size of these defects, which is also the basis of the statistical theory of brittle fracture dating back to Weibull [9]. The matrix material can either be a thermoset or a thermoplastic material (in rare cases also elastomers). While the former was used more often in engineering, load-bearing applications in the past due to their high strength and stiffness and their minor sensitivity towards environmental aspects, the use of the latter is increasing in recent years [10]. Without covalent bonds between the chains, thermoplastics exhibit lower effective mechanical properties [5]. Furthermore, they are susceptible to temperature and - dependent on the exact substance - also to moisture, albeit to a more varying degree [11]. However, they are meltable, which enables them to have superior recycling potential, and they can be manufactured rapidly, which reduces production costs [10].

2.1.1 Fiber reinforcement architectures

Fibers can be incorporated into the matrix in a variety of ways. The dimension and shape of the reinforcing phase is decisive for the mechanical properties, and each type of reinforcement is also directly related to the process developed for it. [5]

2.1.1.1 Continuous (Co) fiber reinforcement

Continuous fiber reinforcement signifies the deliberate insertion of fiber rovings in a specific direction, where the length of the fibers usually exceeds or equals that of the part. For the latter reason, it is often also referred to as endless fiber reinforcement. Continuous fiber reinforced polymers (CoFRP) are normally

made up of different single layers, each with a defined orientation. There are pre-impregnated semi-finished products, which are usually available in sheet, tape or strand form. These include prepregs such as unidirectional (UD) tapes, which, apart from layers with a 0° orientation, also always have layers with a 45° or 90° orientation. There are also pure fiber semi-finished products, so-called textiles (sometimes also referred to as fabrics), which are not pre-impregnated. In the latter case, a distinction is made between the widespread non-crimp fabrics and woven, braided and knitted fabrics. Both types of semi-finished products can ultimately be stacked in so-called laminates. By selecting the arrangement of plies, the material behavior can be tailored to the main load directions. As a result, it can range from almost unidirectional to quasi planarisotropic properties. [5] All continuously reinforced polymers have in common that they exhibit outstandingly good specific mechanical properties in comparison to DicoFRP and a less complex and thus easy-to-characterize microstructure, but in return they offer little design freedom [12].

2.1.1.2 Discontinuous (Dico) fiber reinforcement

In the case of discontinuous fibers, either a cut to a defined length below the component length takes place in the process, or the fibers are cut by shearing in extruders to a length distribution that tends to be dominated by relatively short fibers compared to CoFRP. They therefore typically have a more complex, heterogeneous microstructure, while both single fibers [13] and fiber bundles [14] may be encountered. In general, DicoFRP exist with thermoplastics and thermosets. The best known type of the latter are so-called sheet molding compounds (SMC). SMC is a typical representative of a bundle structure [14]. Discontinuous fiber reinforcement can be further subdivided into short fiber and long fiber reinforced polymers, abbreviated as SFT and LFT for thermoplastics [5]. The distinction is made on the basis of the aspect ratio $a_\mathrm{r} = \frac{l}{d}$ (length to diameter), with a value below 100 still considered a short fiber material and a value above 100 a long fiber material [15].

Long fiber reinforced thermoplastics (LFT)

LFTs are characterized by their good processability and the possibility of economical production of larger batches compared to CoFRP, as well as typically better mechanical properties than SFT due to their longer fibers [5]. Together with the possibility of recyclability by melting down the thermoplastic, this group presents a particularly interesting combination of challenges and opportunities, which is why they are subject of this work. They are typically produced in extrusion compression molding processes. These start with the plastification of the polymer-fiber mixture in an extruder and the subsequent placement of the produced plastificate/initial charge in a mold cavity in a press. It is then followed by the closure of the mold resulting in the final product, e.g., in the shape of a plate, which was mostly used in this work. There is another distinction between the so-called LFT-G (granulate) and LFT-D (direct) process. The two processes differ in that semi-finished products, i.e. pellets containing thermoplastic matrix and long fibers, are processed in the LFT-G process. The LFT-D process, on the contrary, which is the production method of the material used in this dissertation, is characterized by the in-line processing of thermoplastic granulate and the direct introduction of continuous fiber rovings into the polymer melt (cf. Figure 2.1).

The advantages of the LFT-D process are the independence of prefabricated semi-finished products and the associated cost benefits, high output capacities and flexibility in fiber and polymer adaptations [18]. Thermoplastics like polyethylenterephthalat (PET), polycarbonate (PC), polypropylene (PP) or acrylonitrile butadiene styrene (ABS) can be used in this process. In this work, polyamide 6 (PA6) was the thermoplastic matrix material used.

2.1.1.3 Continuous discontinuous (CoDico) fiber reinforcement

In order to profitably exploit the contrasting advantages of continuous and discontinuous fiber reinforced polymers (cf. Table 2.1), the idea of continuous

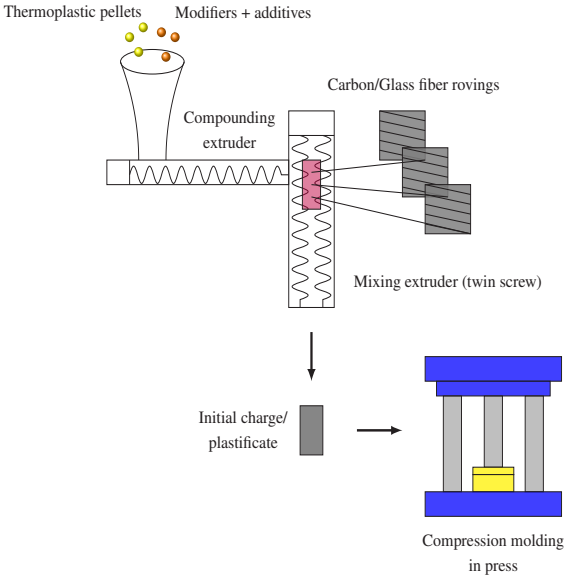


Figure 2.1: Schematic representation of the long fiber reinforced thermoplastic direct (LFT-D) process. Based on [16] and Blarr et al. [17].

discontinuous fiber reinforced polymers, or CoDicoFRP for short, evolved [12]. It makes use of the design freedom and cost-effective production of DicoFRP while locally reinforcing the component with CoFRP at points subject to particularly intensive stress.

Table 2.1: Advantages and disadvantages of CoFRP and DicoFRP to illustrate the effect of joint use. Data based on [12].

Aspect	Co	Dico
Mass-specific mechanical properties	+	-
High fiber volume content	+	-
Degree of fiber alignment	+	-
Low microstructure complexity	+	-
Easy formability	-	+
Low manufacturing cost	-	+
High production rate	-	+
Low waste rate	-	+

2.1.2 Basic elastic mechanics of FRP

In this section, elastic mechanics of FRP are presented, starting with the basic homogenization equations and followed by macroscale modeling approaches like the rule of mixture and corresponding Reuss and Voigt bounds as well as the Halpin-Tsai model. The concept of critical fiber length for DicoFRP is also explained in this context. After outlining the idea of microscale and RVE models as well as mean field models, the Mori-Tanaka model is given as a typical representative of the latter.

2.1.2.1 Homogenization

In continuum mechanics, homogenization is understood as the attempt to describe a heterogeneous material by means of a model that represents this complex structure in a highly simplified way and yet leads to a plausible deformation behavior, i.e., macroscopically equivalent as the heterogeneous material, for the load cases investigated. Hence, there are a local stress $\boldsymbol{\sigma}(\boldsymbol{x})$ and a local strain $\boldsymbol{\varepsilon}(\boldsymbol{x})$ that are heterogeneously distributed in the material. They are related to

the overall global behavior, so that the effective stress $\bar{\boldsymbol{\sigma}}$ and effective strain $\bar{\boldsymbol{\varepsilon}}$ are given by so-called localization tensors $\mathbb{L}$ and $\mathbb{M}$ in the following way:

$$\begin{aligned} \boldsymbol{\varepsilon}(\boldsymbol{x}) &= \mathbb{L}(\boldsymbol{x})[\bar{\boldsymbol{\varepsilon}}], \\ \boldsymbol{\sigma}(\boldsymbol{x}) &= \mathbb{M}(\boldsymbol{x})[\bar{\boldsymbol{\sigma}}]. \end{aligned} \tag{2.1}$$

The following applies to the localization tensors

$$\langle \mathbb{L}(\boldsymbol{x}) \rangle = \mathbb{I}^S \quad \text{and} \quad \langle \mathbb{M}(\boldsymbol{x}) \rangle = \mathbb{I}^S. \tag{2.2}$$

Applying the Hill-Mandel condition [19, 20], which states that the work of an entire system does not change viewing it from the microscale or the macroscale, the effective stress $\bar{\boldsymbol{\sigma}}$ and effective strain $\bar{\boldsymbol{\varepsilon}}$ can be given by a volume averaging $\langle \cdot \rangle$ over the representative volume V:

$$\begin{aligned} \bar{\boldsymbol{\sigma}} = \langle \boldsymbol{\sigma}(\boldsymbol{x}) \rangle &= \frac{1}{V} \int_V \boldsymbol{\sigma}(\boldsymbol{x}) \quad dV, \\ \bar{\boldsymbol{\varepsilon}} = \langle \boldsymbol{\varepsilon}(\boldsymbol{x}) \rangle &= \frac{1}{V} \int_V \boldsymbol{\varepsilon}(\boldsymbol{x}) \quad dV. \end{aligned} \tag{2.3}$$

Assuming linear elastic behavior, the relation between effective stress $\bar{\boldsymbol{\sigma}}$ and effective strain $\bar{\boldsymbol{\varepsilon}}$ is given by the generalized Hooke's law

$$\bar{\boldsymbol{\sigma}} = \bar{\mathbb{C}}[\bar{\boldsymbol{\varepsilon}}] = \langle \mathbb{C}(\boldsymbol{x})[\boldsymbol{\varepsilon}(\boldsymbol{x})] \rangle, \tag{2.4}$$

with the local stiffness tensor $\mathbb{C}(\boldsymbol{x})$ and the effective stiffness tensor $\bar{\boldsymbol{\sigma}}$. The effective stiffness tensor $\bar{\mathbb{C}}$ analogously is defined by

$$\bar{\mathbb{C}} = \langle \mathbb{C}(\boldsymbol{x}) \mathbb{L}(\boldsymbol{x}) \rangle. \tag{2.5}$$

2.1.2.2 Macroscale modeling

When modeling composites, fundamentally different approaches have to be considered. The previous homogenization equations are the basis for the following rule of mixture (ROM) and the Voigt and Reuss bounds as first-order energy principles, which can be classified as a macroscopic homogenization approach, as well as the semi-empirical approach of Halpin-Tsai. When modeling on the macroscale, the material is considered as a homogeneous, anisotropic material. Stresses and strains in different directions can be determined by experiments, and reverse-engineering can be used to determine the unknown constants in the phenomenological models ultimately. Although these methods are an obvious simplification of the true situation, since they ignore locally varying material properties that a composite always exhibits, they are popular due to their computational simplicity, especially in industry or for large systems. [5]

Rule of mixture (Reuss and Voigt bounds)

The simplest micromechanical, energy-based model is the so-called rule of mixture (ROM). Assuming a perfectly aligned unidirectional fiber reinforced composite under uniaxial tension σ_{11} (cf. Figure 2.2) and a still intact material, the strains of matrix and fibers have to be equal as they are bonded to each other.

This follows the so-called Voigt assumption [21]. Furthermore assuming linear elastic behavior of both constituents according to Hooke, this can be written as

$$\varepsilon_{\mathrm{c}} = \varepsilon_{\mathrm{m}} = \varepsilon_{\mathrm{f}} = \frac{\sigma_{\mathrm{c}}}{E_{\mathrm{c}}} = \frac{\sigma_{\mathrm{m}}}{E_{\mathrm{m}}} = \frac{\sigma_{\mathrm{f}}}{E_{\mathrm{f}}}. \tag{2.6}$$

In this and in the following equations, the subscripts c, m and f describe the respective composite, matrix and fiber property.

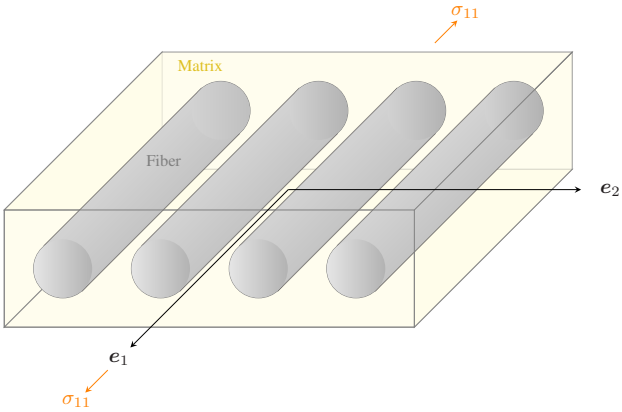


Figure 2.2: Schematic graphic of a perfectly unidirectional continuous fiber reinforced polymer under uniaxial tension σ_{11}. Based on [5].

Based on the premise that stress is defined by

$$\sigma = \frac{F}{A}, \tag{2.7}$$

the force equilibrium

$$F_\mathrm{c} = F_\mathrm{m} + F_\mathrm{f} \tag{2.8}$$

can be rewritten with the help of Equation (2.6) as

$$E_\mathrm{c}\varepsilon_\mathrm{c}A_\mathrm{c} = E_\mathrm{m}\varepsilon_\mathrm{m}A_\mathrm{m} + E_\mathrm{f}\varepsilon_\mathrm{f}A_\mathrm{f} \tag{2.9}$$

and further as

$$E_\mathrm{c} = E_\mathrm{m}\frac{A_\mathrm{m}}{A_\mathrm{c}} + E_\mathrm{f}\frac{A_\mathrm{f}}{A_\mathrm{c}}. \tag{2.10}$$

Introducing the fiber volume content/fraction v_f this results in

$$E_{11} = E_\mathrm{c} = E_\mathrm{m}(1 - v_\mathrm{f}) + E_\mathrm{f} v_\mathrm{f}. \tag{2.11}$$

It shall be mentioned that while under the given conditions, the shear and bulk modulus follow the ROM, the Young's modulus does only obey the ROM in the case that the Poisson ratio of both fiber and matrix are equal [22]. Otherwise, which is the normal scenario, the real Young's modulus in longitudinal direction E_{11} would actually be larger than given by Equation (2.11), which is the first-order upper bound of Voigt.

Considering the transverse direction, thus a load perpendicular to the fibers σ_{22}, following the Reuss assumption [23] of equal stresses $\sigma_{22} = \sigma_\mathrm{c} = \sigma_\mathrm{m} = \sigma_\mathrm{f}$, one obtains the inverse ROM,

$$E_{22} = E_\mathrm{c} = \left(\frac{1 - v_\mathrm{f}}{E_\mathrm{m}} + \frac{v_\mathrm{f}}{E_\mathrm{f}} \right)^{-1}, \tag{2.12}$$

as the first-order lower bound modulus. It was first shown by Hill in 1952 that those are upper and lower bounds [24]. They are visually displayed in Figure 2.3. The Voigt bound is close to the actual value, as a parallel connection in the fiber direction is a fairly good approximation. The Reuss bound, on the other hand, is much further away, as a series connection in the transverse direction is not a realistic analogy. Accordingly, all material properties of a composite (like the Young's modulus) are located between the two limits. This applies both to the case where the loading is not entirely longitudinal or transverse, but a mixed loading is present, and to the case where there is no optimally uniaxially reinforced FRP, but, for example, a discontinuously reinforced composite. The Voigt and Reuss bounds also directly result from Equation (2.5) in Section 2.1.2.1. After introducing two phases (matrix, fiber) and a corresponding volume fraction of fibers v_f, the equation describing Voigt's upper limit (cf. Equation (2.11)) is reattained easily. By defining the compliance

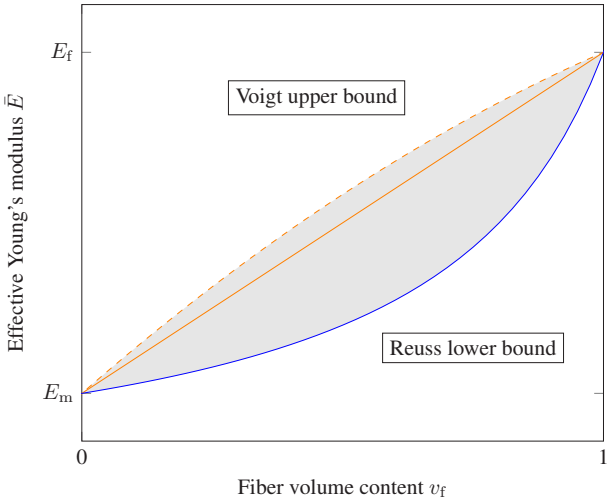


Figure 2.3: Graphic showing the effect of the fiber volume content v_f on the effective Young's modulus $\bar{E}$ of a composite (gray area) with first-order upper Voigt (orange) and lower Reuss bound (blue). The solid orange line signifies the upper bound in the case that both constituents have the same Poisson ratio, the dashed orange line is an elevated adaption for unequal Poisson ratios. Based on [22, 25].

tensor $\mathbb{S} = \mathbb{C}^{-1}$ as the inverse stiffness tensor, the homogenized stiffness tensor of Reuss (cf. Equation (2.12)) is also obtained again.

Hashin–Shtrikman [26] actually showed in 1962 using variational principles that the upper and lower bounds for the elastic moduli can be more narrow than the Voigt and Reuss bounds, which will not be discussed further here. Even with this simplified model, the relevance of local fiber volume contents for homogenization of composites gets conveyed.

Concept of critical fiber length

In the case of DicoFRP, the fibers do not extend over the entire length of the component. Naturally, this raises the question of whether the concept of load transfer between matrix and fiber therefore still works without restriction. In

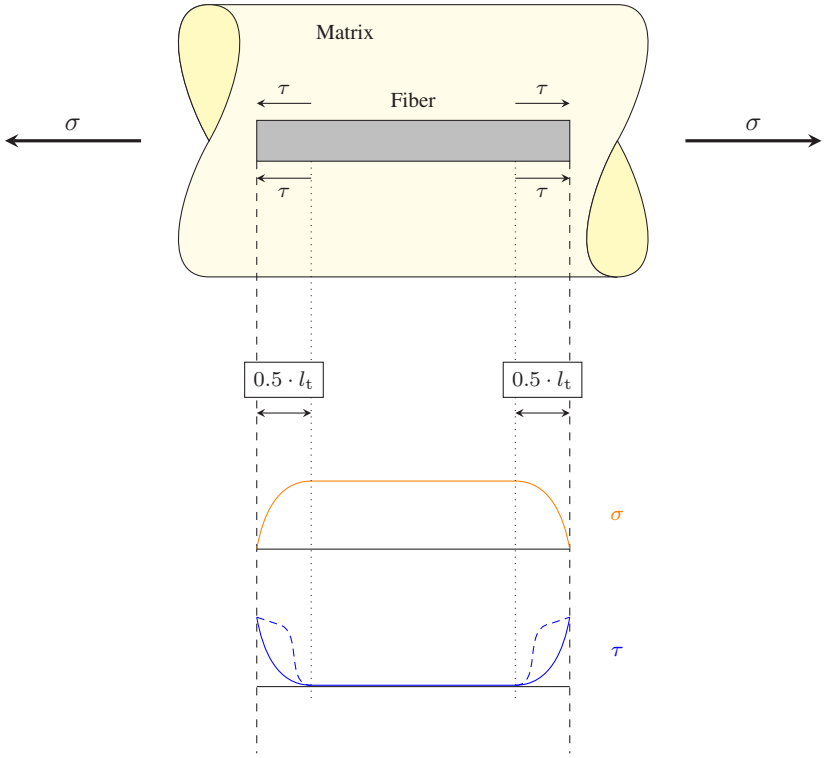


Figure 2.4: Schematic of a single discontinuous fiber embedded in a matrix. The composite experiences tensile stress in fiber direction. The respective load transfer from matrix to fiber is depicted as qualitative plots of the tensile stress distribution in the fiber and the shear stress distribution in the matrix. The blue line of the shear stress corresponds to the material response of a fully elastic matrix (roughly that of a thermoset), whereas the dashed blue line relates to a viscoelastic matrix, hence a thermoplastic material. Based on [5, 27].

order to approach this issue, the model of a single, discontinuous fiber embedded in a cylindrical matrix is used (cf. Figure 2.4).

When this composition is subjected to tensile stress along the fiber direction, the matrix deforms and thereby transmits load to the fiber through shear, especially at both ends of the fibers. The shear stress decreases the further one gets away

from the edge of the fibers. The maximum normal stress is therefore induced in the fiber when the fiber is long enough for the shear stress to become zero before the middle of the fiber length. This allows maximum force transfer to occur, loading the favorable tensile properties of the fibers instead of the more critical interface properties. The course of the shear stress at the interface depends on the matrix type. In the case of mostly elastically deforming thermosets, a sharp peak appears at the end of the fiber (blue curve in Figure 2.4), whereas the viscoelastic material behavior of thermoplastics tends to provide a softer transition (blue dashed curve in Figure 2.4). [5]

Assuming - as a modeling simplification - that in the case of the thermoplastic matrix the shear stress at the edge increases abruptly to τ and then remains constant at this value up to the edge of the fiber, the shear force at the interface is given by

$$F_{\mathrm{Shear}} = C_{\mathrm{f}} \tau \frac{1}{2} l_{\mathrm{t}} = \frac{\pi d l_{\mathrm{t}} \tau}{2}, \tag{2.13}$$

with the circumference of the fiber $C_{\mathrm{f}} = \pi d$ and the length of the fiber necessary for load transfer l_{t}. The maximum fiber force can be calculated by

$$F_{\mathrm{f}} = A_{\mathrm{f}} \sigma_{\mathrm{f}} = \frac{\pi d^2 \sigma_{\mathrm{f}}}{4}, \tag{2.14}$$

with the area of the fiber $A_{\mathrm{f}} = \pi r^2 = \pi \frac{d^2}{4}$.

Using Equation (2.6), as the load is parallel to the fiber direction, leads to

$$\frac{\sigma_{\mathrm{f}}}{E_{\mathrm{f}}} = \frac{\sigma}{E_{11}}. \tag{2.15}$$

In order to obtain the length necessary to transmit the entire load from the matrix to the fiber, the shear force and the force in the fiber have to be equated, leading to

$$l_{\mathrm{t}} = \frac{\sigma_{\mathrm{f}} d}{2\tau} = \frac{E_{\mathrm{f}}}{E_{11}} \frac{\sigma d}{2\tau}. \tag{2.16}$$

In the case of failure, either the fiber breaks or the interface fails. Optimally, both occurs simultaneously, reaching the bonding shear strength τ_{b} and the ultimate fiber strength σ_{fu}. This leads to the reformulation of Equation (2.16) as

$$l_{\mathrm{c}} = \frac{\sigma_{\mathrm{fu}} d}{2\tau_{\mathrm{b}}}, \tag{2.17}$$

with l_{c} being called the critical fiber length. [5]

The importance of a minimum fiber length and in general the influence of fiber length becomes clear in this simplified model. In the following, homogenization approaches will be discussed which explicitly take more account of such microstructural properties.

Halpin-Tsai

As a scalar and one of the most simple methods, the Halpin-Tsai model [28] approaches short-fiber reinforced composite materials by including the length to diameter ratio $\frac{l}{d}$. In the Halpin Tsai equations, a load-direction dependent empirical factor ξ is introduced, defined as follows:

$$\begin{aligned} \xi_{11} &= 2\left(\frac{l}{d}\right) \quad \text{(longitudinal direction)}, \\ \xi_{22} &= 2 \qquad\quad \text{(transverse direction)}. \end{aligned} \tag{2.18}$$

It is therefore a semi-empirical homogenization method and an adaptation of Voigt ($\xi = \infty$) and Reuss ($\xi = 0$). While this geometry factor ξ could be expressed correctly as a combination of elastic constants and differences in

Poisson ratios, the reasonable assumption that the engineering stiffness equations are insensitive towards differences of the Poisson ratios was made [29]. The validity of the expressions/values for ξ developed from this assumption was verified both experimentally and by computer calculations of the exact equations [29]. Additionally defining η and λ as

$$\begin{aligned} \eta_{11,22} &= \frac{\frac{E_\mathrm{f}}{E_\mathrm{m}} - 1}{\frac{E_\mathrm{f}}{E_\mathrm{m}} + \xi_{11,22}}, \\ \lambda &= \frac{\frac{G_\mathrm{f}}{G_\mathrm{m}} - 1}{\frac{G_\mathrm{f}}{G_\mathrm{m}} + 1}, \end{aligned} \tag{2.19}$$

leads to the following equations for the longitudinal and transverse Young's modulus E_{11} and E_{22} and for the shear modulus G_{12}

$$\begin{aligned} E_{11} &= E_\mathrm{m}\Big(\frac{1 + \xi_{11}\eta_{11}v_\mathrm{f}}{1 - \eta_{11}v_\mathrm{f}}\Big), \\ E_{22} &= E_\mathrm{m}\Big(\frac{1 + \xi_{22}\eta_{22}v_\mathrm{f}}{1 - \eta_{22}v_\mathrm{f}}\Big), \\ G_{12} &= G_\mathrm{m}\Big(\frac{1 + \lambda v_\mathrm{f}}{1 - \lambda v_\mathrm{f}}\Big). \end{aligned} \tag{2.20}$$

For $\frac{l}{d} \to \infty$, hence in the case of a CoFRP, these equations can be converted, e.g., with the help of L'Hôpital's rule, into the ROM (cf. Equation (2.11)).

The Poisson ratios are defined as

$$\begin{aligned} \nu_{21} &= \nu_\mathrm{m}(1 - v_\mathrm{f}) + \nu_{\mathrm{f}21}v_\mathrm{f}, \\ \nu_{12} &= \nu_{21}\frac{E_{22}}{E_{11}}, \end{aligned} \tag{2.21}$$

with the first index describing the contraction direction and the second index the load direction. [5]

The previous equations include the fiber volume content and also a (at least average) fiber length. The stress-strain relation is given as

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \underbrace{\begin{pmatrix} C_{11} & C_{12} & C_{16} \\ C_{12} & C_{22} & C_{26} \\ C_{16} & C_{26} & C_{66} \end{pmatrix}}_{=C_{ij}} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix}, \tag{2.22}$$

with the components of the stiffness tensor defined by

$$C_{11} = \frac{E_{11}}{1 - \nu_{12}\nu_{21}}, \tag{2.23a}$$

$$C_{12} = \nu_{21} C_{11}, \tag{2.23b}$$

$$C_{16} = 0, \tag{2.23c}$$

$$C_{22} = \frac{E_{22}}{1 - \nu_{12}\nu_{21}}, \tag{2.23d}$$

$$C_{26} = 0, \tag{2.23e}$$

$$C_{66} = G_{12}, \tag{2.23f}$$

if the loading directions coincide with the main fiber orientation. In the case that the load and fiber direction of the composite are now offset by an angle θ, the components of the off-axis stiffness matrix C'_{ij} can be calculated as follows:

$$\begin{bmatrix} C'_{11} \\ C'_{22} \\ C'_{12} \\ C'_{66} \\ C'_{16} \\ C'_{26} \end{bmatrix} = \begin{pmatrix} c^4 & s^4 & 2c^2s^2 & 4c^2s^2 \\ s^4 & c^4 & 2c^2s^2 & 4c^2s^2 \\ c^2s^2 & c^2s^2 & c^4+s^4 & -4c^2s^2 \\ c^2s^2 & c^2s^2 & -2c^2s^2 & (c^2-s^2)^2 \\ c^3s & -cs^3 & cs^3-c^3s & 2(cs^3-c^3s) \\ cs^3 & -c^3s & c^3s-cs^3 & 2(c^3s-cs^3) \end{pmatrix} \begin{bmatrix} C_{11} \\ C_{22} \\ C_{12} \\ C_{66} \end{bmatrix}, \tag{2.24}$$

where $c = \cos\theta$ and $s = \sin\theta$. [30] The equations are given in Voigt notation. This way, the fiber orientation can be considered in the Halpin-Tsai homogenization through the inclusion of the amount of fibers aligned in a specific angle. In practice, the occurrence of fibers in a specific angle range of, e.g., 20°, is incorporated instead.

Shear-lag modified Halpin Tsai

Another well-known model for unidirectional discontinuous short fiber reinforced polymers is the so-called shear-lag model according to Cox [31]. The name is based on the fact that the fiber and matrix experience different displacements when a load is applied. This leads to a "delay" in the displacement near the fiber-matrix interface compared to the matrix. A precise derivation can be found in the work by Fu et al. [32]. The final result is the following equation for the stiffness in fiber direction:

$$E_{\mathrm{L}} = E_{\mathrm{f}} v_{\mathrm{f}} \left(1 - \frac{\tanh(\chi l/2)}{\chi l/2}\right) + E_{\mathrm{m}}(1 - v_{\mathrm{f}}), \tag{2.25}$$

with l as the average fiber length and χ given by

$$\chi = \sqrt{\frac{2G_{\mathrm{m}}}{r_{\mathrm{f}}^2 E_{\mathrm{f}} \ln\left(R/r_{\mathrm{f}}\right)}}. \tag{2.26}$$

Therein, the average fiber radius is given by r_{f} and R denotes the mean separation of the fibers normal to their length. Equation (2.25) can then be used as E_{11} in the Halpin-Tsai model (cf. Equation (2.20)) resulting in the shear-lag modified Halpin-Tsai equations. To the author's knowledge the use of shear-lag in the Halpin-Tsai model was first considered by Fu et al. in 2002 [30].

2.1.2.3 Microscale/Multiscale modeling

When modeling on the microscale, on the other hand, every fiber, the matrix, each void and the geometric arrangement has to be considered. For this purpose, RVEs are used at the micro level (µm). While this is possible on a small scale, with glass fibers being between 14 µm and 20 µm in diameter and carbon fibers between 5 µm and 7 µm, there can be so many fibers in just one square millimeter that this is hardly a realistic approach computationally when considered at the component level. If an RVE is used for homogenization and coupled with macroscale modeling (mm), this is considered multiscale modeling. These unit cells in microscale modeling should be chosen to be as small as possible but as large as necessary. This means that they should optimally represent the global, effective material behavior and not only local features, while at the same time being small enough for efficient computations. The choice of the size of this RVE is therefore challenging, especially for DicoFRP. The material properties can then be homogenized within the RVE. The macroscopic material behavior can be obtained by assembling a component in a finite element (FE) model recurrently from copies of this RVE. [5]

Depending on the microstructure, multiscale modeling can also mean using a material unit discretely on the mesoscale. Using the example of SMC, which has a typical bundle structure, the fiber bundles can be segmented as a unit from CT images [14]. This bundle-matrix structure can then be the basis for

structural material simulation [33] as well as for a mesoscale simulation of the mold filling process [34].

Microstructure generation for RVEs

In this context, the creation of an RVE that can be used discretely in simulations is an entirely new issue. There are programs that can basically automatically create a discrete RVE model from a CT image. However, this only works for very high-resolution and high-contrast CT images and comes with certain limitations in accuracy. Classically, image analysis of CT images and experimental procedures are used to determine quantities such as fiber orientation, fiber length distribution, fiber volume content and pores for the size of the intended RVE and then an attempt is made to recreate the microstructure from this information. Therefore, microstructure generators are programmed. These are typically based on so-called sphere-packing algorithms. Therein, one attempts to pack particles or fibers into a cell in a non-overlapping manner and to match the specified properties such as fiber volume content, fiber orientation and fiber lengths as precisely as possible. In order to achieve the packing, scientist have relied on a wide variety of approaches, such as mechanical compression or more chemically influenced ideas of molecular dynamics, right up to a shrinking RVE cell. Typical examples are random sequential adsorption (RSA) [35], the approaches by Lubachevsky and Stillinger [36] or Torquato and Jiao [37] and mechanical contraction [38]. These packing algorithms have limitations. Particularly high fiber volume contents are a problem for them, but curvature of the fibers is also difficult to incorporate. However, both effects were observed in the material used in this work. In particular, the visible mixture of fiber bundles and individual fibers cannot be covered, even with newer, improved or extended approaches. Therefore, this topic will be addressed again in Section 2.2.4 in the State of the Art. With the use of AI to generate CT images of microstructures, a completely new direction of microstructure generation and possibly eventually RVE generation will be opened up.

2.1.2.4 Mean field homogenization

Mean field homogenization can be regarded as a middle ground between the homogenization approaches mentioned so far. The aim is to determine mean material parameters for a certain volume, which is supposed to represent the microstructure of the composite. The important assumption here is that each fiber is an ellipsoid in the matrix and the aspect ratio of the ellipse represents the aspect ratio $\left(\frac{l}{d}\right)$ of the fiber. It is further assumed that the fibers are straight and well distributed and that there is no fiber-fiber interaction. The homogenized material properties are then to be determined for this RVE with the simplified structure. [5, 39, 40]

In the context of mean field homogenization, the analytical solution for stress and strain in an infinite linear-elastic body with an ellipsoidal inclusion by Eshelby in 1957 was decisive [41]. He found that there is a uniform strain $\boldsymbol{\varepsilon}_{\mathrm{inclusion}}$ in the inclusion when it is exposed to eigen-strain $\boldsymbol{\varepsilon}_{\mathrm{m}}$, related by

$$\boldsymbol{\varepsilon}_{\mathrm{inclusion}} = \mathbb{N}\boldsymbol{\varepsilon}_{\mathrm{m}}. \tag{2.27}$$

The strain in the infinite elastic body and the ellipsoidal inclusion are therefore not equal. $\mathbb{N}$ is a fourth-order tensor that is a function of the fourth-order stiffness tensors of matrix $\mathbb{C}_{\mathrm{m}}$ and inclusion $\mathbb{C}_{\mathrm{f}}$ defined by

$$\mathbb{N} = \left[\mathbb{I} - \mathbb{E}\mathbb{C}_{\mathrm{m}}^{-1}(\mathbb{C}_{\mathrm{f}} - \mathbb{C}_{\mathrm{m}})\right]^{-1}, \tag{2.28}$$

with $\mathbb{I}$ as the fourth-order identity tensor and $\mathbb{E}$ being the fourth-order Eshelby tensor, whose components are defined as a function of the aspect ratio of the inclusion (with which the average fiber length is taken into account) and the Poisson ratio of the elastic body around it. The exact definitions of the components can be retrieved from the original paper [41]. [5] An overview of the different options of homogenizing a composite can be seen in Figure 2.5.

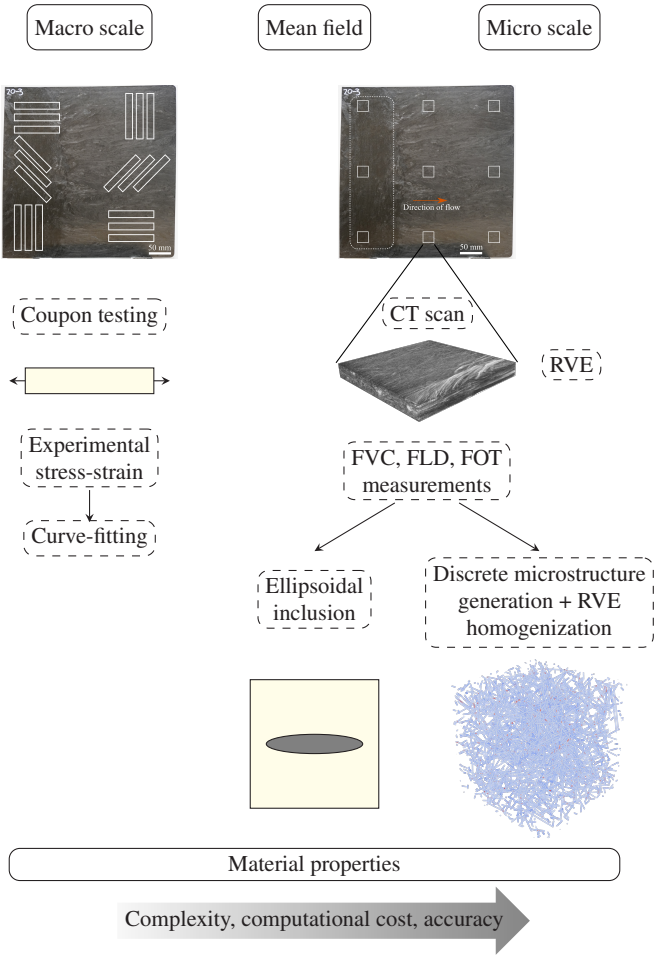


Figure 2.5: Overview of the various homogenization options for a fiber reinforced composite with increasing complexity from left to right. For further explanation, the corresponding text sections should be consulted. Images of the CF-PA6 plates used in this work are shown at the top, with possible specimen geometries for different evaluation methods. The 3D view of the scan is an image of a 25 mm × 25 mm × 3 mm GF-PA6 sample. The generated microstructure on the far right is a specially created microstructure with 1000 fibers and a fiber volume content of 0.2, which was implemented using an RSA algorithm in which spherocylinders were packed.

Mori-Tanaka

As Eshelby deduced the above expressions only for a single ellipsoidal inclusion, no interactions between those inclusions are allowed for and hence the stress in the inclusion can be caused by the matrix only. As this is not the case in a composite, Mori and Tanaka [42] extended this model to a two-phase system with interactions between the inclusions, resulting in one of the most well-known mean field homogenizations. [5]

A new tensor of fourth-order $\mathbb{T}$ is introduced as an adaption of $\mathbb{N}$, leading to

$$\boldsymbol{\varepsilon}_{\mathrm{inclusion}} = \mathbb{T}\boldsymbol{\varepsilon}_{\mathrm{m}}. \tag{2.29}$$

$\mathbb{T}$ is given depending on $\mathbb{N}$ and the volume fraction v_{f} as

$$\mathbb{N} = \Big[(1 - v_{\mathrm{f}})\mathbb{I} + v_{\mathrm{f}}\mathbb{T}\Big]^{-1}\mathbb{T}. \tag{2.30}$$

If the volume content of the fibers is zero, $\mathbb{T}$ and $\mathbb{N}$ are equal. The stiffness tensor of the composite can then be calculated as

$$\mathbb{C}_{\mathrm{c}} = \bar{\mathbb{C}} = \mathbb{C}_{\mathrm{m}} + v_{\mathrm{f}}\langle(\mathbb{C}_{\mathrm{f}} - \mathbb{C}_{\mathrm{m}})\mathbb{T}\rangle. \tag{2.31}$$

The fiber volume content and the aspect ratio are already taken into account in this expression. There are subsequently two options for also taking the fiber orientation into account if it is not a unidirectional material. The reference tensor $\mathbb{H}^0$ (see Section 2.2.3.1 and [43]) can either be multiplied subsequently after the stiffness determination. This corresponds to a linear dependency. Alternatively, the orientation determination can already be coupled directly to the tensor $\mathbb{T}$, which no longer complies with linearity in the dependency and, to the author's knowledge, was introduced for the first time by Benveniste [44]. More details can be found in the works of Bauer and Böhlke [45] and Benveniste [44]. The

former paper also shows that this makes no major difference in the effective values for one type of fiber in the composite. However, the linear adjustment of the orientation no longer makes sense if, for example, there are two fiber types with different orientations in a hybrid composite.

2.2 Quantitative metrics of DicoFRP microstructures and their determination

Over time, various parameters have been developed for the quantitative description of the complex, heterogeneous microstructure of discontinuous fiber reinforced polymers. These are outlined below and established methods - whether destructive or non-destructive, experimental or by image evaluation - that lead to the determination of these quantities are described.

2.2.1 Fiber volume content

The fiber volume content (FVC), often also referred to as fiber volume fraction (FVF) and in this work denoted as v_{f}, determines the proportion of fibers in the total volume of the material. In general, a higher fiber concentration implies a higher elastic modulus, as well as higher strength and better impact behavior, although the effect on the latter two typically inverses at some point (at least for DicoFRP) due to the formation of non-impregnated bundles [5].

2.2.1.1 Experimental destructive methods for FVC determination

A first starting point to get an approximate FVC of a plate/part is, of course, the value nominally set in the process during fiber feeding. However, since this information is quite inaccurate and the fiber content varies across a plate due to the process, a measurement of the FVC of the finished material is of interest. Typically, a smaller sample is cut out of a part and the fiber volume content is

measured destructively by removing the matrix and then measuring the fiber weight compared to the total weight of the sample. This method is also known as thermogravimetric analysis (TGA). The polymer is most commonly removed by incineration, also known as pyrolysis, as it is the most straightforward approach [5]. The temperature and time profile of the incineration process must be adjusted individually for each polymer, exemplified by Rohde et al. [46] for polypropylene. The decisions also depend on the time available and whether the fibers are measured further afterwards and therefore may not become more brittle or sintered. For glass fibers, this works well in principle, but for carbon fibers, which are the most frequently treated in this work, problems arose because the carbon fibers can be attacked by the burning process. If the specimens were incinerated too short or at too low a temperature, the matrix was not yet completely dissolved and the fibers were therefore still attached to each other. However, if the specimens were processed longer or at higher temperatures, the fibers were already visibly attacked. Due to the experienced difficulties of pyrolysis for CFRP, this method was not applied further after initial tests. Alternatively, the matrix can be dissolved using acid. Here too, experience for certain polymers plays a role in the exact test procedure (choice of different acids, their concentration, time periods and temperatures). The exact method and the parameter values will follow in the Methods section. Subsequently, when the fibers are separated, they are weighed with a high-precision scale and divided by the total mass m_{tot} of the sample to obtain the fiber mass content:

$$w_{\mathrm{f}} = \frac{m_{\mathrm{f}}}{m_{\mathrm{tot}}}. \tag{2.32}$$

To determine the fiber volume content on this basis, the densities of the constituents and the trivial relationship $m_{\mathrm{m}} = m_{\mathrm{tot}} - m_{\mathrm{f}}$ must be used, leading to

$$v_\mathrm{f} = \frac{\frac{m_\mathrm{f}}{\rho_\mathrm{f}}}{\frac{m_\mathrm{f}}{\rho_\mathrm{f}} + \frac{m_\mathrm{m}}{\rho_\mathrm{m}}}. \tag{2.33}$$

2.2.1.2 Determination of FVC through consideration of composite density

With the density of the composite ρ_c defined as

$$\rho_\mathrm{c} = w_\mathrm{f}\rho_\mathrm{f} + (1 - w_\mathrm{f})\rho_\mathrm{m}, \tag{2.34}$$

the fiber weight content w_f can be determined as

$$w_\mathrm{f} = \frac{\rho_\mathrm{c} - \rho_\mathrm{m}}{\rho_\mathrm{f} - \rho_\mathrm{m}}. \tag{2.35}$$

The density of the two constituents (fiber and matrix) is usually known from data sheets. The density of the composite can be determined by using Archimedes' principle - provided the material is non-porous. Otherwise, the volume of the composite sample can also be determined very accurately by CT measurement and the mass can be weighed.

2.2.1.3 Computational non-destructive methods for FVC determination

There are multiple reasons why a non-destructive determination of the fiber volume content is desirable. Firstly, CT scans are used for every holistic characterization of FRP, as the fiber orientation, for example, cannot really be determined well locally in another way (apart from using the method of ellipses on micro sections [5]). The non-destructive determination of any additional

parameter from a 3D CT image makes the investigation more efficient instead of going to the further expense of pyrolysis/acid dissolution. Furthermore, in case of an experimental FVC determination, a round or rectangular sample between 10 mm and 30 mm is usually processed as a whole. This does not provide any information about the course of the fiber volume content along different axes. However, there is good reason to suspect that this is of particular interest with regard to the thickness of flat samples, e.g., to investigate the shell-core effect appearing in injection molding [5]. A low FVC on the outer surfaces of the specimen and thus a lower stiffness and strength could be decisive, especially for bending loads with maximum bending stress as tension or compression at the outermost edge of the cross-section. Theoretically, it is also possible to obtain 3D information on the FVC destructively by milling off thin layers and subsequent pyrolysis, which has already been done [5]. However, this procedure requires a lot of effort and measurement inaccuracies are particularly serious given the small amount of material and the potential fiber shortening due to the milling process. In addition, neither pyrolysis nor chemical dissolution are possible if natural fibers are involved, which are attacked by both.

The method of choice from classical image evaluation would be global thresholding. One makes use of the different attenuation coefficients of the fibers and the matrix, which cause them to appear with different gray values in the CT image. Hence, a gray value as threshold is sought which divides the histogram of all gray values in the image exactly at the point, where the material, that the voxels represent, changes. The location of these voxels is not taken into account in the case of thresholding procedures. One can simply search for a threshold manually in commercial software such as ImageJ by creating a threshold and visually estimating how many fibers are detected. However, there is also a large number of so-called automatic thresholds that are calculated based on analytical considerations, like the Otsu threshold [47], the moment-preserving method by Tsai [48] or the mean threshold [49–51]. They are pre-implemented both in ImageJ as well as in image processing packages available for Python or C++. The computation of the most known ones are explained in Section 2.3.4.2. You can then divide the number of pixels/voxels detected as fibers by the total

number and obtain a fiber volume content. The difficulty with this method is that the determined FVC depends directly on the threshold value. However, the true threshold cannot be determined even with high-resolution CT images with good contrast. Pinter [52] has applied a large number of different threshold techniques to GF-LFT and GF-SMC and compared them with TGA measurements, but cannot find one that fits perfectly and advises that these values should be treated with caution and that high contrast between fiber and matrix is required. Gandhi et al. [5] introduce a new µCT procedure that attempts to get rid of the ambiguity of selecting the true threshold. They chose the midpoint between the mean value that represents the glass fibers in the histogram and the mean value that represents the matrix as the threshold value. Although this does not represent the true threshold, small adjustments to higher or lower values only results in the FVC distribution being slightly shifted up or down, but its course remains the same. They then normalize the values of the distribution by dividing them by the average fiber concentration of the entire data set. This shifts the curves back to approximately the same values, regardless of which threshold was originally used. However, this approach only works if two clear peaks are recognizable in the histogram of the CT images. This is not the case for carbon fibers with a diameter of 5 µm - 7 µm, which show a low contrast to polymers in CT due to their similar densities. This challenge will be elaborated on in Section 2.3 about computed tomography.

After reviewing further literature, it becomes apparent that a simple threshold between fiber and matrix is clearly too imprecise [53–55]. They are still widely used in the case of pores and voids as air inclusions are usually easy to detect [56]. Even in that case though, deep learning (DL) solutions have outperformed thresholding in specific cases like small voids or low porosity [57]. Combinations of DL and thresholding have been explored for low contrast and noisy X-ray images as well [58]. For the specific problem of reliable determination of the FVC of carbon fiber reinforced polymers (CFRP) through image segmentation of CT images, DL presents a promising opportunity. Convolutional neural networks (CNN) in particular have proven to be able to handle low contrast images and improve statements resulting from them [59–64]. It seems especially

tempting in cases like these as necessary filtering and image processing steps do not have to be found by the engineer, notwithstanding that they are often needed for the training of the CNN in advance. Thus, in the Methods (Chapter 3) a CNN will be presented for this use case as well as an adapted thresholding method as comparison.

2.2.2 Fiber length distribution

The fiber length distribution (FLD) describes the amount of fibers as a function of their length in a sample volume. It plays an important role for the mechanical properties as described in Section 2.1, especially when including damage mechanisms [52]. Young's modulus, strength, and impact behavior all improve with rising fiber length with the effect saturating at very high lengths or even decreasing in the case of high fiber volume contents [5]. It can be given as a probability density function $f_l(l)$. Analogous to classical probability density functions, f_l is non-negative, integrable and normalized in such a way that the total integral takes the value one:

$$\int_0^\infty f_l(l)\mathrm{d}l = 1. \tag{2.36}$$

A two-parameter Weibull distribution is mostly used for modeling the fiber length distribution [65–68], which is defined as

$$f_l(L) = \frac{m}{n}\left(\frac{l}{n}\right)^{m-1} \exp\left[-\left(\frac{l}{n}\right)^m\right]; \quad l > 0, \tag{2.37}$$

with n and m as scale and shape parameters and l as the fiber length.

Alternatively, length histograms are used in practice, in which fibers inside of defined intervals of length are summed up.

2.2.2.1 Experimental destructive methods for FLD determination

There are several ways to determine the fiber length destructively, all of which are connected to the matrix removal, which is also used to determine the fiber volume content. Secondly, in most cases, a certain representative proportion of the fibers is selected first, as there are usually too many in a sample to be able to measure them meaningfully. Gandhi et al. [5] recommend a sample size of 1000 to 2000 fibers. After some kind of method for the dispersion of fibers is applied, the fibers are separated. The fibers can be dispersed either manually or by air on a scanner. As this causes difficulties, most current methods are based on dispersion in liquid and subsequent mixing either manually or using ultra-sound [46, 69], including the two commercial methods described briefly below. The fiber lengths could be determined subsequently, for example, by progressive filtering, a fast, mechanical but rather imprecise process. Direct optical methods are more common. Therefore, a digital image is created either through microscopy or an optical document scanner and the image is afterwards analyzed. The individual length of each fiber can be detected either manually by selecting respective endpoints or with the support of image processing algorithms. Hybrid approaches are also used. [5]

FASEP

With the commercially available system FASEP (IDM Systems, Darmstadt, Germany) [70] fiber lengths up to a theoretical length of 25 mm and also bent or crossed fibers of fiber reinforced thermoplastic materials can be detected. In a first step, as already mentioned, a sample is either pyrolysed or dissolved in acid in order to remove the matrix. After the fibers are separated, they must be dispersed. This is done by adding the remaining fibers to distilled water to which glycerine has been added. The mixture is gently stirred and multiple of these diluting steps are carried out by adding more water and glycerine [5]. A small amount of this mixture is then pipetted into a 90 mm diameter Petri dish.

A digital image of the fiber mixture is obtained using a customized dark-field imaging approach. The optical resolution of 1700 dpi corresponds to a pixel size of 14.9 µm. The tracking in the image itself is carried out using an ALF algorithm developed in-house by the FASEP company. [52, 69] Typically, several of these subsets of the sample are examined in succession to obtain a set of at least 3000 fibers and the results are combined into a histogram of the frequencies.

FibreShape FiVer

The sample preparation can be performed analogously to the FASEP approach. Distilled water and a little detergent are then added together with the fibers to a sample container measuring 6 cm × 12 cm. The fibers are distributed as evenly as possible with a spatula without breaking them. The sample vessel holder is then placed in the so-called FiVer scanner. An image is scanned using the Silverfast software. The real optical resolution corresponds to 8 µm, which is why fiber lengths between 30 µm and 5 cm can be detected. Depending on the inhomogeneity of the sample, five to seven of such images are taken per sample, which can correspond to over 10,000 elements after the measurement. In particular, the high resolution also makes it possible to measure fibers with a thinner diameter, such as carbon fibers. The FibreShape (IST AG, Vilters, Switzerland) program is then used to measure the corresponding lengths from the images. The exact length detection algorithm of the company is not known to the author. The output is the geodetic length, as the fibers that are measured are often curved. The geodetic length indicates the corresponding length that would be measured if the fiber was straight. The program calculates the value on the basis of the equality of the area or circumference of the bent and the curved fiber image. The procedure complies with the specifications of ISO 9276-1 and ISO 13322. [52]

This method is the one used for the determination of the fiber length distribution in this work, mainly because it was the technique available at FIBRE (Bremen), where the chemical matrix dissolution was performed before.

2.2.2.2 Computational non-destructive methods for FLD determination

Multiple algorithms have been developed for the non-destructive measurement of fiber lengths from μCT images. To the author's knowledge all of them have been developed for the example of glass fiber reinforced polymers.

A method for the measurement of the length of short fiber reinforced polymers has been introduced by Salaberger et al. [71] in 2011, where they present two concepts. The first procedure works by accurately detecting a starting point and an end point of every fiber. Therefore, an anisotropic diffusion filter was applied first to reduce noise without blurring any edges. The image is then binarized by applying the Otsu threshold in order to isolate the fibers from the matrix. In order reduce the information per fiber to their medial axis and therefore only a single line of voxels, a binary thinning is conducted. This process is also called skeletonization [72]. In the following, a cluster analysis is performed, which has the purpose of dividing touching fibers, i.e. regions with clustered fiber-associated voxels, into individual objects. This method was introduced by Pfeifer [73]. Clusters are defined as a voxel that has more than two neighbors. They are recognized morphologically. The cluster information is then used for the fiber tracing, where the fibers are followed to their endpoint or until a cluster is reached. If detected segments fall below a user-defined length, they are assigned to image noise and are not evaluated as fibers. Subsequently, all detected segments are analyzed and identified as fibers or merged into fibers in the event of a cluster encounter. The joining decision is made by analyzing the angles of all segments in a cluster. Salaberger et al. chose a kink angle of 160° and a cluster distance of 14 μm, which corresponds to the approximate fiber

diameter as parameters of this algorithm. Subsequently, when all fiber paths are found, the length determination itself can be performed.

Since straight fibers can be assumed in the case of short fibers, the length of the ith fiber is simply determined by the Euclidean distance of the start point x_{i1}, y_{i1}, z_{i1} and end point x_{i2}, y_{i2}, z_{i2} of the fiber as

$$l_i = \sqrt{(x_{i2} - x_{i1})^2 + (y_{i2} - y_{i1})^2 + (z_{i2} - z_{i1})^2}. \qquad [13] \qquad (2.38)$$

As the second concept presented by Salaberger et al. [71] is similar to the approach by Teßmann et al. [74] presented in the next paragraph, it will not be described in detail in this work. It works directly on the gray value image without binarization and requires the detour via the determination of the fiber orientation by calculating the Hessian matrix and analyzing their eigenvalues. The Hessian matrix is a measure of the local curvature of a 2D/3D image. While originally being used in order to detect valleys and ridges in topology problems [52], Daniels et al. [75] first implemented a 3D Hessian matrix for the evaluation of local orientations in volumetric images in 2006. Mathematically it is defined as the square matrix of the second order partial derivative of the image $I(x)$:

$$\boldsymbol{H}(\boldsymbol{x}) = \nabla^2 I(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial x \partial z} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} & \frac{\partial^2 I}{\partial y \partial z} \\ \frac{\partial^2 I}{\partial z \partial x} & \frac{\partial^2 I}{\partial z \partial y} & \frac{\partial^2 I}{\partial z^2} \end{bmatrix}. \qquad (2.39)$$

In addition to the local orientation, the gray value distribution across the cross section of a fiber is taken into account. It should roughly match a Gaussian distribution perpendicular to the axial direction and hence show the highest gray value in the center of the fiber. Thereby, one can reduce the fibers to their center voxels. This dataset can then be examined analogously to the

approach before to detect the start and end point of the fibers and find the length. They attain up to 97 % accuracy in another paper investigating the reproducability and accuracy of the described approach for short glass fiber reinforced polypropylene [76]. However, mean fiber lengths decreased for lower resolutions. Since the curvature of the fibers, which generally occurs in long-fiber reinforced polymers, is not taken into account by these approaches, two further methods are described below.

Teßmann et al. [74] introduced an alternative approach for the fiber length determination from µCT images of long fiber reinforced thermoplastic specimens in 2010. This approach is based on the reasonable assumptions that (a) the fibers have a cylindrical shape, (b) the gray value profile of the fibers is the highest at their centers as mentioned in the paragraph before and (c) that all fibers have a constant and previously known diameter. Subsequently, the segmentation is divided into several sub-steps. The image is filtered and reduced by a closing operation first. This step is necessary as standard segmentations such as seeded region growing [77] fail in case of tightly packed fibers. A morphological erosion filter is therefore applied to the image, in order to enhance the fiber borders. Subsequently, the center voxels of the fibers have to be determined through a discrimination function based on an eigenvalue analysis, which was introduced by Frangi et al. [78]. The already mentioned Hessian matrix is computed for every voxel for this step. The thereby detected center points are used as starting seeds for the tracing algorithm. Therefore, a model-based cylinder approximation scheme is used which is based on the fiber shape assumption mentioned before. A starting point candidate is followed along the minimum eigenvector in both directions resulting in a center point list, from which the circular shape can be built. Once a fiber is completely segmented, it is removed from the original image and the procedure is repeated until there are no more prospective fibers in the image. Finally, the fiber lengths can then be determined easily as

$$L_i = ns, \tag{2.40}$$

with n denoting the number of center point voxels of one fiber and s the isotropic voxel size. In the paper by Teßmann et al. [74], certain special cases of crossing fibers or partial fiber segmentation are explained in more detail. Finally, the algorithm was tested on artificially generated test images and images of selected real components.

Finally, the FLD determination approach by Pinter et al. [13] from 2016 is presented here. They introduced a reliable procedure to determine FLDs of LFT material which - contrary to Teßmann's method - shall even work for moderate image qualities. It starts with a pre-processing of the image with the help of a CircularVoting filter developed by Bertram and Pinter [79] and available as open source software on SourceForge as part of the so-called Composight package. Through this pre-filter, connections of almost touching fibers are avoided by thinning them and thus increasing their distances. For the exact details of the CircularVoting method, the reader is referred to the cited paper. However, the idea is based on the combination of a coherence measure with a surface normal overlap measure. The latter uses only the fixed fiber radius of the glass fibers. The former makes use of the eigenvalues of the structure tensor of the image. The structure tensor is also used for determining the fiber orientation tensors in this work and will therefore be introduced here. Krause et al. [80] were to the author's knowledge one of the first who used the structure tensor approach to acquire orientation information from CT images of FRP in 2010. It is defined as follows:

$$\begin{aligned}\mathbf{S} =& \nabla I_\sigma(\boldsymbol{x}) \otimes \nabla I_\sigma(\boldsymbol{x}) = \nabla I_\sigma(\boldsymbol{x}) \nabla I_\sigma(\boldsymbol{x})^\intercal = \\ & \begin{bmatrix} \left(\frac{\partial I_\sigma}{\partial x}\right)^2 & \frac{\partial I_\sigma}{\partial x}\frac{\partial I_\sigma}{\partial y} & \frac{\partial I_\sigma}{\partial x}\frac{\partial I_\sigma}{\partial z} \\ \frac{\partial I_\sigma}{\partial y}\frac{\partial I_\sigma}{\partial x} & \left(\frac{\partial I_\sigma}{\partial y}\right)^2 & \frac{\partial I_\sigma}{\partial y}\frac{\partial I_\sigma}{\partial z} \\ \frac{\partial I_\sigma}{\partial z}\frac{\partial I_\sigma}{\partial x} & \frac{\partial I_\sigma}{\partial z}\frac{\partial I_\sigma}{\partial y} & \left(\frac{\partial I_\sigma}{\partial z}\right)^2 \end{bmatrix}.\end{aligned} \tag{2.41}$$

Working on the first numerical derivative of an image combined with a fixed "window function", in this case a Gaussian blur of width σ used for smoothing, it is a robust approach [80–82].

After the CircularVoting pre-filter is applied, a user-defined threshold is applied on the image in order to binarize it [52]. Then, the fibers are reduced to their medial axis through skeletonization [72]. The fibers are tracked with the help of a modified version of the ImageJ plugin Skeletonize (2D/3D) and AnalyzeSkeleton by Ignacio Arganda-Carrera [83, 84]. The original code was not intended for fibers that do not show any junctions. It was therefore modified by Pinter et al. [13, 52] taking the local orientations into account in order to remove cross-over links in the resulting graphs and connect branches that belong to the same fiber. For more details on the method, the readers are referred to the sources [13, 52].

While the first method by Salaberger et al. [71] only works for short-fiber reinforced polymers, the latter two by Teßmann et al. [74] and Pinter et al. [13] can also handle scans of long-fiber reinforced polymers. However, both emphasize the need for high resolution, even in Pinter's work, and both only work with glass fibers. However, a high resolution of e.g. 5 µm for glass fibers may correspond to three or even four voxels across the diameter. This would be equivalent to less than 2 µm resolution for carbon fibers. The frequent dichotomy in this work between the extremely high resolution required and the poor image quality, high noise and low contrast meant that this goal could not be achieved. Accordingly, these algorithms were not used for the material in this work and only destructive methods were used to determine the FLD.

2.2.3 Fiber orientation distribution

Most fibers in short and long fiber reinforced polymers do not have a uniform orientation. A necessary description of the orientation distribution can be given by a probability distribution function of the orientation, ψ, [43], sometimes called orientation distribution function (ODF) [85] or fiber orientation density

distribution function [86], which all describe the same property. First of all, it is assumed that the orientation of a fiber in any coordinate system can be described by a unit vector $\boldsymbol{p} \in \mathcal{S}$, where $\mathcal{S} = \{\boldsymbol{p} \in \mathbb{R}^3 \mid \|\boldsymbol{p}\| = 1\}$. $\mathcal{S}$ denotes the two-dimensional unit sphere as the set of all possible directions of $\boldsymbol{p}$. The components of $\boldsymbol{p}$ can be determined using the angles that the fiber makes with the axes of the coordinate system (cf. Figure 2.6) using trigonometric functions as follows:

$$\begin{aligned} p_1 &= \sin\,\theta\,\cos\,\phi \\ p_2 &= \sin\,\theta\,\sin\,\phi \\ p_3 &= \cos\,\theta. \end{aligned} \tag{2.42}$$

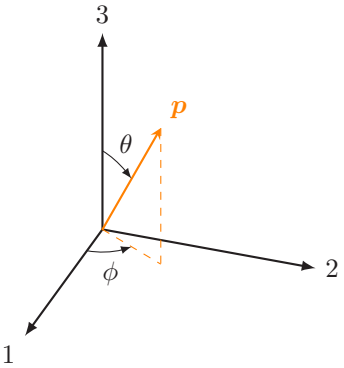


Figure 2.6: Coordinate system and fiber with angles to the coordinate axes. Based on a TikZ library [87] but changed for the case at hand.

The length of the vector is fixed, fulfilling

$$p_i p_i = 1. \tag{2.43}$$

The fiber orientation distribution function (FODF) $\psi : \mathcal{S} \rightarrow \mathcal{P}$ describes the probability of finding a fiber in direction $\boldsymbol{p}$ with $\mathcal{P} = \{\psi \in \mathbb{R} \mid 0 \leq \psi \leq 1\}$ [43]. ψ is defined in a way that the probability of finding a fiber between angles θ_1 and $(\theta_1 + d\theta)$ as well as between ϕ_1 and $(\phi_1 + d\phi)$ is given by

$$\mathcal{P}(\theta_1 \leq \theta \leq \theta_1 + d\theta,\, \phi_1 \leq \phi \leq \phi_1 + d\phi) = \psi(\theta_1, \phi_1) \sin\theta_1 \, d\theta \, d\phi \text{ [43].} \tag{2.44}$$

In practice, the distribution of orientations can be given in the form of a histogram as a discrete approximation to the underlying probability density function for orientation [88] (as can be seen in Figure 4.33 on the right side).

The function $\psi(\boldsymbol{p})$ has to fulfill multiple mathematical conditions. First of all, fibers have an orientation but do not have a direction [85] meaning a fiber with angles θ, ϕ is identical to one with angles $\pi - \theta, \phi + \pi$ to the function [43]. In other words, the beginning and end of a fiber cannot be distinguished [86]. Hence, $\psi(\boldsymbol{p})$ is periodic and therefore a symmetric function

$$\psi(\boldsymbol{p}) = \psi(-\boldsymbol{p}). \tag{2.45}$$

The function $\psi(\boldsymbol{p})$ is also non-negative [85]:

$$\psi(\boldsymbol{p}) \geq 0. \tag{2.46}$$

Secondly, $\psi(\boldsymbol{p})$ has to be normalized as every fiber has an orientation [43], i.e.,

$$\int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \psi(\theta, \phi) \sin\theta \; d\theta d\phi = \oint \psi(\boldsymbol{p}) dp = \int_{\mathcal{S}} \psi(\boldsymbol{p}) dp = 1, \tag{2.47}$$

using the integral over the surface of the unit sphere and consequently over all possible directions of $\boldsymbol{p}$ [43].

Lastly, $\psi(\boldsymbol{p})$ fulfills the continuity condition [86]

$$\dot{\psi} = -\text{grad}_{\mathcal{S}}(\psi(\boldsymbol{p})\dot{\boldsymbol{p}}). \tag{2.48}$$

This equation describes the change of the fiber orientation distribution function with time when the fibers move with the bulk motion of the fluid. $\psi(\boldsymbol{p})$ can therefore be determined when an appropriate expression describing the average fiber angular velocities $\dot{\theta}, \dot{\phi}$ or the average fiber direction velocity $\dot{\boldsymbol{p}}$ is provided [43]. This last condition is particularly crucial in the context of modeling and simulating a flow process, which will not be discussed in detail in this paper.

While the fiber orientation distribution function is a closed, explicit [25], complete and unambiguous [43] description of the fiber orientation state, it can become arbitrarily complex [86]. Hence, Kanatani [89] as well as Advani and Tucker [43] suggested the use of moments of the function, which are called fiber orientation tensors. Inspired by the Fourier series expansion, they offer improved storage possibilities, reduced complexity and the application of algebraic operations [85].

2.2.3.1 Fiber orientation tensors (FOT)

While Kanatani [89] distinguishes between orientation tensors of first and third kind, only the tensors of first kind that are also discussed in the work of Advani and Tucker [43] will be discussed here.

Orientation tensors as statistical moments of the distribution function are defined as the integral of the dyadic product(s) of the vector $\boldsymbol{p}$. As the odd-order integrals amount to zero due to the symmetry of $\psi(\boldsymbol{p})$, there are only even-order tensors, with the second- and fourth-order tensors being the most commonly used ones [43, 85, 86]:

$$
\begin{aligned}
\boldsymbol{A} &= \int_{\mathcal{S}} \psi(\boldsymbol{p})\, \boldsymbol{p} \otimes \boldsymbol{p}\ \mathrm{d}p, \\
A_{ij} &= \oint \psi(\boldsymbol{p})\, p_i p_j\ \mathrm{d}p, \\
\mathbb{A} &= \int_{\mathcal{S}} \psi(\boldsymbol{p})\, \boldsymbol{p} \otimes \boldsymbol{p} \otimes \boldsymbol{p} \otimes \boldsymbol{p}\ \mathrm{d}p, \\
A_{ijkl} &= \oint \psi(\boldsymbol{p})\, p_i p_j p_k p_l\ \mathrm{d}p.
\end{aligned}
\tag{2.49}
$$

In the index notation, the base $\boldsymbol{e}_i$ has been omitted respectively. In practice, after the orientation of N fibers is measured in a region of interest, e.g., through image-processing of µCT images, the orientation tensors of second- and fourth-order can be calculated via

$$
\begin{aligned}
\boldsymbol{A} &= \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{p} \otimes \boldsymbol{p})_k\ , \\
A_{ij} &= \frac{1}{N} \sum_{i=1}^{N} (p_i p_j)_k\ , \\
\mathbb{A} &= \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{p} \otimes \boldsymbol{p} \otimes \boldsymbol{p} \otimes \boldsymbol{p})_k\ , \\
A_{ijkl} &= \frac{1}{N} \sum_{i=1}^{N} (p_i p_j p_k p_l)_k\ \text{[88, 89]}.
\end{aligned}
\tag{2.50}
$$

A set of measured FOT of second order will later be referred to in caligraphic letters, such as $\mathcal{T}_{\mathrm{m}}$.

Orientation tensors are fully symmetric [43], hence

$$\begin{aligned} a_{ij} &= a_{ji} \quad \text{or} \quad \boldsymbol{A} = \boldsymbol{A}^T, \\ a_{ijkl} &= a_{jikl} = a_{kijl} = a_{lijk} = a_{klij}, \quad \text{etc.} \end{aligned} \tag{2.51}$$

holds. The symmetry reduces the amount of necessary tensor components to uniquely describe an orientation tensor of second order to only six (a_{11}, a_{22}, a_{33}, a_{12}, a_{13}, a_{23}) instead of nine. As a consequence of the normalization condition in Equation (2.47), the trace of an FOT of second order has to be one:

$$\boldsymbol{A} \cdot \boldsymbol{I} = \mathrm{tr}(\boldsymbol{A}) = 1, \tag{2.52}$$

with the identity tensor $\boldsymbol{I}$. This relationship further reduces the necessary tensor components to only five [90, 91].

All higher order tensors contain the respective tensors of lower order [85], leading to the following contraction [86]

$$\mathbb{A} : \boldsymbol{I} = \boldsymbol{A}. \tag{2.53}$$

In contrast, determining a higher order tensor from a lower order tensor is not trivial and requires some kind of closure approximation, which will be shortly taken up again in the next paragraph of this section.

The distribution function $\psi(\boldsymbol{p})$ could be reconstructed exactly by means of Fourier analysis if all FOT up to order $n = \infty$ were available [92]. In all other cases, it can be approximately recovered from an FOT, as shown exemplary in the following equation for a second order tensor:

$$\psi(\boldsymbol{p}) \approx \frac{1}{4\pi} + \frac{15}{8\pi} \boldsymbol{B}\boldsymbol{F}(\boldsymbol{p}), \tag{2.54}$$

with $\boldsymbol{B} = \boldsymbol{A} - \frac{1}{3}\boldsymbol{I}$ and $\boldsymbol{F}(\boldsymbol{p}) = \boldsymbol{p} \otimes \boldsymbol{p} - \frac{1}{3}\boldsymbol{I}$ [43].

This expression of the distribution function through FOT as 3D tensorial Fourier series, is also called spherical harmonic expansion [89, 93]. It is cut in Equation (2.54) after the first two terms. For the respective relationship for higher order tensors, the reader is referred to the article by Advani and Tucker [43] and to the one by Jack et al. [94]. It shall also be noted that orientation tensors are a truncation of the full series partly shown in Equation (2.54) and higher order tensors represent the distribution function more accurately. However, second- and fourth-order tensors are sufficient in practice and state of the art. [43]

Closure approximations

A closure approximation $\mathcal{C}$ is a mapping $\mathbb{A} = \mathcal{C}(\boldsymbol{A})$ [92]. The need for closure approximations descends from process simulations of fiber suspensions. The orientation evolution can be described by approaches like Jeffery's model [95] or the model by Folgar and Tucker [96] who included the FODF and added a diffusion term to the former equation [86]. When expressing the aforementioned models in terms of FOT, the fourth-order tensor $\mathbb{A}$ is required in order to solve the transport problem.

Many different closure approximations have developed over time with varying complexity and accuracy. The quadratic closure can be considered the simplest analytical closure and is given by

$$\mathbb{A}^{Q} = \boldsymbol{A} \otimes \boldsymbol{A} \tag{2.55}$$

and is only exact for a unidirectional FRP [43, 86, 97]. Contrarily, the linear closure [43, 98] gives correct results for isotropic fiber orientations [86]. Then, there are hybrid closures combining these ideas [99, 100] and more advanced orthotropic fitted closures [101, 102] working with fitting of eigenvalues. The widely used invariant-based optimal fitting (IBOF) closure [103, 104] is also a physically more consistent approach [92] and was used, e.g., by Meyer

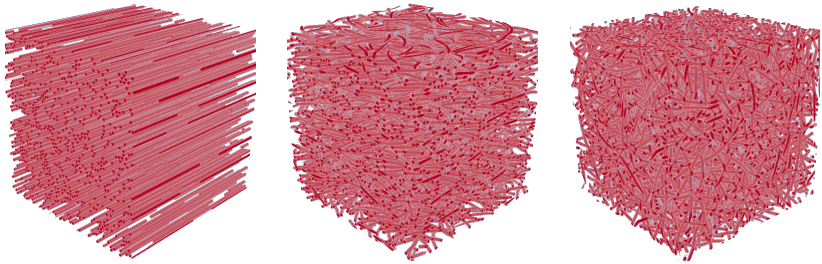
et al. [34] in their simulation of the compression molding process of Sheet Molding Compounds or by Brylka [90] for linear elastic homogenization of polypropylene LFT with Mori-Tanaka as a good compromise between accuracy and computational cost [86]. For a more comprehensive analysis and discussion of the various closure algorithms, the reader is referred to the work of Breuer et al. [105].

It should be mentioned at this point that the use of closures inevitably introduces errors. For example, Brylka calculates maximum errors in the resulting stiffness of 7 % on the sample scale and 15 % in the individual layers across the thickness by using second-order FOT and an IBOF closure in his Mori-Tanaka approach compared to the direct use of fourth-order FOT [90]. Furthermore, it should be noted that there is also the possibility of using indirect closure methods [93, 106], which take the detour of reconstructing the FODF ψ from the second-order tensor, which can then be used to determine the fourth-order tensor [92]. However, these are based on assumptions too that can vary depending on the approach. Shannon's maximum entropy method (MEP) [107] is a well-known example.

Examples of FOT and correlation with concrete fiber distributions in FRP

In order to be able to refer an orientation tensor to an actual fiber distribution, visible, e.g., in a CT scan, three simple FOT of a unidirectional, approximately planar and isotropic state are presented in Figure 2.7 alongside a respective FRP microstructure portraying the orientation state of the FOT.

The microstructures were generated with the "fSAM" algorithm by Lauff et al. [108]. An almost planar isotropic fiber orientation state is considered as generating microstructures with eigenvalues closer to $a_{33} = 0$ is computationally extremely expensive. Further fiber packing algorithms will be mentioned in this work in Section 2.2.4.



$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0.49 & 0 & 0 \\ 0 & 0.49 & 0 \\ 0 & 0 & 0.02 \end{bmatrix} \qquad \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$$

Unidirectional | Almost planar isotropic | Isotropic

Figure 2.7: Examples of different orientation states in generated microstructures of specific fiber distributions and respective orientation tensors.

Parametrization of FOT

Any FOT of second order $\boldsymbol{A}$ is symmetric (see Equation (2.51)), and positive semi-definite as all of its eigenvalues are non-negative, $\lambda_i \geq 0$ (cf. Equation (2.46)). Consequently, it can be rotated in a way that only the diagonal of the matrix is populated, hence it can be diagonalized. This so-called eigendecomposition of a matrix is defined as such

$$\boldsymbol{A} = \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^T = \boldsymbol{R} \star \boldsymbol{\Lambda}, \tag{2.56}$$

with $\boldsymbol{R} \in SO(3)$ being an orthogonal matrix ($\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$ holds), whose i-th column is the eigenvector $\boldsymbol{v}_i$ of $\boldsymbol{A}$ (the three eigenvectors are orthonormal to each other) and is sometimes called rotation matrix for aforementioned reasons. $\boldsymbol{\Lambda}$ is the diagonal matrix with the corresponding eigenvalues $\boldsymbol{\Lambda} = \boldsymbol{I}\boldsymbol{\lambda}$ with $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^\intercal$. It exists a common ordering convention of the eigenvalues [85, 91, 109–111] following

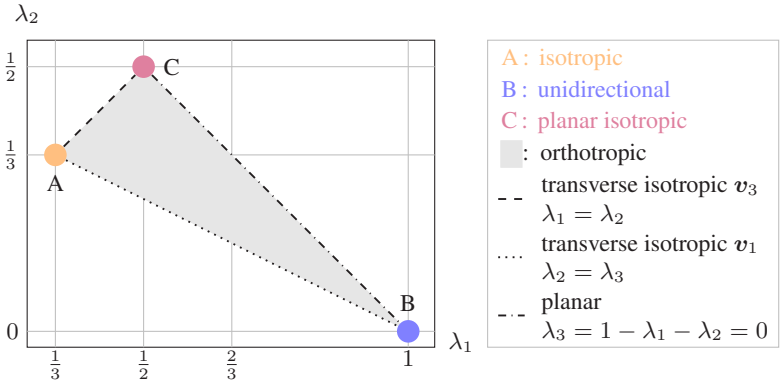


Figure 2.8: Material symmetries and constraints defining the orientation triangle. Based on [85].

$$\lambda_1 \geq \lambda_2 \geq \lambda_3. \tag{2.57}$$

Equation (2.52) translates to

$$\lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{2.58}$$

This parametrization of orientation tensors allows the depiction of the variety of FOT via pairs of λ_1 and λ_2 (alongside a mapping $\boldsymbol{R}$ defining the orientation coordinate system) in a two-dimensional space [85]. This representation is also known as orientation triangle, which can be seen in Figure 2.8 and also, e.g., in the works of Bauer and Böhlke [85], Gajek et al. [91], Goldberg et al. [109], Cintra et al. [101], Chung et al. [103] or Köbler et al. [110].

It simplifies the assignment of different material symmetries to the specific FOT. The weakest material symmetry of orthotropy as the gray area in Figure 2.8 is thereby defined by three points of extremal orientation states of isotropy (A), unidirectionality (B) and planar isotropy (C). The connection lines between these three points describe special material symmetries themselves, with the connection between A and B depicting states of transverse isotropy regarding

the principle axis of $\boldsymbol{v}_1$, the connection between A and C portraying transverse isotropy regarding $\boldsymbol{v}_3$ and the connection between C and B represents a planar state of fiber orientation. The reader is referred to Bauer and Böhlke's comprehensive work on the variety of fiber orientation tensors [85] for more details.

Visualization of fiber orientation tensors through tensor glyphs

Especially for the FOT interpolation described in Section 2.2.3.3, a simple and descriptive but comprehensive way to visualize fiber orientation tensors and their anisotropy was necessary. Therefore, the possibility to portray symmetric positive definite (SPD) tensors by glyphs (often ellipsoids) was used, which is a well-known concept in scientific visualization and could be found mostly for diffusion tensors [112–114] in the beginning. It is especially helpful for tensor field visualization, where a tensor is available at different points in a grid. The glyphs (sometimes also called icons) portrayed at each grid point are a symbolic parametric object to visually represent the features of the tensor it portrays [115]. Hence, the attributes of the tensors somehow have to be mapped to the parameters (or degrees of freedom) of the glyph [115]. So as described by Kindlmann in [116], one takes an initial glyph geometry G (in our case the ellipsoid geometry) and translates it into a tensor glyph $G_{\boldsymbol{A}}$ as following

$$G_{\boldsymbol{A}} = \boldsymbol{R}\boldsymbol{\Lambda}G, \tag{2.59}$$

and then plots $G_{\boldsymbol{A}}$ at the specific grid location of the given orientation tensor $\boldsymbol{A}$. It goes back to the spectral decomposition of an SPD tensor described in Equation (2.56), but by specifically not rotating back, the orientation of the tensor can be depicted. The shape of the ellipsoid is representative of the eigenvalues and concomitant with the tensor's anisotropy and the orientation of the ellipsoid is representative of the eigenvectors, respectively. More specifically,

the three eigenvalues determine the radii of the ellipsoid in the three main spatial directions. The center of the ellipsoid is set to the tensor's position. Then, the coordinates in the variables x, y and z are rotated with the rotation matrix (eigenvectors) to represent the orientation of the tensor.

Kindlmann further developed superquadric tensor glyphs in 2004 [116] to battle problems of asymmetry and ambiguity, which were further generalized in 2010 [117] and even used for fiber orientation tensors as well [92]. However, the standard glyph visualization was sufficient in this work.

The glyphs offered a way to interpret the success of different interpolation methods by visualizing the evolution of orientation tensors across FOT fields. The rendering in this work was done with MATLAB R2020b with the help of the "plotDTI" function of the fanDTasia ToolBox by Barmpoutis et al. [118].

In Figure 2.9, some examples for special tensor glyphs rendered with the function above can be seen. Multiple fields of realistic tensor glyphs follow in the course of this work.

2.2.3.2 Computational methods for determination of fiber orientation tensors from µCT scans

In order to determine the fiber orientation from µCT scans, two generally different approaches are conceivable. As already described in Section 2.2.2.2 on computational determination of fiber lengths, one can track the center line of each fiber to get a vector for every fiber orientation, as done in [76]. However, this method requires high resolution and sufficient contrast of the images and both aspects are lacking in our case and carbon fiber reinforced polymers in general. Luckily, the orientation of the fibers can - in contrast to their length - also be determined on a voxel basis not including their connectivity [81]. Pinter presented an evaluation of the accuracy of three methods of this kind in [81] (also in [52, 119]). The first method was introduced by Lampert and Wirjadi and is based on an anisotropic Gaussian filter [120]. This filter returns higher

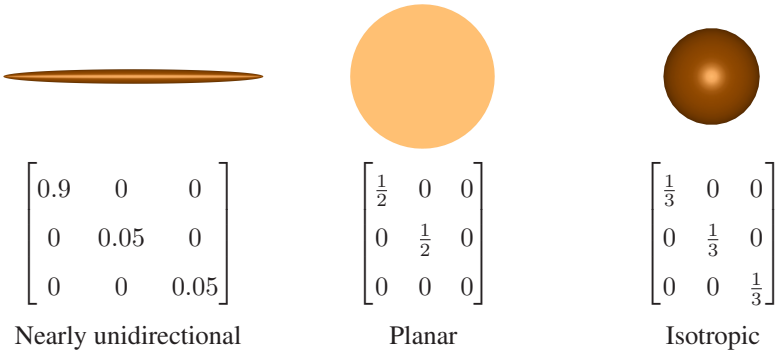


Figure 2.9: Examples of tensor glyphs rendered with the "plotDTI" function of Barmpoutis et al. [118] in Matlab R2020b. At the left, a nearly unidirectional state can be seen, as a completely unidirectional case is impossible to visualize as an ellipsoid (see section above). The high anisotropy can be identified by the slim ellipsoid body and the strong orientation in the x-direction (horizontally to the right) is visible. This is followed by a completely planar state, which is visualized as a flat plate. The isotropy in the visible plane can be seen due to its complete roundness and the missing shading implies the missing expansion in the z-direction (normal to the visible plane). Lastly, the isotropic case leads to a perfect sphere with equal dimensions in all coordinate directions. As for the planar case, no preferred orientation can be perceived.

values if the fiber in the currently examined voxel aligns better to an orientation from a table of given orientations and vice versa. The second concept is based on the already introduced Hessian matrix (compare Equation (2.39)) proposed by Daniels et al. [75]. Lastly, the also already introduced structure tensor (cf. Equation (2.41)) was tested. All of the three implemented methods were published in the open source project Composight by Bertram and Pinter [79]. As the last method performed the best for artificial images and real images of glass fiber reinforced polymers and Pinter et al. [81] showed that it was even the only one working for low contrast pictures with carbon fiber reinforced polymers, the structure tensor approach as implemented by Bertram and Pinter [79] was used in this work and it is therefore the only one described in further detail. A small adjustment made later is explained in the Methods (cf. Section 3.6).

Described by van Ginkel [121] as a way to analyze orientation in 3D images, Krause et al. [80] used a structure tensor based approach first for orientation analysis of FRP in 2010. The structure tensor $\boldsymbol{S}$ is composed of the dyadic product of the so-called Gaussian derivative $\nabla I_\sigma(\boldsymbol{x})$ [80] defined as

$$\nabla I_\sigma(\boldsymbol{x}) = \nabla(\boldsymbol{K}_\sigma \star \boldsymbol{I})(\boldsymbol{x}), \tag{2.60}$$

which can already be seen in Equation (2.41). It consists of the partial derivatives of the function I combined with a Gaussian blur $\boldsymbol{K}_\sigma$ in a fixed window with width of σ. The Gaussian derivative is a regularization, as the computation of gradients without it is ill-posed [80]. The application of a Gaussian blur is the same as convolving the image with a Gaussian function and will be revisited in Section 2.3.4.1 on filters. Subsequently, both Krause and Pinter use an additional, not mandatory Gaussian smoothing $\boldsymbol{K}_{\rho_1}$ with the standard deviation ρ_1 as additional regularization leading to the final structure tensor $\boldsymbol{J}_{\rho_1}$:

$$\boldsymbol{J}_{\rho_1}(\boldsymbol{x}) = \boldsymbol{K}_{\rho_1} \star (\nabla I_\sigma \nabla I_\sigma(\boldsymbol{x})^T)(\boldsymbol{x}). \tag{2.61}$$

Pinter et al. [81] point out that the second blurring parameter ρ_1 has to be larger than the first one, σ, that is used for the derivative. The resulting structure tensors are flat ellipsoids in the case of fiber-like structures that are aligned perpendicular to the local fiber orientation, see Figure 2.10. The fiber orientation vector is finally calculated by evaluating the smallest eigenvalue of the structure tensor and its corresponding eigenvector.

The resulting fiber orientation vectors are further processed in MATLAB to orientation tensors, pseudocolor images, etc., which will be explained in the Methods (cf. Section 3.6).

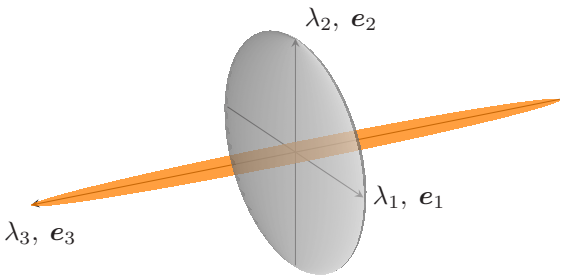


Figure 2.10: The flat structure tensor (gray) of a fiber-like neighborhood (orange) that is parallel to vector $\boldsymbol{e}_3$. The gradient directions are hence spread out but perpendicular to $\boldsymbol{e}_3$ resulting in a pancake-like shape of the structure tensor with $\lambda_1 \approx \lambda_2 \gg \lambda_3$.

2.2.3.3 Necessity of fiber orientation tensor interpolation

The fiber orientation in a sample of FRP material can accordingly be determined by a CT scan and the subsequent application of the method presented above. The question of the size of the sample volume is not insignificant here. This question is discussed in more detail in Section 3.1.6. However, it is trivial that a fiber orientation tensor of a rather small sample may represent very locally occurring fiber orientation behavior (e.g. a strong anisotropic orientation of a single fiber bundle). In contrast, a very large sample would on the one hand make it impossible for the algorithm to detect small fiber bundles because the resolution would then be too poor. On the other hand, a fiber orientation tensor obtained in this way would mean a massive averaging of a large range of different fiber orientations, which would probably result in very isotropic FOT that are not representative.

In this work, specimen sizes with an edge length between 10 mm and 30 mm were used depending on the application (see Section 3.1.6) at a plate size of 400 mm × 400 mm × 3 mm. It is immediately clear that a specific orientation tensor of such a sample represents only a fraction of the entire fiber orientation behavior of the pressed plate (cf. Figure 2.11).

For stiffness modeling, however, a representative averaged FOT of such a plate is often required, which cannot be obtained in this way. Specimens for tensile

Figure 2.11: Visualization of the interpolation problem: Nine measured tensors of 10 mm × 10 mm × 3 mm specimens are shown as blue tensor glyphs at the corresponding extraction points. A transparent photo of an original plate with the dimensions 400 mm × 400 mm × 3 mm is shown in the background. The behavior of the fiber orientation between the small measured areas is unknown. Based on Blarr et al. [17].

tests, where one would like to compare the discovered stiffnesses and strengths in different directions with measured fiber orientations, are by standard also larger than the CT specimens. In addition, simulations of the pressing process usually provide an FOT at each mesh point, which is difficult to compare with only a few FOTs determined at specific points. However, cutting and scanning the entire plate and then carrying out a fiber orientation analysis would be immensely time-consuming. With an average scan length of two hours and approximately another hour of work in post-processing, this seems impossible to implement in practice. Therefore, the scientific question of determining tensors at various grid points from a few given tensors at specific grid points arose. This corresponds to the mathematical problem of tensor interpolation. While the interpolation of scalar values has been widely studied, the interpolation of tensors is less well-known.

Current state of research on tensor interpolation [1]

The problem of tensor interpolation can be theoretically avoided for fiber orientation tensors, compared to, for example, stress and strain tensors, by switching to the scalar distribution function. It must be mentioned, that this switching is not unique, as the second order FOT contains too little information. Nevertheless, the function can be recovered under assumptions and a subdivision into discrete directions and subsequent interpolation would equal a Euclidean interpolation of the scalar-valued function $\bar{\Psi}$. In fact, this corresponds exactly to averaging the tensor components (with weights depending on distance). This can be explained by the fact that an integral is a linear mapping. However, this standard method led to a kind of "artificial" isotropy, which is shown by a change of shape in the manner of a rounding in the representation form of tensor glyphs. This does not necessarily seem to represent a useful averaging, as this issue of so-called tensor swelling arose both in the field of medical technology when interpolating diffusion tensors from MRI images [122–124] and in the case of FOT mapping in process simulation applications [125]. This swelling effect is due to non-monotonic interpolation of the tensor determinant and the Euclidean method does not preserve the positive definiteness. It was part of scientific discussions whether a more isotropic tensor as an interpolation between two anisotropic tensors that point in different directions still constitutes a reasonable behavior of fibers. It would imply that in a region of changing flow direction first some fibers turn and others stay in the original direction until most are turned in the end. The opposite idea that most turn first a bit and later completely into the new direction would rather be indicative of tensors in the center to be not significantly less anisotropic than the two next to it. As a reference solution, the first method for tensor interpolation implemented and used in this thesis is this Euclidean interpolation, which will be called component averaging (CA) from now on.

[1] This section is extracted from the author's publication [17] with only slight linguistic changes.

The complex Riemannian interpolation is another "global" interpolation method [126]. However, if more than two input arguments are used, the underlying computations can only be solved implicitly, which requires an iterative and therefore computationally expensive calculation. Since this thesis explicitly seeks an application-oriented method that is as fast and simple as possible while maintaining the highest possible quality, Riemannian interpolation will not be discussed further. Another logarithmic, but explicitly solvable approach was introduced by Arsigny et al. [122]: the Log-Euclidean tensor interpolation method. This method was already considered for FOT by Krauß and Kärger [125]. As it showed similar behavior in [125] as the basic Euclidean interpolation (component averaging), it will not be used in this work.

A completely different approach are so-called decomposition methods. These methods make use of the fact that SPD tensors can be decomposed into eigenvalues and eigenvectors in spectral decomposition (cf. Equation (2.56)). In terms of tensor glyphs, the eigenvalues are responsible for the shape, while the eigenvectors are responsible for the orientation of the tensor in space. Thus, shape and orientation can be weighted (according to various possible distance measures) and interpolated separately and then recomposed into a tensor. This can be done directly via the eigenvalues and eigenvectors, or via detours with the help of other invariants and, for example, quaternions. This method allows in particular the resolution of the swelling effect [123, 125]. It should be mentioned here that there is not only one decomposition method, but this must be understood as a kind of umbrella term, which can be executed very differently in the individual steps. The only previous use of this concept for FOT by Krauß and Kärger [125] differs from the one implemented in this work, for example. The exact concepts used for the decomposition method will be explained in detail in the Methods chapter (cf. Section 3.7.3) and have been previously summarized by the authors in [17, 127], but basic considerations about the decomposition method will be elaborated in the following paragraph.

As far as the shape is concerned, a direct interpolation of the eigenvalues would be conceivable. There are also approaches which handle it this way [128]. But

Ennis et al. [124] developed the concept of orthogonal invariants, which seem to perform very well for physical problems. Each set of invariants decomposes the tensor shape with an orthogonal basis so that the derivatives of these invariants I_i with respect to the (fiber orientation) tensor $\mathbf{A}$ behave as follows:

$$\frac{\partial I_i}{\partial \boldsymbol{A}} \cdot \frac{\partial I_j}{\partial \boldsymbol{A}} = 0, i \neq j. \tag{2.62}$$

Ennis et al. established the so-called K- and R-invariants, which are also used in this paper. They argue that while eigenvalues form a set of orthogonal invariants as well, they have the disadvantage of not isolating essential attributes of tensor shape like size and anisotropy which the sets of K- and R-invariants provide instead. A direct interpolation of the eigenvalues has still been performed as a test but indeed seemed to distort the results and will not be considered in more detail in this work.

When it comes to the interpolation of the orientation of the tensors, there are multiple works which focus on the interpolation of rotations in 3D, i.e., elements in the 3D rotation group, which is also called $SO(3)$, as a separate mathematical problem independent of tensors or the shape of the same [129–132]. However, many of these papers focus on the smooth rotation between two or few different orientation states. These methods do not necessarily perform just as well for multiple and spreaded orientation tensors. Typically, the orientation of a general basis in 3D linear algebra is described by Euler angles with respect to a fixed coordinate system. Generally, it must be considered that an orientation can be described by 24 different coordinate systems. This ambiguity is counteracted with the help of conventions, this includes the determination to use a right-handed system as well as the sorting of the eigenvalues according to magnitude. However, after the conventions still four possible coordinate systems remain to describe an orientation. Depending on the choice it can be influenced whether between two orientations, which, e.g., lie only 20° apart, the interpolated tensor rotates by 10°, which corresponds to the - at least at first - obviously correct option, or by 170°, which would correspond to the mirrored coordinate system.

For two tensors between which one wants to interpolate, it therefore makes sense to implement a query and restrict the size of the angle. However, for a set of measured tensors $\mathcal{T}_{\mathrm{m}}$ at multiple supporting points, where the ones further away from the one that is currently to be calculated have less weight but are still included in the calculation of this interpolated tensor, this becomes less obvious. For this reason, no angle restriction is implemented in this work (cf. Section 3.7.3). This aspect will be taken up again in the Discussion (cf. Section 5.4.4.4).

A much-investigated method is the orientation interpolation via quaternion. Quaternions as described by Hamilton extend the complex number system and are usually represented in the following form:

$$\begin{aligned} &q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k} \quad \text{with} \\ &\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1 \quad \text{(compare complex numbers).} \end{aligned} \tag{2.63}$$

The result is a four-dimensional number system (mathematically: a vector space) with a real part consisting of one real component and an imaginary part consisting of three components, which is also called the vector part. Multiplication of quaternions is noncommutative. Quaternions allow in many cases a computationally elegant description of three-dimensional Euclidean space, especially in the context of rotations. By using quaternions instead of Euler angles the problem of Gimbal lock can be avoided and they are simpler to compose. Compared to the rotation matrices, quaternions are more compact, efficient, and numerically stable. Regardless, not all ambiguities can be circumvented. The unit quaternions q and $-q$ represent the same rotation. This means that there is a 2:1 homomorphism from quaternions of unit norm to the 3D rotation group $SO(3)$. In other words, $SO(3)$ is double-covered by quaternions. This sign ambiguity has to be paid attention to when computing a quaternion from the rotation matrix.

Alternative fiber orientation interpolation methods were developed, e.g., by Köbler et al. [110]. They developed a mechanical interpolation method in a surrogate model, i.e., they first calculated the material response for discrete fiber orientations and then used linear interpolation on the fiber orientation triangle (a material model as a function of the orientation triangle). While this yields good results, it requires a prior mechanical model and is therefore of no interest in the application thought of in this work.

The use of AI for fiber orientation tensor interpolation has been explored by Sabiston et al. [133]. The authors used a large number of FOT obtained - as in this work - from μCT images of multiple plates of the same process. This represented their ground truth, which they used to train the artificial neural network (ANN), which was subsequently able to predict the tensor components for plates of this process with less deviation than the variability was between neighboring microstructure units. Since this represents an entirely new, non-physical, nor classically linear-algebraic way, a neural network is used as the third interpolation method in this work. However, in contrast to Sabiston's work, the ANN is trained using only the nine measured tensors considered for all methods for comparison. Then, the remaining 160 are predicted using the trained network. Even though the nine tensors give five values of information each, the use of AI with such a small amount of input data is rare, but it is intended to assess whether this can still produce reasonably useful results, or whether a useful result can be expected with a small additional number of given tensors.[2] The usefulness of this approach will be taken up again in the Discussion (cf. Section 5.4.4.3).

Therefore, in this thesis, three different methods are presented in the Methods chapter (see Section 3.7) and their results in the Results chapter (Section 4.6) to obtain a full-field distribution of fiber orientation tensors over a plate. The classical Euclidean interpolation (averaging of the tensor components), a decomposition method based on the separate interpolation of the shape via

[2] Up until here, this section is extracted from the author's publication [17] with only slight linguistic changes.

orthogonal invariants and the orientation of the tensors via quaternions and the adaptation of the ANN developed by Sabiston et al. [133] to the given application are tested. All ideas were not developed in the context of fiber orientation tensors but in theoretical papers focused on mathematics. However, the application of the first two methods to fiber orientation tensors was done by Krauß and Kärger [125] for a slightly different background, namely the mapping of fiber orientation tensors between different meshes. Sabiston's [133] ANN worked with a large number of input tensors from different plates and was intended for the general prediction of average fiber orientations of a process. The application of these methods to small numbers of fiber orientation tensors measured from CT images and the generation of larger numbers between the given ones is, to the author's knowledge, a novelty that first appeared in the self-published paper [17] and proceeding [127].

2.2.4 Representative volume elements and microstructure generation

In order to avoid having to reproduce the complex structure of an entire component when modeling FRP, it is common practice in homogenization approaches to develop a small sub-unit that is representative of the entire microstructure. Such unit cells, whose material response should correspond to the average of an entire part, are - as already mentioned - also called representative volume elements (RVE) [134–136]. Some authors have introduced the idea of statistical (or stochastic) volume elements (SVE) [137–139] incorporating microstructural variability. The edge lengths of such volume elements are often difficult to determine and can be estimated using measured fiber lengths, for example, so as not to shorten them too much or change of microstructural quantities like fiber volume content between sizes. This will be discussed again in this thesis when it comes to specimen extraction (cf. Section 3.1.6). The fiber structure within an RVE can be created either by direct discretization of voxel data from a CT image. This requires, first, that the resolution is high enough for single fiber detection and second, that the contrast to the matrix is high enough for correct

thresholding. However, such an artificial microstructure can also be generated algorithmically and independently of image data. A CT image may serve as a model and fiber volume contents, fiber length distributions and fiber orientation distributions/tensors already measured from the CT images or experimentally can be used as parameters for packing the fibers into the cell. The class of algorithms that can be used for this purpose are called packing algorithms, which developed from the original sphere-packing algorithms. A number of different approaches exist, starting with rather simple ideas such as random sequential adsorption (RSA) introduced by Feder in 1980 [35], where objects are successively packed into the cell at random locations without overlap. A more dynamic approach with velocity vectors is that of Lubachevsky-Stillinger [36]. The mechanical contraction method of Williams-Philipse [38], on the other hand, works by shrinking the cell and only then removing overlaps. Finally, a completely optimization-based approach is that of Torquato-Jiao [37]. These algorithms have been used succesfully for sphere packings and also for simply-structured fiber packings. Schneider [140] introduced the so-called sequential addition and migration (SAM) algorithm in 2017 reliably producing microstructures of short fiber reinforced polymers even with larger aspect ratios and volume fractions. Schneider [141] improved and extended this approach to be able to generate long fiber reinforced structures by allowing curvature of the fibers. There are also other works dealing with microstructure generation of LFT material [142]. Even though curvature of the fibers and higher aspect ratios seem manageable by now, most of these approaches still struggle to reliably generate a microstructure seen in the carbon fiber reinforced polyamide 6 that was subject of this work (cf. Section 4.1). In particular the fineness of the carbon fibers (5 µm - 7 µm) and the mixed bundle and single fiber structure are challenging, especially as single fibers are not even detectable in the CT images. The approach to determining fiber orientation from CT images in the case of this problem is a voxel-based one in this work. So is it possible to generate voxel-based microstructures instead of discrete fibers? In fact, the idea of Generative Adversarial Networks, short GANs, developed by Ian Goodfellow et al. [143], has been around in the field of AI since 2014, generating images from training images. Hence, in this work the generation of two-dimensional

artificial microstructure CT images via GANs is explored with the possible future extension towards directly creating RVEs with these networks.

2.3 Computed tomography

After Wilhelm Conrad Röntgen famously discovered a radiation with the ability to penetrate optically opaque objects at the end of the 19th century, called X-rays [144], the first computerized computed tomography (CT) system was realized by Hounsfield [145] in 1972. Together with Cormack, he won a Nobel Prize in 1979 for this development, as did Röntgen previously for the discovery of X-rays. With time, micro-computed tomography (µCT) systems developed that increased the range of CT applications from medical use towards non-destructive material testing.

2.3.1 Theoretical principles

2.3.1.1 X-ray generation

X-ray radiation is on the electromagnetic spectrum and is generated through the deceleration of fast electrons that are emitted from a heated filament and then enter a solid metal anode [144]. The thereby emitted wavelengths range between approximately 10^{-8} m and 10^{-13} m. The electron velocity v determines the radiation energy. The velocity depends on the acceleration voltage U_a between cathode and anode. Energy conservation leads to the following equation,

$$eU_\mathrm{a} = \frac{1}{2} m_\mathrm{e} v^2, \tag{2.64}$$

with the charge of an electron e and the mass of an electron m_e, with which the electron velocity can be determined. Typical ranges of acceleration voltages can lie between 25 kV and 500 kV, whereas the smaller values are typically

chosen for medical diagnostics and higher values are rather relevant for material testing, where radiation exposure hazards can be neglected. [144]

2.3.1.2 Photon-matter interaction

When X-rays enter a material, the amount of photons, i.e., the radiation density, decreases exponentially due to absorption and scattering. This attenuation of the radiation intensity is material-dependent. The correlation is expressed in the Beer–Lambert law [146–148]

$$I(\eta) = I_0 \mathrm{e}^{-\mu\eta}, \tag{2.65}$$

with the radiation intensity I, a distance variable η, the attenuation coefficient μ and the initial condition $I(0) = I_0$. The linear attenuation coefficient μ is thereby the sum of a scatter coefficient μ_s and an absorption coefficient α [144]:

$$\mu = \mu_\mathrm{s} + \alpha. \tag{2.66}$$

The unit of μ is m^{-1}. Apart from being influenced by the photon energy, the attenuation coefficient is also dependent on the type of irradiated material and more specific on its effective atomic number and its mass density ρ [25, 149]. As a result, the mass attenuation coefficient μ_m gives a density invariant attenuation coefficient and is hence defined as

$$\mu_\mathrm{m} = \frac{\mu}{\rho}. \tag{2.67}$$

Mass attenuation coefficients are often given in $\frac{\mathrm{cm}^2}{\mathrm{g}}$ and tabular collocations of values for different elements and compound materials can be found, e.g., in [150] or in [151] [25].

For composites, the effective mass attenuation coefficient $\bar{\mu}_{\mathrm{m}}$ is calculated from the respective weight fraction w_i and mass attenuation coefficient $\mu_{\mathrm{m},i}$ of each constituent i:

$$\bar{\mu}_{\mathrm{m}} = \sum_i w_i \mu_{\mathrm{m},i} \qquad \text{[25, 152].} \tag{2.68}$$

2.3.1.3 Functionality of a computed tomography system

CT devices consist of the X-ray tube with its anode and cathode that create the effective target area, which is also called the optical X-ray focus. In a CT system, X-rays are created from a nearly point-like source, as bigger diameters result in a so-called penumbra (shadow area, that is reached by only a part of the light) fringe (for details, the reader is referred to Buzug [144]). After passing the object to be scanned, it impinges on a detector, which in modern CT devices typically consists of a scintillator medium and a photon detector (for details cf. [144]) as shown in Figure 2.12. An image of the geometry and material composition of the object can be generated by the detector based on the difference between the initially transmitted beam intensity and the intensity received due to attenuation by the object material. This difference is converted to a spectrum of different gray values. Before the scan, the initial transmitted beam intensity is typically measured in a so-called correction step, where the tube is raised to nearly the uppermost position in order to irradiate onto the detector with no object interference.

The platform on which the object is placed and the detector can be moved in horizontal direction. Thereby, the image region and the resolution can be adjusted [52]. The object platform rotates 360° in the beam path. The CT user can set a number of projections, $n_{\text{projections}}$. Projections are the singular 2D scans taken during a stop of the rotation. Hence, the object rotates in steps of $\varphi = 360°/n_{\text{projections}}$. All of the 2D projections then have to be reconstructed by an algorithm to create the required 3D image of the object. Many different

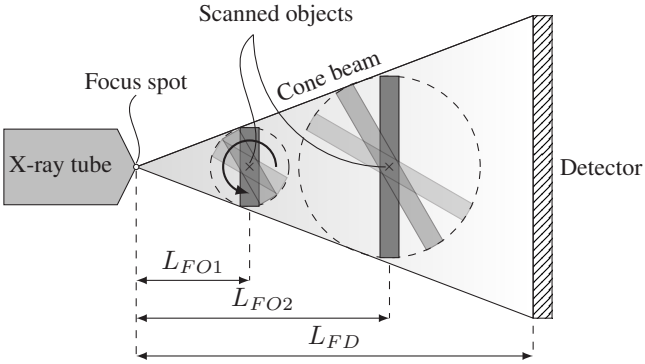


Figure 2.12: Simplified setup of a cone beam computed tomography device. Two alternative positions for differently sized objects in the beam path are shown to illustrate the dilemma between sample size and resolution. Based on Schöttl [25].

algebraic and statistical reconstruction methods have developed over time, many are summed up in [153]. In the present work the cone-beam algorithm that works with convolution-backprojection of Feldkamp et al. [154] is used. It is fairly common and implemented in VG Studio Max 3.4.2; a software that was used in this work for all immediate post-processing, including reconstruction, registration and cutting of regions of interest (ROI). For further processing, the volumetric image obtained this way, is often cut again in 2D slices for easier processing (Matlab, Python) or easier visualization (ImageJ). However, these slices are linearly aligned with a coordinate axis (often perpendicular to the thickness of the specimen) and not a rotation angle.

The volumetric images consist of discrete, in this case cubic, volume elements, that are called voxels (from "**vo**lume **x el**ement"); analogously to pixels in 2D. The voxel size or the in this case isotropic voxel edge length are a measure of the image resolution [25]. The geometric magnification m_g is defined as the division of the distance between focus and detector L_FD by the distance between focus and object L_FO:

$$m_\mathrm{g} = \frac{L_\mathrm{FD}}{L_\mathrm{FO}}, \tag{2.69}$$

as visible in Figure 2.12. The voxel size l_v can subsequently be determined based on the detector pixel size l_d and the magnification as

$$l_\mathrm{v} = \frac{l_\mathrm{d}}{m_\mathrm{g}}. \qquad [25, 155] \tag{2.70}$$

2.3.2 Specific challenges of CT scans of CFRP

2.3.2.1 Low contrast between constituents

Both carbon fibers and polyamide 6 are mainly composed of C elements. Therefore, they have similar densities and even the density independent attenuation coefficient values are very close. For a photon energy of 400 keV, carbon has a mass attenuation coefficient μ_m of $9.546e^{-2}$ [150] and polyamide 6 has an experimentally measured mass attenuation coefficient of $10.86e^{-2}$ at 356 keV [156]. This chemical similarity leads to poor contrast between the two materials in the CT scans, which impedes the determination of fiber-dependent microstructure quantities.

2.3.2.2 Impossibility of single carbon fiber detection

The used µCT device in this work (cf. Section 3.6) could theoretically produce images up to a resolution of 1 µm/voxel. However, there are two main reasons why the resolutions of all scans used in this work were significantly higher: The image noise rises with higher resolutions (cf. next Section 2.3.2.3) and the required specimen sizes to realistically depict the microstructure were too large for higher resolutions. Hence, the resolutions of the scans used in this work range between 8.57 µm/voxel and 25.98 µm/voxel. The diameter of carbon fibers is between 5 µm and 7 µm. In order to reliably segment an individual fiber, one requires voxel sizes that are three to six times smaller than the fiber diameter [13, 25, 76, 157, 158]. This is of course not the case with the resolutions of the scans used in this work. Therefore, it must always be assumed that all methods

only detect agglomerations of fibers or fiber bundles, but not individual fibers. However, since a large proportion of the fibers appear to be in fiber bundles (smaller and larger), the results later still match well with those determined experimentally. In general, only methods without direct fiber detection and instead voxel-based approaches are developed and used in this work.

2.3.2.3 Blur and noise

As Sprawls describes in [159], the main characteristics of CT imaging exacerbating the detection of structures are blur and noise. The former reduces the visibility of small objects and hence decreases image detail. The latter reduces the visibility of low-contrast objects. Fibers fit both of these descriptions. While small voxels and edge-enhancing filters can reduced blurring, small voxels absorb fewer photons and therefore increase noise, which is caused by the variation in attenuation coefficients between voxels. Noisy images can be improved by large voxels, increasing the radiation dose or using smoothing filters. The latter, however, increases blurring again. [159] Since the measures for reduced blur and noise are partially mutually exclusive, the focus of the work was on the reduction of noise, since single-fiber detection was excluded anyway. Hence, larger voxels, increased radiation dose and smoothing filters were used for image improvement.

2.3.2.4 Conflict between image resolution and object size

The size of the specimens was determined individually for different examinations by considerations including fiber length, measured fiber volume contents in different specimen sizes and reasonable RVE microstructures (cf. Section 3.1.6). Therefore, they were at edge lengths between 10 mm and 30 mm. In order to get them completely into the beam path, they had to be moved away from the source accordingly (see Figure 2.12) and this decreased the resolution in turn. This problem generally occurs for CT images and is sometimes called

the conflict between sample size and image resolution [25] and is therefore not limited to the material combination used.

2.3.3 Artifacts

CT images are subject to a variety of artifacts originating from different reasons [144]. There are, e.g., partial volume artifacts, appearing when an object boundary is not located exactly at the edge of a detector element leading to an averaged intensity in these elements and a blurred object boundary. Another well-known artifact is that of beam-hardening. This effect refers to the non-monochromatic energy spectrum of the radiation changing along the path. Low-energy, i.e., soft X-rays are absorbed more than high-energy, hard X-rays, so that the beam hardens while going through the object. This can result in streaking artifacts or cupping artifacts that will cause the middle of the image, which is normally the area around the rotation axis, to decrease in gray value and hence appear darker. Often objects are set in the beam in slight angular offset to the rotation axis due to this artifact. There are various other artifacts like ring artifacts, motion artifacts, etc. which shall not be explained in detail here as this work is not centered around the making of CT scans but rather around their FRP-specific analysis and evaluation but can be explored in appropriate literature, like [144].

2.3.4 Image processing

The processing of digital images through algorithms with the help of digital computers is commonly termed digital image processing [160]. The goals of this processing can range from classification, through feature extraction to pattern recognition. Some of the many conventional techniques that can be used in image processing and are specifically helpful for the application at hand are explained in the following subsections. AI-based methods are explained in Section 2.4. Henceforth, an image has to be understood as a function $i(x, y)$

mapping locations in images to specific - in this case - gray values. For example, in the case of an 8bit gray value image, there exists a gray value between 0 (black) and 255 (white) for every pixel (x, y).

Image processing aims in general at the enhancement or modification of image properties and/or at the extraction of valuable information like edges. The changes can be made directly on the image plane itself, which is called the spatial domain. In contrast, the manipulations can also be performed in a transform domain, which implies a transfer to the transform domain, processing there and a back transfer to the spatial domain. The latter concept is fundamentally based on Fourier transform and will not be explained in more detail in this work; however, important basics for it like convolution (which is also the base for many other concepts in image processing) will be covered in other sections of this work (cf. Section 2.4.1.6). The following sections instead concentrate on spatial processing, of which again two basic categories exist, which are intensity transformations and spatial filtering. While intensity transformations manipulate individual pixels of an image and are fundamental for contrast manipulation, histogram processing and image thresholding (which will be discussed in Section 2.3.4.2), spatial filtering performs operations in a neighborhood of every pixel in an image. The latter concept will be elaborated on in the next Section 2.3.4.1.

2.3.4.1 A selection of spatial filters

A central role in filtering play so-called kernels or masks, an array characteristic for the specific filter. It is a small rectangle with values slid across every pixel of an image, which changes the value of the central pixel according to the result of the filtering operation including the surrounding pixels. This process of moving a filter mask over the image and computing the sum of products at each location is also called correlation [160] and defined as

$$f(x,y) * I(x,y) = \sum_{i=0}^{m} \sum_{j=0}^{n} f(i,j) I(x+i, y+j), \tag{2.71}$$

with a filter $f(x,y)$ of size $m \times n$ (both odd integers) and the image $I(x,y)$. Correlation is similar but not identical to convolution; a small subtlety that is not adequately represented in most literature. In the case of convolution, there would be minus signs instead of the addition on the right, which flip f (i.e., rotate it by 180°). Correlation therefore corresponds to convolution if the filter f is simply rotated by 180°. However, since most filter masks are known by their representation for correlation, this is also the used convention in this work. It is anyways only relevant in the case of an asymmetric kernel.

Depending on the kernel, a filter can have a smoothing effect on an image or a sharpening effect, e.g., for enhancing edges. Smoothing filters are used for blurring and noise reduction and will be elaborated on in the next paragraphs. Linear spatial filter output the average of the pixels in the neighborhood of the filter mask, which is why they are also called averaging filters or lowpass filters. They reduce sharp transitions in intensity, leading to reduced noise but increased blurring of edges. [160] Different kinds of smoothing filters will be shortly addressed in the following. It shall be mentioned that different, odd kernel sizes are possible, depending on the desired effect (the larger the kernel, the stronger the blur) and the image resolution. For simplicity reasons, a kernel size of three is depicted for all displayed filters. Furthermore, the question of how to handle the edges of an image naturally arises. When the central pixel in the kernel is an edge pixel, some of the kernel positions are not occupied. One can use only the pixels next to the edge pixels as central pixels but that results in a reduction of dimensions of the resulting image. Another possibility is the so-called *padding*, which refers to the addition of extra rows and columns of values to solve that problem. For example, the well-known so-called zero padding signifies the addition of zeros around the borders, leading to a black frame around the image. There are also other padding possibilities like mirroring the value next to the border, etc. and the reader is referred to [160–163].

A standard 3×3 *blur filter*, which is a linear smoothing filter, gives the average of the nine surrounding pixels:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{2.72}$$

For reasons of computational simplicity, the coefficients are all one and the normalization factor is multiplied afterwards. An important characteristic of the standard blur filter is the uniformed weighting of all pixels in the kernel. It is sometimes also called a *box filter*.

The *Gaussian blur* filter is similar to the box filter but uses a weighted mean, with neighborhood pixels that are closer to the central pixel having a higher weight. The weights drop in the intensity of a Gaussian normal function. Thus, the kernel can be determined by the two-dimensional Gaussian function defined by

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{2.73}$$

with x and y being the respective distances to the horizontal and vertical center of the kernel and σ being the standard deviation of the Gaussian kernel. An often used approximation of the correctly calculated Gaussian blur kernel looks as follows:

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{2.74}$$

The Gaussian smoothing is supposed to deliver a more natural blur and better preserve edges in the image compared to the average blur.

A *median filter*, as a representative of order-statistic, nonlinear filters, does not compute a new value from the values present in the kernel but instead replaces the central pixel value with the median value existent in the mask, as demonstrated in the following Figure 2.13.

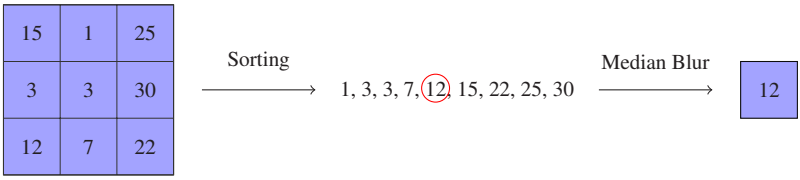


Figure 2.13: Graphical demonstration of the principle of the median blur filter.

It is highly effective against so-called salt and pepper noise and robust against outliers. Unlike in the case of the average blur filter, a rectangular kernel dimension is impossible; it must be square.

Another noise reducing, smoothing filter is the *bilateral filter* [164]. It is a non-linear filter used to blur images while preserving object edges. The filter owes the preservation of edges to the fact that, unlike the Gaussian blur, the weights depend not only on the Euclidean distance of the pixels so not only on their geometric closeness but also on their photometric similarity (e.g., range differences, such as color intensity, depth distance, etc.). It prefers near values to distant values in both domain and range. Different filter kernels can be used for both the closeness function $c(\xi, \boldsymbol{x})$ and the similarity function $s(\phi, i)$. Gaussian functions of the Euclidean distance between their arguments are widely used for both. So bilateral filtering denotes the combined domain and range filtering and delivers an output image $o(\boldsymbol{x})$ for an input image $I(\boldsymbol{x})$ as

$$o(\boldsymbol{x}) = k^{-1}(\boldsymbol{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\xi) c(\xi, \boldsymbol{x}) s(I(\xi), I(\boldsymbol{x})) d\xi, \tag{2.75}$$

with the normalization

$$k(\boldsymbol{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, \boldsymbol{x}) s(I(\xi), I(\boldsymbol{x})) d\xi. \qquad [164] \qquad (2.76)$$

2.3.4.2 Thresholding approaches

Thresholding refers to the selection of one or more intensity values/limits that divide the histogram into two or more areas in each of which only the pixels/voxels of a specific object/material appear. These groups are often called classes. This subdivision of the image can be highlighted by binarization:

$$o(x, y) = \begin{cases} 1, & \text{if} \quad i(x, y) > T \\ 0, & \text{if} \quad i(x, y) \leq T \end{cases} . \qquad (2.77)$$

If more than two threshold values would be necessary for the segmentation problem at hand, one oftentimes has to switch to alternative methods. Furthermore, the successful application of certain thresholding methods requires in particular clear separation of the intensity peaks and low noise content, which makes it clear why this method can also cause difficulties when used for the CT scans in this work. However, due its (computational) simplicity, thresholding is still highly relevant to this day. There are different subcategories of thresholding. The basic case of a constant threshold across an entire image is called *global thresholding*. Conversely, there is *variable thresholding*. If the threshold at a point is dependent on its neighborhood, this variable thresholding is also called *local* or *regional thresholding*. The term *dynamic* or *adaptive thresholding*, on the other hand, is used when the threshold T depends directly on the spatial coordinates x and y. However, Gonzalez and Woods point out the inconsistent use of these terms in the image processing literature. [160]

The determination of a global threshold, if applicable, often requires a more sophisticated algorithm and cannot be easily deduced. In the following, the calculation of three well-known, global and automatic thresholding methods are described as examples, which are usually already pre-implemented in scientific image processing software such as ImageJ (FIJI) or in MATLAB libraries or Python image processing packages and were used in different cases throughout this work.

The method by Nobuyuki Otsu, called *Otsu's method* or *Otsu threshold*, finds a threshold separating foreground and background by minimizing the intra-class, also called the within-class intensity variance, or equivalently, by maximizing the inter-class, also called between-class, variance [47]. Granted a picture is represented in L gray levels and a threshold t exists at one of those levels. The within-class variance σ^2_{W} is defined as follows

$$\sigma^2_{\mathrm{W}} = \omega_0\sigma_0^2 + \omega_1\sigma_1^2, \tag{2.78}$$

with ω_0 and ω_1 being the probabilities of the respective class occurrence and σ_0 and σ_1 being the respective class variances. The probabilites are calculated by

$$\omega_0(t) = \sum_{i=0}^{t-1} p_i \quad \text{and} \quad \omega_1(t) = \sum_{i=t}^{L-1} p_i. \tag{2.79}$$

Of course, $\omega_0 + \omega_1 = 1$ holds.

The class variances are given by

$$\sigma_0^2 = \sum_{i=0}^{t-1}(i-\mu_0)^2\frac{p_i}{\omega_0} \quad \text{and} \quad \sigma_1^2 = \sum_{i=t}^{L-1}(i-\mu_1)^2\frac{p_i}{\omega_1}. \tag{2.80}$$

The between-class variance σ^2_{B} is given by

$$\begin{aligned}\sigma_{\mathrm{B}}^2 &= \omega_0(\mu_0 - \mu_{\mathrm{T}})^2 + \omega_1(\mu_1 - \mu_{\mathrm{T}})^2 \\ &= \omega_0\omega_1(\mu_1 - \mu_0)^2,\end{aligned} \tag{2.81}$$

with the class mean levels

$$\mu_0(t) = \sum_{i=0}^{t-1} \frac{ip_i}{\omega_0(t)} \quad \text{and} \quad \mu_1(t) = \sum_{i=t}^{L-1} \frac{ip_i}{\omega_1(t)} \quad \text{and} \quad \mu_{\mathrm{T}} = \mu(L) = \sum_{i=0}^{L-1} ip_i. \tag{2.82}$$

Finally, the optimal threshold t is found by either maximizing the between-class variance or minimizing the within-class variance:

$$t = \arg\left\{\max_{0\le t\le L-1}\left\{\sigma_{\mathrm{B}}^2(t)\right\}\right\} = \arg\left\{\min_{0\le t\le L-1}\left\{\sigma_{\mathrm{W}}^2(t)\right\}\right\} \text{[47, 165]}. \tag{2.83}$$

Another automatic thresholding procedure is the *moment-preserving threshold* by Tsai [48]. Thereby, a threshold value is selected so that the first three moments of image i in the resulting binarized image o are preserved. The moments of i are defined by

$$m_i = \frac{1}{n}\sum_j n_j(z_j)^i = \sum_j p_j(z_j)^i, \tag{2.84}$$

with n being the entire amount of pixels of image i, n_j being the total number of pixels in i with the gray value z_j and $p_j = \frac{n_j}{n}$. After having applied the threshold and binarized the image, the first three moments of o can be computed in the following way:

$$m'_i = \sum_{j=0}^{1} p_j (z_j)^i, \quad i = 1, 2, 3; \tag{2.85}$$

where p_0 and p_1 signify the fractions of below-threshold and above-threshold pixels, respectively. Preserving the first three moments of i in o means,

$$m'_i = m_i, \quad i = 1, 2, 3 \tag{2.86}$$

has to hold. Furthermore,

$$p_0 + p_1 = 1. \tag{2.87}$$

The last two equations can be translated into the following equation system

$$\begin{aligned} p_0 z_0^0 + p_1 z_1^0 &= m_0, \\ p_0 z_0^1 + p_1 z_1^1 &= m_1, \\ p_0 z_0^2 + p_1 z_1^2 &= m_2, \\ p_0 z_0^3 + p_1 z_1^3 &= m_3. \end{aligned} \tag{2.88}$$

m_i is calculated with the help of Equation (2.84) and m_0 is defined to be 1. Then, the equations are solved to obtain p_0 and p_1, which inevitably leads to choosing t such that

$$p_0 = \frac{1}{n} \sum_{z_j \leq t} n_j. \tag{2.89}$$

There might not exist an exact value for t fulfilling this condition, so the closest gray value should be chosen. [48]

Moreover, the *mean threshold* [49–51] is defined by choosing t as the mean value of the distribution of pixel values. Analogously, the *median threshold*, as a special method of Doyle's p-tile method [166], is set so that 50 % of pixels lie in each of the two binary classes [49].

The use of those different thresholding techniques concerning carbon fiber reinforced polymers will be taken up again in the Methods (Chapter 3, Section 3.4.2.2 and Section 3.6).

2.4 Artificial intelligence

The term *artificial intelligence* describes the intelligence of machines or software. It is a field of study in computer science and its technologies have spread across various applications throughout industry, governments, science and entertainment. Initiated by Alan Turing in the 1950s [167], AI rose to its initial promise only in the last decade as large computational power and big data storage and handling possibilities emerged [168]. This led to the AI boom, sometimes called *AI Spring* [169], in the later 2010s and early 2020s.

The large field of AI is further differentiated. *Machine learning (ML)* is probably the largest sub-category [168]. It refers to algorithms that can automatically improve the execution of a specific task [170]. Hong et al. [168] emphasize that the major difference between conventional programming and machine learning programming is ML working through induction. In the former, a programmer has to formulate general rules and the algorithm is able to deduce the result for specific observations, while in ML the algorithm itself finds general rules from the given observations [168]. The programmer's work lies in the development of the structure and environment of the algorithm for it to reliably induce those rules. Either another subcategory of ML or a mixture of AI and ML concepts (depending on the definition), is the so-called *deep learning (DL)* [168]. Deep learning refers to processing the observations through ANNs that are inspired by biological neural networks. The adjective "deep" stresses the use of multiple

layers in the network. There is the further differentiation into *supervised* and *unsupervised* ML. The former denotes networks that work on input data that was labelled by a human, while the latter type signifies networks that have to find common patterns and make predictions without any other human guidance [170].

In the following, the general theoretical principles of ANN will be elaborated on, before specific types of those networks will be outlined.

2.4.1 Theoretical principles of ANNs

2.4.1.1 Single layer perceptrons

The smallest sub-unit of an ANN is a so-called single layer perceptron (SLP). Based on the original neuron model by McCulloch and Pitts in 1943, it was first introduced as perceptron by Rosenblatt in 1957 and is a linear classifier whose structure resembles that of a biological neuron (cf. Figure 2.14). [171, 172]

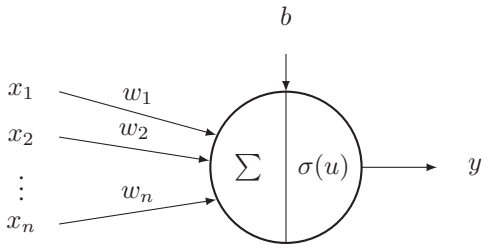


Figure 2.14: Schematic depiction of a single layer perceptron as described by Rosenblatt [171]. Based on [173].

Different inputs $\{x_1, x_2, ..., x_n\}$ (similar to the impulses collected by dendrites of a biological neuron) are weighted based on their relative importance by the weights $\{w_1, w_2, ..., w_n\}$ and subsequently summed up by a linear aggregator. A lower threshold value b, also known as the bias, determines whether an output signal is afterwards triggered. The activation function $\sigma(u)$ depends on

the difference between b and the result of the linear aggregator, the activation potential u. This results in the output signal y of the SLP

$$y = \sigma(u) = \sigma\left(\sum_{i=1}^{n} w_i x_i - b\right). \tag{2.90}$$

The activation function thereby restricts the possible output values to a set range. [174] Equation (2.90) can be simplified by adding a dummy input $x_0 = 1$ and the corresponding weight $w_0 = b$ to

$$y = \sigma\left(\sum_{i=0}^{n} w_i x_i\right) = \sigma(\boldsymbol{x} \cdot \boldsymbol{w}) = \sigma(\boldsymbol{w}^\intercal \boldsymbol{x}), \tag{2.91}$$

which increases the vector dimensions of $\boldsymbol{x}$ and $\boldsymbol{w}$ by 1 [175].

There are many different possible activation functions including the step function, linear function, sigmoid function, tanh function or a rectified linear unit (ReLu) [174]. Assuming a simple bipolar step function for demonstration purposes, which is given by

$$y = \sigma(\boldsymbol{x} \cdot \boldsymbol{y}) = \begin{cases} 1 & \text{if} \quad \boldsymbol{w} \cdot \boldsymbol{x} > 0, \\ -1 & \text{if} \quad \boldsymbol{w} \cdot \boldsymbol{x} \leq 0, \end{cases} \tag{2.92}$$

one gets a binary output y independent of the (possibly non-binary) input [174, 175]. If only two input values and weights were given, this would result in the following inequalities

$$\begin{aligned} w_1 x_1 + w_2 x_2 - b \geq 0, \\ w_1 x_1 + w_2 x_2 - b < 0, \end{aligned} \tag{2.93}$$

which creates a linear boundary between two classes [174]. In case of more than two input and weight parameters, this approach has to be extended to a hyperplane defined by $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ [176]. This function of SLP as classifier is shown in Figure 2.15.

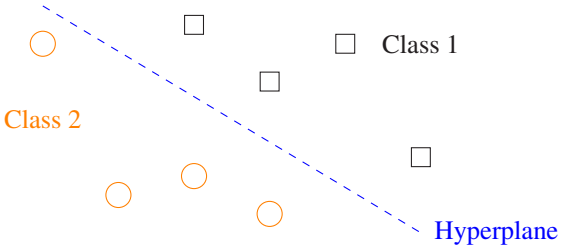


Figure 2.15: Schematic two-dimensional illustration of a hyperplane (dashed blue line) separating two classes originating from the system of inequalities given by a SLP. Based on [176].

The perceptron learns by adjusting the weights when it has made wrong decisions. This updating or learning rule is a recursive algorithm. Suppose that the respective class label z_i of a given input value x_i is given by

$$z_i = \begin{cases} 1 & \text{if} \quad x_i \in \text{of the first class,} \\ -1 & \text{if} \quad x_i \in \text{of the second class.} \end{cases} \tag{2.94}$$

For a set of p training samples $(x_i, z_i) \quad \forall \quad i = 0, 1, ..., p$, then $y = y(w)$ gives the (binary) prediction of z_i for a given x_i based on the current weights and bias. The change of the weights w_i is then defined as

$$w_i \rightarrow w_i + \Delta w_i, \tag{2.95}$$

with

$$\Delta w_i = r_1(z_i - y_i)x_i. \tag{2.96}$$

This learning rule is also called *delta rule* and is a gradient descent algorithm for optimization. The parameter $r_1 \in [0, 1]$ denotes the learning factor or more commonly known the *learning rate*. The convergence of this learning rule after a finite number of iterations was proven by Rosenblatt in the perceptron convergence theorem. [171, 175, 176]

2.4.1.2 Multi-layer perceptron (MLP)

A standard neural network normally consists of multiple layers of perceptrons, more precisely an input layer, some number of hidden layers and an output layer; so at least three layers. The input layer is often not counted in the amount of layers as it performs no processing and only has a distribution function towards the next layer [177]. These kinds of neural networks are often referred to as "vanilla" neural networks emphasizing their simplicity and them being the first of their kind [178]. A simple fully connected MLP with one hidden layer can be seen schematically in Figure 2.16.

The input values $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ are often referred to as "features" in the context of neural networks making $\boldsymbol{x}$ the feature vector [174]. Following, one or multiple layers of hidden notes h_i process the input until it reaches the final output layer, where the results are given out in the output vector $\boldsymbol{y}$. An activation function maps the weights to the following layer [179]. The name multi-layer perceptron is misleading or incorrect in that contrary to the step function used as activation function by the original perceptron, modern neural networks rely on nonlinear kinds of activation functions providing them the ability to distinguish data that is not linearly separable [180]. Different kinds of activation functions will be shortly outlined in Section 2.4.1.5. In this simple case, connections are only permitted between nodes in consecutive layers and in a forward direction, which is why they are also called *feedforward* networks [175].

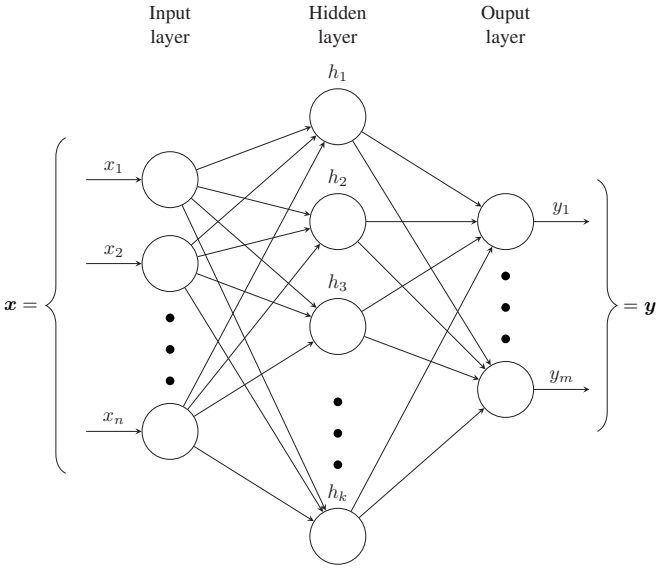


Figure 2.16: Schematic visualization of a simple MLP structure with every node incorporating a linear aggregator, activation function and bias. Based on [176].

Analogously to Equation (2.90), assuming that all layers use the same activation function σ, the output of a single node j in a specific hidden layer $h^{(l)}, 1 \leq l \leq N$, is given by

$$h_j^{(l)} = \sigma \left(\sum_{i=1}^{k_{l-1}} w_{ij}^{(l)} h_i^{(l-1)} - b_j^{(l)} \right), \tag{2.97}$$

where k_l denotes the total number of nodes in the hidden layer $h^{(l)}$ and $w_{ij}^{(l)}$ the weights between a unit j in the lth hidden layer and another unit i in the preceding layer [181]. It shall be noted that different layers may have different activation functions, which would require a distinction to be made between the different σ. Equivalent expressions can be found for the output of the first hidden layer $h^{(1)}$ as

$$h_j^{(1)} = \sigma\left(\sum_{i=1}^{n} w_{ij}^{(1)} x_i - b_j^{(1)}\right), \tag{2.98}$$

with the number of input features n, and the output values y_j as

$$y_j = \sigma\left(\sum_{i=1}^{k_{l-1}} w_{ij}^{(l)} h_i^{(l-1)} - b_j^{(l)}\right). \text{[181]} \tag{2.99}$$

Considering an example of two consecutive hidden layers $h^{(1)}$ and $h^{(2)}$ with the number of nodes k_1 and k_2, exemplarily shown in Figure 2.17 with $k_1 = 4$ and $k_2 = 3$, the output of the first unit $h_1^{(2)}$ in the second layer can be calculated based on Equation (2.97) as

$$h_1^{(2)} = \sigma(w_{11}^{(2)} h_1^{(1)} + w_{12}^{(2)} h_2^{(1)} + ... + w_{1k_1}^{(2)} h_{k_1}^{(1)} - b_1^{(2)}). \tag{2.100}$$

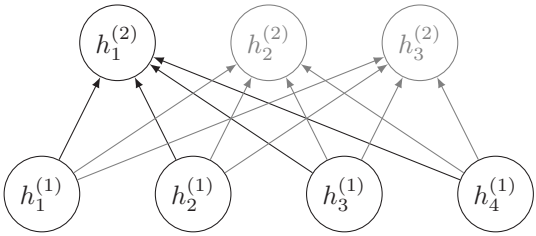


Figure 2.17: Schematic description of two consecutive hidden layers $h^{(1)}$ and $h^{(2)}$ of an MLP. For the sake of simplicity, previous or subsequent layers are not shown. In order to illustrate the calculation of the output of node $h_1^{(2)}$, all other units that are not directly connected are grayed out.

Repeating this calculation for all other units in the second layer results in k_2 different equations. Combining all outputs of the preceding layer as a single vector $\boldsymbol{h}^{(1)}$ and the biases as vector b yields the linear algebraic description of the output vector $\boldsymbol{h}^{(2)} = h_1^{(2)}, h_2^{(2)}, ..., h_{k_2}^{(2)}$ of the second layer as

$$\boldsymbol{h}^{(2)} = \sigma\left(\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1k_1} \\ w_{21} & w_{22} & \cdots & w_{2k_1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k_2 1} & w_{k_2 2} & \cdots & w_{k_2 k_1} \end{bmatrix} \begin{bmatrix} h_1^{(1)} \\ h_2^{(1)} \\ \vdots \\ h_{k_1}^{(1)} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{k_1} \end{bmatrix}\right), \tag{2.101}$$

with the weight matrix $\boldsymbol{W} = [w_{ij}^{(2)}]$ for $1 \leq j \leq k_1$ and $1 \leq i \leq k_2$. This symbolic tensor notation compresses the process between two layers into one equation and simplifies the use in code:

$$\boldsymbol{h}^{(2)} = \sigma\left(\boldsymbol{W}^{(2)}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)}\right). \tag{2.102}$$

Generalized for any hidden layer $\boldsymbol{h}^{(l)}$, this yields

$$\boldsymbol{h}^{(l)} = \sigma\left(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}\right). \quad [182, 183] \tag{2.103}$$

Their more complex structure allows MLPs to be applied to a variety of problems from pattern recognition [181] over clustering to optimization [176]. However, some disadvantages of fully connected layers include

- the large number of parameters
- slow convergence of training and
- bad generalization effect when applied to new training data. [179]

While the complexity of an MLP or an ANN can be increased arbitrarily both in the amount of layers as well as in the amount of nodes per layer, this leads to the requirement of high computational power or time and to increased risk of so-called *overfitting* [184]. Overfitting denotes the network being perfectly

trained on the input/training data and hence also fitted to the noise and irregular peculiarities thereof, instead of finding a general predictive rule that allows good predictions on new data points [185]. Increased model complexity and overfitting is tantamount to decreasing bias (error due to the used model's incapability of capturing the underlying model) and increasing variance (error due to sensitivity to noise in the data). The opposite called underfitting hence typically shows high bias and low variance. This dilemma of finding the optimal model complexity is depicted in Figure 2.18. [186]

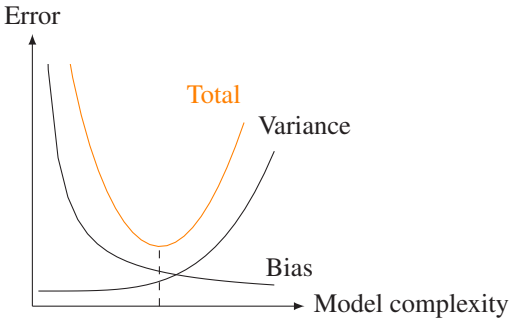


Figure 2.18: Qualitative depiction of the relationship between variance and bias and their effect on the error of the network. Models too simple to capture the underlying real model typically show high bias and low variance; they are underfitting. In contrast, an overly complex model is likely to have low bias and high variance and is therefore overfitting. Based on [186].

2.4.1.3 Training and backpropagation

The basic training process of a neural network is based on iterative optimization, illustrated in Figure 2.19.

The training of a multilayer ANN is in general similar to the training of an SLP, which was introduced in Equations (2.95) and (2.96) as delta rule. The production of output through the forward propagation of activation was introduced in the last section. However, in most modern feedforward networks a

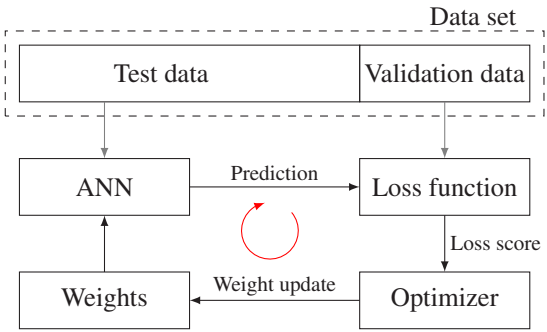


Figure 2.19: Illustration describing the training of weights in neural networks according to [25, 187]. Extracted from Blarr et al. [188].

second training phase is involved, where the error is propagated backwards in the so-called *backpropagation*. [177] This efficient application of a chain-rule based supervised learning [189] was introduced by Linnainmaa [190] with significant improvements and analyses by Werbos [191] and Rumelhart [192]. According to Rojas, the backpropagation algorithm can be decomposed into four main steps that are the feedforward computation, the backpropagation to the output layer, the backpropagation to the hidden layer and finally the weight updates [193]. When the results of the feedforward computation are available, there exists a network output, hence prediction, $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{w})$, for every of the q training vector pairs of input and output $(\boldsymbol{x}, \boldsymbol{y})$. Then, an error or cost function E comparing output $\boldsymbol{y}$ and desired target $\boldsymbol{z}$ is to be minimized

$$\min_{w} E = \sum_{p=1}^{q} E_p, \tag{2.104}$$

where E_p is some kind of distance function. The sum of squares error (SSE) based on Euclidean distance and given by

$$E_p = ||z_p - y_p||^2, \tag{2.105}$$

is a popularly applied error function. [194]

E is computed solely through the composition of the node functions and hence forms a continuous and differentiable function of the l network weights $\{w_1, w_2, ..., w_l\}$. In order to minimize this error, an iterative process of gradient descent is used, for which the gradient of the error function has to be determined given by

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, ..., \frac{\partial E}{\partial w_l} \right). \tag{2.106}$$

The weights can subsequently be adjusted incrementally by descending the gradient as follows

$$\Delta w_i = -r_1 \frac{\partial E}{\partial w_i}, \qquad i = 1, 2, ..., l, \tag{2.107}$$

with learning rate r_1 defining the step length of each iteration. This procedure is repeated for a finite number of steps, steadily choosing the correction path with the steepest descent, until a minimum of the error function is found with $\nabla E = 0$ or a satisfatory reduction of the error function is achieved. [193]

This so-called *generalized delta rule* [194] is illustrated in Figure 2.20.

Disadvantages of classical backpropagation include slow convergence speed and the strong dependence on chosen parameters, like initial weights and especially the learning rate. In addition, real error surfaces do not look like the simplified shape in Figure 2.20 but instead often show ravine-like features and many dent-like local minima [173]. This may result in the backpropagation training the network into a local minimum that it might not be able to escape. However, reaching the global minimum is in practice not always necessary, provided a set of weights is found that works accurately enough for the application at hand. [173, 195]

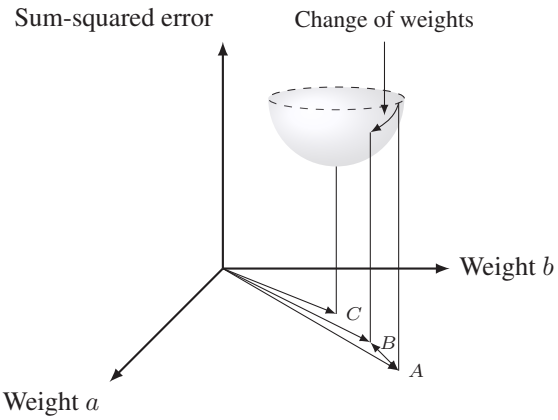


Figure 2.20: Idealized bowl-shaped error surface for 2D weights, where the weights are changed following the steepest path towards the bottom C, which is the desired minimum (with current weights at point A and corrected weights at B). Based on [173].

2.4.1.4 Optimizing algorithms

Even though a variety of optimizing algorithms to minimize the loss function have been developed, only two will be shortly addressed in this section, which are the stochastic gradient descent (SGD) and the ADAM optimizer. The choice is based on the fact that these two perform well for common problems and are among the most widely used. Furthermore, of the three different networks implemented in this work, one works with SGD and the other two with ADAM.

Stochastic gradient descent (SGD) The basic gradient descent, first published by Cauchy [196], has been introduced in the previous Section 2.4.1.3 in Equation (2.106) and Equation (2.107) and is assembled here for the sake of completeness again:

$$w_{i+1} \rightarrow w_i - r_1 \nabla E(w_i). \tag{2.108}$$

It shall be noted that $r_1 = r_{1,i}$ in the case of non-constant learning rates [197]. It is sometimes also referred to as batch gradient descent as it is an optimization method performed on the entire training set, which requires high computational effort [197, 198]. In contrast there are stochastic optimization methods, of which SGD, first described by Robbins and Monro [199] and in a form closer to today's SGD in DL by Kiefer and Wolfowitz [200], is the most prominent representative [197]. SGD performs a weight update on every training example x_i, y_i:

$$w_{i+1} \rightarrow w_i - r_1 \nabla E(w_i, x_i, y_i) \qquad [198], \tag{2.109}$$

which can be regarded as a stochastic estimate of the actual entire gradient. SGD typically achieves fast iterations due to its reduced computational demand in exchange for a lower convergence rate [197]. However, in practice nowadays, often neither purely stochastic nor purely batch optimization methods are used, as the SGD can be used on randomly chosen subsets of the data, or so-called *mini-batches* without constraints [197]. It is mostly still referred to as SGD [198] or sometimes as mini-batch gradient descent.

The ADAM optimizer One of the most used optimizing methods in DL is the Adaptive Moment Estimation (ADAM) presented by Kingma et al. in 2014 [201, 202]. It is an algorithm for first-order gradient-based optimization based on adaptive estimates of lower-order moments [201]. While there are many different definitions of the ADAM optimizer [202], the original algorithm as in [201] is introduced in the pseudocode (Algorithm 1) below.

Empirical results show that the ADAM optimizer outperforms many other known optimizers like "AdaGrad" or SGD both in logistic regression as well as in multilayer fully connected neural networks and deep convolutional neural networks [201] concerning its training cost. Its local convergence has been proven [202]. For further information, the reader is referred to the original paper [201].

Algorithm 1 Pseudo code for ADAM optimizer computation. The proposed parameter settings are $r_1 = 0.001$, $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. Slightly modified from: [201].

Require: r_1: Step size (i.e. learning rate)
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for estimating the moments
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ ▷ Initialisation of first moment vector
$v_0 \leftarrow 0$ ▷ Initialisation of second moment vector
$t \leftarrow 0$ ▷ Initialisation of time step
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ ▷ Obtain gradients w.r.t. the stochastic objective at time step t
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ ▷ Update biased first moment estimate
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ ▷ Update biased second moment estimate
$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ ▷ Compute bias-corrected first moment estimate
$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ ▷ Compute bias-corrected second moment estimate
$\theta_t \leftarrow \theta_{t-1} - r_1 \cdot \hat{m}_t / (\sqrt{\hat{v}_t + \epsilon})$ ▷ Update parameters
end while
return θ_t ▷ Resulting parameters

2.4.1.5 Common activation functions

It has already been mentioned that the activation functions have to be non-linear in order for the model to be able to learn complex correlations. Many different activation functions are possible and have been used, of which some major ones shall be presented in the following. In general, an activation function is defined as a function $h : \mathbb{R} \to \mathbb{R}$, which is continuously differentiable almost everywhere (requirement for backpropagation) [203].

The activation of a single hidden unit $h^{(l)}$ is given by

$$h^{(l)} = \sigma(u) = \sigma(\boldsymbol{w}^{(l)\intercal} \boldsymbol{x}), \tag{2.110}$$

with σ as the associated activation function, $\boldsymbol{w}^{(l)}$ as the corresponding weight vector and $\boldsymbol{x}$ as the input vector (compare Equation (2.91)) [204].

Binary step function The activation function that is closest to the reaction of a biological neuron is a simple binary step function, which is either zero or positive, defined by

$$h^{(l)} = \begin{cases} 1 & \text{if} \quad \boldsymbol{w}^{(l)\intercal}\boldsymbol{x} \geq 0, \\ 0 & \text{if} \quad \boldsymbol{w}^{(l)\intercal}\boldsymbol{x} < 0, \end{cases} \tag{2.111}$$

and depicted in Figure 2.21.

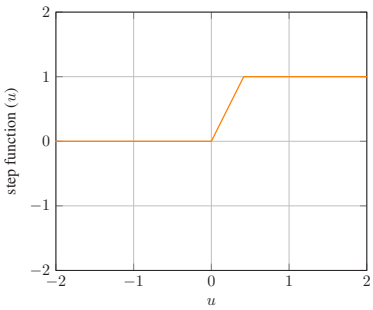


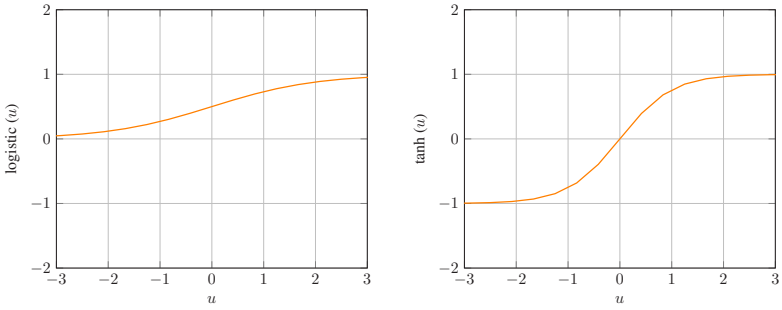
Figure 2.21: Binary step function.

Sigmoid functions Sigmoid functions are bounded and differentiable functions, which are non-decreasing and have exactly one inflection point [205]. Two sigmoid functions are shortly outlined, the *logistic* sigmoid function, which is sometimes only referred to as sigmoid function, and the *hyperbolic tangent* function.

The logistic sigmoid function is defined as follows

$$h^{(l)} = \frac{1}{1 + \exp\left(-\boldsymbol{w}^{(l)\intercal}\boldsymbol{x}\right)}, \tag{2.112}$$

which translates any input between $(-\infty; \infty)$ to the range $[0; +1]$ and can therefore be thought of as a smooth version of the binary step function [205]. While the binary step function is not differentiable at $x = 0$, the logistic function is differentiable infinitely many times on its entire domain [205, 206]. Owing to their effect of "squashing" the real values into a bounded interval, they are occasionally called squashing functions (cf. Figure 2.22) [205]. Hence, they can suffer from vanishing gradients, especially in deep networks [206, 207]. The logistic sigmoid function is therefore often used in rather shallow networks or often in output layers of networks due to its value distribution [206].



(a) Logistic (b) Tanh

Figure 2.22: Sigmoid (a) and tanh (b) activation functions.

As a further development, the similiar hyperbolic tangent function is given by

$$\tanh(u) = 2\text{sigmoid}(2u) - 1, \tag{2.113}$$

which gives the following activation function:

$$h^{(l)} = \frac{2}{1 + \exp\left(-2\boldsymbol{w}^{(l)\intercal}\boldsymbol{x}\right)} - 1 \quad [206, 207]. \tag{2.114}$$

Bounding the input values in the range of $[-1; +1]$ (cf. Figure 2.22(b)), it has a steeper derivative than the logistic function but suffers from vanishing gradients nonetheless [207]. The gradients of sigmoid funtions in general converge towards zero at the outer limits and are therefore soft-saturating:

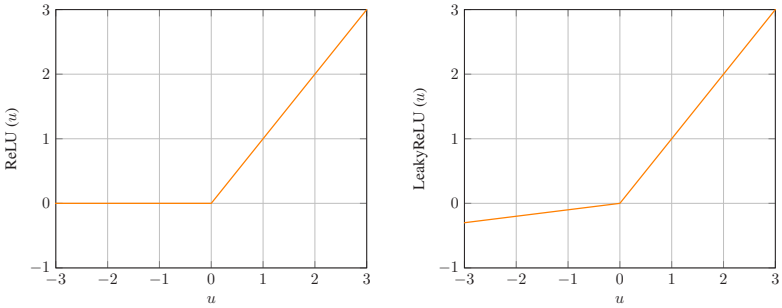
$$\lim_{x \to +\infty} \sigma'(u) = 0, \qquad \lim_{x \to -\infty} \sigma'(u) = 0 \quad [206]. \tag{2.115}$$

Similarly, the hyperbolic tagent is used mostly in output layers as well, especially in the case of input values that were mapped between minus one and one [206].

Rectified Linear Unit (ReLU) and Leaky Rectified Linear Unit Inspired by neuroscience, Nair et al. [208] presented the ReLU as a piecewise linear activation function in 2010 initially for the use in restricted Boltzmann machines, which has since become one of the most used activation functions [207]. In contrast to the previously shown functions, ReLU is a non-saturated function resulting in high convergence speed and the avoidance of vanishing gradients [209]. The function can be expressed by

$$h^{(l)} = \max(\boldsymbol{w}^{(l)\intercal}\boldsymbol{x}, 0) = \begin{cases} \boldsymbol{w}^{(l)\intercal}\boldsymbol{x} & \text{if} \quad \boldsymbol{w}^{(l)\intercal}\boldsymbol{x} \geq 0, \\ 0 & \text{if} \quad \boldsymbol{w}^{(l)\intercal}\boldsymbol{x} < 0, \end{cases} \tag{2.116}$$

and can be seen on the left in Figure 2.23. Disadvantages include the death of neuron units that were not initially activated as their weights will not be updated and a slow training process for gradients constantly being zero [204, 206].



(a) ReLU (b) Leaky ReLU with $\lambda = 0.1$

Figure 2.23: Original ReLU and Leaky ReLU activation function.

On that account, Maas et al. introduced the Leaky ReLU as improvement of the standard ReLU function, which is defined by

$$h^{(l)} = \max(\boldsymbol{w}^{(l)^\intercal}\boldsymbol{x}, 0) = \begin{cases} \boldsymbol{w}^{(l)^\intercal}\boldsymbol{x} & \text{if} \quad \boldsymbol{w}^{(l)^\intercal}\boldsymbol{x} \geq 0, \\ \lambda\boldsymbol{w}^{(l)^\intercal}\boldsymbol{x} & \text{if} \quad \boldsymbol{w}^{(l)^\intercal}\boldsymbol{x} < 0 \quad [204]. \end{cases} \tag{2.117}$$

The slope in the Leaky ReLU function depicted on the right in Figure 2.23 is chosen to $\lambda = 0.01$ as suggested in the original paper [204]. When the unit is saturated and not active, the leaky rectifier allows for a small, non-zero gradient unlike the standard ReLU activation [204]. Sacrificing the so-called hard-zero sparsity of ReLU allows for a gradient which is potentially more robust during optimization [204].

2.4.1.6 Types of layers

The effect of increasing the number of layers to improve a network is limited, which is why alternatives to the previously introduced fully connected layers (also known as linear layers) have been developed. Some of the most important

ones used in the networks in this work are briefly explained below. In some cases, the use of a certain type of layer also determines the name of the network, for example, an ANN that has one or more convolutional layers becomes a CNN.

Dropout layers Introduced by Hinton et al. in 2012, dropout layers can be used in ANNs to prevent overfitting and achieve improved robustness against choices of network architecture [210]. Dropout layers refer to a defined percentage of randomly selected neurons being ignored in an epoch, removing all connections from and to other units temporarily. The approach was partly motivated or at least linked to the fact that more advanced organisms have evolved due to the breaking up of sets of co-adapted genes during sexual reproduction compared to asexual reproduction increasing the robustness of the remaining units over time [211]. An example is given in Figure 2.24.

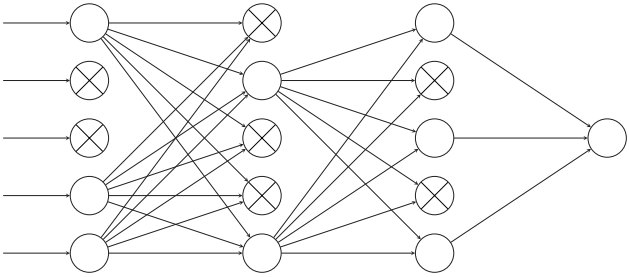


Figure 2.24: Model of a thinned dropout neural network, where crossed out single units are dropped. Based on [212].

One drawback of dropout is the two to three times increased training time due to its stochasticity. It mainly stems from training a different random architecture each time as the dropped out units change every iteration. As for every network, ANNs with dropout layers require extensive hyperparameter tuning with increased demands for network size, hence layers, higher learning rate, etc. [212]

Batch normalization Batch normalization, as proposed by Ioffe and Szegedy in 2015 [213], refers to the normalization of the input of every layer (and repeating so for every mini-batch) in a network as opposed to only the input layer. It can be included in backpropagation [213]. Batch normalization allows the use of higher learning rates, accelerating the training process up to 14 times [213], as one gets rid of the changing distribution of each layer's inputs. It also decreases the importance of careful choice of initial hyperparameters. While being comparable to the impact of dropout layers, for which they can serve as substitute, its additional regularization effect may eliminate the need of dropout layers alltogether in some cases. [213, 214]

Convolutional layers Convolution is continuously defined as follows:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(x)g(t-x)dx \quad \text{[215]}. \tag{2.118}$$

The convolution of two functions f and g can be understood as mirroring of function g along the y-axis, subsequently adding a time offset x and lastly sliding function g from $-\infty$ to ∞. It passes function f on the way. Wherever the two functions intersect, the integral of their product, so the area under function f weighted by function g, is evaluated. Convolution is commutative.

In the context of neural networks, convolutional layers, as fundamental component of any CNN, introduced by Le Cun et al. [216], are used to perform so-called feature extractions. They often consist of a combination of linear convolution and nonlinear activation functions. In this regard, a discrete definition of convolution is applied (changing the integral to a sum and working on finite sequences). It can be understood as a small array of numbers, called kernel or filter, sliding across the input tensor. By multiplying every entry of the kernel and the input element-wise and subsequently summing up these products, one obtains the so-called feature map (cf. Figure 2.25).

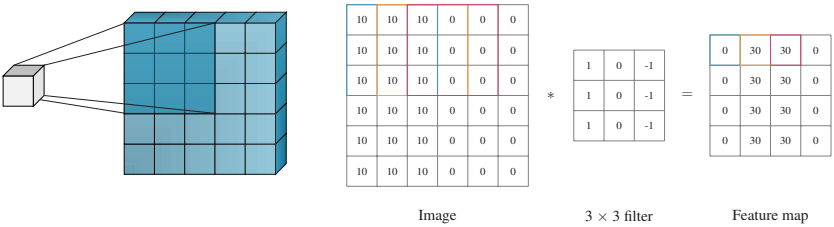


Figure 2.25: Schematic description of a convolutional layer in a neural network. A filter of a specific kernel size, in this case 3×3 is applied on the original image to obtain a so-called feature map. The feature map carries its name due to the fact that the most important features of an image are kept or even enhanced with this method, while the dimensions decrease compared to the original image.

Especially if this is repeated for different filter kernels, the characteristics or features of the input image are enhanced and captured. The output of the convolutional layer typically has decreased dimensions. In the case that this is unwanted, so-called padding, typically zero padding, which signifies the addition of rows and columns of zeros on each side of the input tensor, is applied in order to keep the same in-plane dimension throughout the convolution operation. If in contrast, the size reduction is desired, another parameter influencing the final output size, the stride parameter, comes in play, which defines the distance between two consecutive kernel positions. [217] While setting a stride larger than one can reduce overlap, it can also lead to checkerboard artifacts, especially if the size of the output window and kernel are not a multiple of the stride [218]. The alternative of applying pooling layers for the further reduction of output size will be discussed in the next paragraph.

The final output size can be calculated by

$$O = 1 + \frac{N + 2P - F}{S}, \tag{2.119}$$

with the image dimension $N \times N$, the filter dimension $F \times F$, stride S and thickness of padding P [219]. As an example, in the case in Figure 2.25, with $N = 6$, $P = 0$, $F = 3$ and $S = 1$, the output size of 4 results.

The output vector $\boldsymbol{h}^{(l)}$ of a hidden convolutional layer can be calculated analogously to the equations for MLPs and using the discrete definition of convolution (cf. Equation (2.103)) by

$$h_i^{(l)} = \sigma\left(\sum_{i \in M_j} h_i^{(l-1)} * W_{ij}^{(l)} + b_j^{(l)}\right), \tag{2.120}$$

with a selection of input maps M_j [220, 221]. The weights W_{ij} are updated through backpropagation and stochastic gradient descent just like the previously introduced fully connected layers [221] and the kernel values are the only parameters to be learned (while their size, padding, stride, etc. are hyperparameters that are set in advance) [217]. The inverse operation is used in some CNNs and also in deep convolutional generative adversarial networks (DCGANs); the layers are then called deconvolutional layer.

Pooling layers Pooling layers are used to decrease the in-plane dimensionality. They further introduce a higher translation invariance concerning small shifts and distortions in images, and decrease the number of subsequent learnable parameters. It shall be stressed that in contrast to a convolutional layer there are no learnable parameters in any of the pooling layers. [217] Some common, self-explanatory pooling operations that were used in the CNN in this work are depicted in Figure 2.26, Figure 2.27 and Figure 2.28.

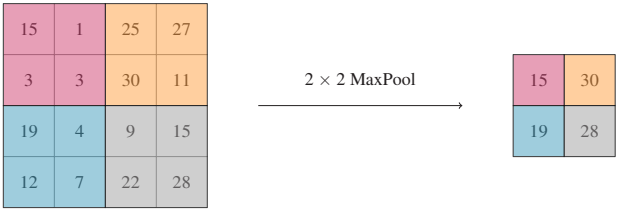


Figure 2.26: Depiction of a MaxPooling layer.

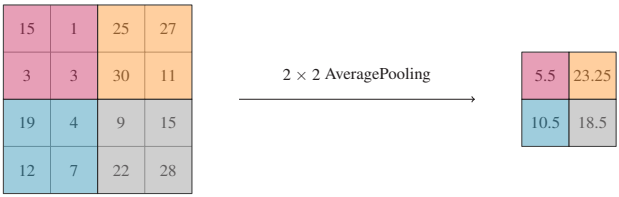


Figure 2.27: Depiction of an AveragePooling layer.

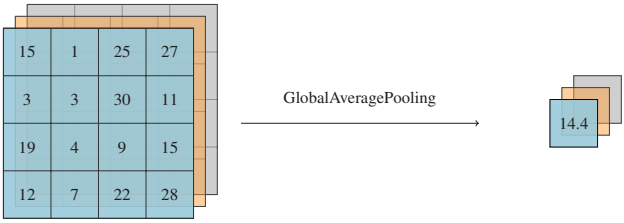


Figure 2.28: Depiction of a GlobalAveragePooling layer.

2.4.2 Generative adversarial networks

Generative adversarial networks, or GANs for short, were introduced by Goodfellow et al. [143] in 2014, and have since been considered one of the biggest findings in AI science [222], which is underlined by the over 78,000 citations of the original paper to date [223]. In 2020 alone, about 28,500 papers related to GANs were published, corresponding to approximately 78 papers every day, which are more than three per hour [222]. The main function of GANs is the generation of realistically looking images after being trained on a given training image data set [143]. Possible areas of application range from 3D object generation via medicine, pandemics (mostly during COVID19), image processing, face detection and text transferring to traffic control and many more [224]. In these application fields, the function of a GAN is not limited to the generation of images per se, but includes noise reduction, classification, detection of human motion, the generation of 3D images from 2D images, face recognition or super-resolution of images [224]. After a slow start, GANs have also established themselves in materials science. They have been used among others

for the inverse chemical design of materials [225, 226], data augmentation in microscopic images for material data mining [227], virtual microstructure design for steels [228], predicting the compressive strength of concrete (using tabular GANs) [229] or the generation of synthetic images of porous aluminum foam [230]. There is relatively little research on GANs in connection with FRP. Corresponding papers can be found on 3D inpainting of µCT images of glass fiber reinforced composites [231], fiber break analysis of unidirectional carbon fiber reinforced polymers by using super-resolution of synchrotron CT images [232], generation of 3D structures from two-dimensional slices of a variety of microstructures [233] and on the generation of realistic 2D transverse microstructures of unidirectional fiber reinforced composites [234]. The sources mentioned raise the hope that GANs also represent a serious option for microstructure generation in relation to discontinuous carbon fiber reinforced CT scans.

2.4.2.1 Basic theory

The Nash equilibrium The original idea of GANs derives from a central concept of game theory [235]. It is based on a two-player zero-sum game with two independently acting, non-communicating players trying to improve, while the gain of one player is exactly the loss of the other player resulting in a total sum of zero for the interests of both sides [235, 236]. In this optimization process, which is also called a minimax game [143, 235], the goal is to reach the so-called Nash equilibrium, which was shown to exist in any infinite game following these criteria by John Nash in 1950 [236]. At this equilibrium point, a one-sided change of strategy would be of no benefit to neither of the players.

Inspired by this concept, the traditional GAN as described by Goodfellow et al. [143] consists of two competing neural networks, one so-called *generator* and one *discriminator*. The generator makes an effort to capture the distribution of real, given training data and tries to create new samples that mimic the real ones. Meanwhile, the discriminator, which is often a binary classifier, aims at

differentiating the real samples from the generated samples as accurately as possible. Driven by their competition, constant improvement is required by both generator and discriminator in order to win the game [143, 237]. In the original paper, the process is compared to counterfeiters (generator) producing fake currency and police (discriminator) trying to detect the counterfeit currency [143]. Once the saddle point of the Nash equilibrium is reached, this adversarial training ends and the generator is considered to have correctly estimated the real data distribution [222]. In reality, reaching the Nash equilibrium is very challenging considering the objective functions being non-convex, the parameters being continuous and the parameter space being high-dimensional [238]. Nevertheless, outstanding results have been achieved with GANs, and they are particularly impressive due to the possibility of generating theoretically infinite data once properly trained.

The traditional GAN structure Figure 2.29 shows the structure of the basic GAN (also called vanilla GAN) as a flowchart. Therein, a generator G is fed with noise samples z from a prior defined noise function such as a Gaussian distribution and outputs generated samples $G(z)$. Those are provided alongside samples from the real data distribution to the discriminator D, which tries to classify the data by assigning a label of zero for data that it detects as generated and one for data that it detects as real. With this discriminator feedback and the actual true labels, a loss function can be calculated, based on which the weights of the generator and discriminator network are updated using backpropagation.

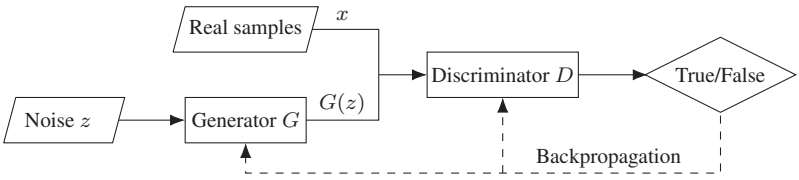


Figure 2.29: Flowchart diagram of the basic structure of the vanilla GAN as proposed by Goodfellow et al. Based on [235].

In the vanilla GAN, both generator and discriminator are normal MLPs with trainable parameters θ [143]. A value function $V(G, D)$ is used to evaluate the cost of the training. During the training process, only the parameters of one model are updated, while the parameters of the other are fixed [239].

The MLP representing the generator G is a differentiable function with parameters θ_{G}. In order to learn the distribution p_{g} over real data $\boldsymbol{x}$, a mapping of the input noise variables $p_{\mathrm{z}}(\boldsymbol{z})$ to the data space, $G(\boldsymbol{z}, \theta_{\mathrm{G}})$, is required. The second MLP representing the discriminator, $D(\boldsymbol{x}, \theta_{\mathrm{D}})$ outputs a single scalar. $D(\boldsymbol{x}) \in [0, 1]$ describes the probability that $\boldsymbol{x}$ came from the training data and not the distribution of the generator p_g. [143] Hence, the generator is trained to maximize the probability $D(G(\boldsymbol{z}))$ or to minimize $\log\left(1 - D(G(\boldsymbol{z})\right)$ leading to the following equation for the training of the generator, while the discriminator is fixed:

$$\min_{G} V(D, G) = \mathbb{E}_{\boldsymbol{z} \sim p_{\mathrm{z}}(\boldsymbol{z})}\left[\log\left(1 - D(G(\boldsymbol{z}))\right)\right] \quad \text{[143]}. \tag{2.121}$$

The expectation operator $\mathbb{E}$ denotes the expected value of the specific distribution function [179]. The parameters θ_{G} are updated by calculating and subsequently descending the stochastic gradients based on this equation. [143]

The discriminator is provided with samples from the unknown input distribution $p_{\mathrm{data}}(\boldsymbol{x})$ and the (at the training time of the discriminator) fixed generator samples. Hence, the value function is given by

$$\begin{aligned} &\max_{D} V(D, G) \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\mathrm{data}}(\boldsymbol{x})}\left[\log D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z} \sim p_{\mathrm{z}}(\boldsymbol{z})}\left[\log\left(1 - D(G(\boldsymbol{z}))\right)\right] \quad \text{[143]}. \end{aligned} \tag{2.122}$$

The discriminator parameters get updated analogously by ascending the stochastic gradient based on this equation.

Goodfellow et al. further proved that global optimality is only reached for $p_{\text{data}} = p_{\text{G}}$, so if the input data distribution matches the distribution learned by the generator. In this case, $D(\boldsymbol{x}) = \frac{1}{2}$ holds, which results in regression in the training process. [143]

Combining Equation (2.121) and Equation (2.122) results in the following minimax game for GAN optimization:

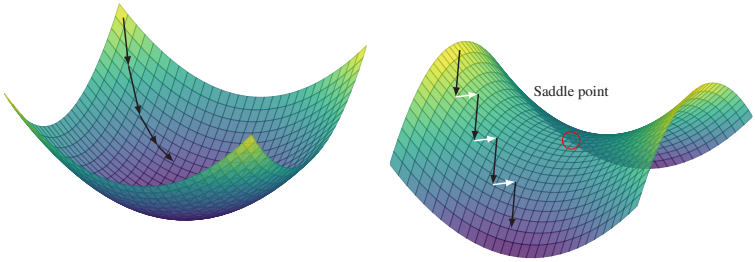
$$\begin{aligned} &\min_G \max_D V(D,G) \\ &\quad = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \left[\log D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})} \left[\log\left(1 - D(G(\boldsymbol{z}))\right)\right]. \end{aligned} \tag{2.123}$$

$\log\left(1 - D(G(\boldsymbol{z}))\right)$ gives the cross-entropy between $[0 \quad 1]^T$ and $[D(G(\boldsymbol{z})) \quad 1 - D(G(\boldsymbol{z}))]^T$. Proposed by Rubinstein [240] in 1997, the cross-entropy is a measure of the quality of a model for a probability distribution. As the name already implies, it uses the cross entropy or Kullback–Leibler divergence as a measure for the closeness of two sampling distributions [241]. Analogously, $\log D(\boldsymbol{x})$ gives the cross-entropy between $[1 \quad 0]^T$ and $[D(\boldsymbol{x}) \quad 1 - D(\boldsymbol{x})]^T$. Hence, the value function $V(G, D)$ is a Binary Cross Entropy (BCE) function, which is commonly used in binary classification problems [238]. In fact, any monotonically increasing function could be used instead of a logarithmic function [242].

After introducing the equations, the training procedure ought to have become evident: The discriminator receives a mini-batch of the generator samples and a mini-batch of the true samples. Then, the stochastic gradient of the discriminator equation is ascended to update the discriminator. The generator receives noise samples and the stochastic gradient of the generator equation is descended, and so on. A corresponding pseudocode can be taken from [143]. Theoretically, the discriminator could be updated k times and the generator only once per iteration, but in the original paper they use the least expensive and meanwhile most used option with $k = 1$.

If the discriminator is too good in early stages of learning, $\log(1 - D(G(\boldsymbol{z})))$ saturates. This issue can be solved by training the generator to maximize $\log D(G(\boldsymbol{z}))$ instead, which provides much stronger gradients in early learning. [143]

The already mentioned familiar difficulty in training GANs stems from the fact that due to the minimax game in Equation (2.123), the optimal weights for an adversarial network correspond to saddle points of the loss functions and not minima thereof, which is the case for conventional ANNs [242]. This issue is visualized in Figure 2.30.



(a) Standard ANN **(b)** Adversarial networks

Figure 2.30: Schematic illustration of gradient methods for a standard neural network (a) and adversarial networks (b). Classical loss functions are bounded from below, which means that following the gradients normally leads to a minimum, where the method is stopped. As opposed to this, loss functions of adversarial networks may be unbounded from below and the training alternates between minimization and maximization steps. If those two are unbalanced, the solution path might "slide off" the loss surface and never reaches the desired saddle point, making the algorithm unstable and resulting in a sudden "collapse" of the network. Based on [242].

The solution for the saddle point problem is determined using the alternating stochastic gradient method, changing between the descent and ascent step. Typically, the gradients are updated by an automated solver like the already introduced ADAM optimizer. [242]

2.4.2.2 Common challenges when training GANs

Apart from the general problem of non-convergence, there are specific typical challenges when training GANs, two of which will be elaborated on in the following paragraphs. Those issues mainly occur due to each model trying to negate the effect of the other one [243]. Solving one of the issues does not necessarily lead to success as these different challenges may occur parallel or at different stages of the training process [244].

Mode collapse The first common challenge is the so-called mode collapse, also called catastrophic collapse, which signifies the creation of very similar looking output data lacking variety by the generator [244]. Mode collapse occurs when the discriminator is stuck in a local optimum [244]. Typically, the generator can then produce one realistic data sample, which the discriminator recognizes as real, and will stick with it; sacrificing the goal of generating a data distribution similar to the training data distribution. Alternatively, only a particular, limited subset of modes is learned by the generator, which is sometimes called partial mode collapse [238, 245]. The discriminator function often shows sharp gradients around real data points in local equilibria [246] and hence the discriminator gradients point to similar directions for many similar points (discriminator overfitting) (cf. Figure 2.31). As the discriminator processes every generated image independently, there is no interaction between the gradients (also called catastrophic forgetting [247]). The generator is therefore not told to diversify its outputs. [248]

One possible solution to avoid mode collapse is the use of previously mentioned minibatch discrimination, where the discriminator is fed with multiple images at the same time rather than single images [248]. In addition, multiple further developments of GANs, like deep convolutional GANs [250], were proposed to tackle this issue by using different network structures, new objective functions or different training algorithms. Some important specific types are mentioned in Section 2.4.2.3.

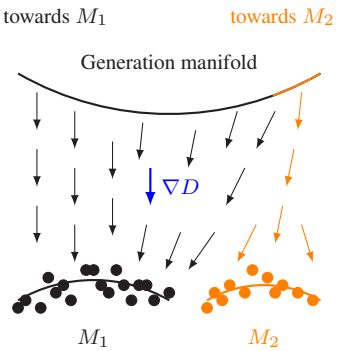


Figure 2.31: Simplified depiction of mode collapse with the generator gradients pushing to M_1 for most of the input noise z. Only if $G(z)$ is very close to M_2, the second mode is reached by the gradients. Provided that a corresponding z has a low probability in the noise distribution, the mode is visited rarely and the generator has little possibility to improve in this area. Based on [249].

Vanishing gradients If the discriminator becomes too good at its job of assigning zero to fake images and one to real ones, the generator gradients approach zero. This behavior is called vanishing gradients and prevents training progress. It typically appears when the discriminator is already close to the optimal solution. [238, 244] It can be addressed, e.g., by using the Wasserstein distance as an alternative method to calculate the loss of a GAN (see Section 2.4.2.3) [251].

2.4.2.3 Other types of GANs

In this section, a short, by no means comprehensive list of popular adaptions of the original GAN idea is presented.

First described by Arjovsky et al. [251], the Wasserstein GAN (WGAN) has the same structure as the vanilla GAN [143] but uses the Wasserstein distance, also known as Earth Mover distance, to calculate the generator and discriminator loss [235, 252]. It was intended to solve the problem of vanishing gradients [235], but can also itself show problems of bad image quality of the generated

samples or convergence failure. This was improved by Gulrajani et al. [253] by adding a gradient penalty, introducing the WGAN-GP.

DCGANs as introduced by Radford et al. [250] in 2016, do not only consist of fully connected layers, but obtain deconvolutional (more precisely known as transposed or fractionally strided convolutions) layers in the generator and convolutional layers in the discriminator. This is the kind of GAN that was used in this work as well. While the basic structure is the same, DCGANs typically include batch normalization in the hidden layers allowing for a deeper gradient propagation as the risk of generator collapse is decreased. In order to avoid sparse selection of gradients, ReLU activation is used in the generator (except for the output layer, which uses tanh) and Leaky ReLU activations are selected for the discriminator with the sigmoid activation in the output layer. [179]

Conditional GANs (cGANs) described by Mirza et al. in 2014 [254] condition the generator and discriminator on discrete auxiliary information by adding this information, normally a corresponding label, as another input layer. This enables the trained generator to output a result based on a requested label instead of generating random outputs. [254] In the case of continuous variables, the given additional input information can be divided into separate classes. The further development of a Continuous conditional GAN (CcGAN) by Ding et al. might be another option in that case [255].

Another extension of the cGAN would be the information maximizing GAN (InfoGAN) as introduced by Chen et al. in 2016 [256].

2.4.3 Neural networks in materials science

AI is making its way into all scientific fields; materials science is no exception. Apart from classical ML algorithms, which have been used for a while, neural networks are increasingly used and classified as DL at more complex scales (multiple layers). They can be used to predict mechanical properties [257], weld characteristics [258], crack propagation [259], and crystal structures [260] as

well as to classify surface defects [261], fracture surfaces [261, 262] and microstructures [263, 264], or to segment phase fractions [265], grain boundaries [266], or precipitates [267]. Deep learning can also play a role in materials science for object recognition [263], crack detection [25], feature extraction from transmission electron microscopy (TEM) or scanning electron microscopy (SEM) images [268], or the creation of digital twins [269]. Another technique is increasing image resolution through super resolution (SR) networks, which has been used sporadically in materials science [270], but is also emerging in commercially available software. The use of GANs in materials science was already elaborated on in the prior Section 2.4.2.

In this work, three neural networks are implemented and evaluated. One rather classical ANN for tensor interpolation, one CNN for the determination of FVC from 3D scans and one DCGAN for the generation of realistic 2D CT images.

3 Materials and Methods

The Methods chapter first discusses the material and the CT scans, which serve as the basis for all investigations in this thesis. The methods used to examine the plastificates are then explained. This is followed by descriptions of the methods for determining fiber volume content, fiber length distribution and fiber orientation distribution. After the interpolation of the fiber orientation tensors has also been explained, the chapter concludes with an elaboration of the generative adversarial network used for artificial image generation.

3.1 Materials and processes

In the following, the material constituents are presented, as well as the process used to produce the final composites, the sizes and extraction points of the specimens and the considerations behind these.

3.1.1 Polyamide 6

PA6, sometimes also referred to as polycaprolactam, Perlon or Nylon 6, is a thermoplastic, semicrystalline polymer from the group of polyamides. The number 6 was assigned because of the six carbon atoms between the nitrogen atoms in the molecule chain. The chemical formula is $(C_6H_{11}NO)_n$. In contrast to other polyamides, which are mostly formed by condensation, PA6 is synthesized by ring-opening polymerization of ε-caprolactam (compare Figure 3.1).

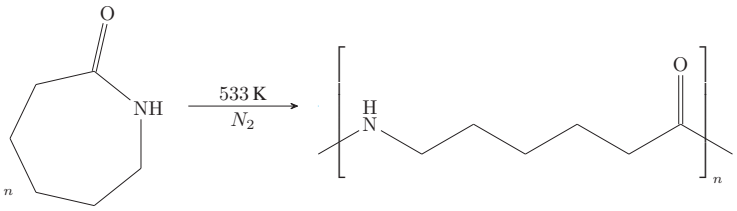


Figure 3.1: Schematic depiction of the synthesis of polyamide 6 by ring-opening polymerization of ε-caprolactam. The ε-caprolactam is heated at about 533 K - 553 K in an inert atmosphere of nitrogen for multiple hours until the ring breaks and polymerization takes place. Small amounts of water are initially required, which is why this synthesis process is also known as hydrolytic polymerization. However, alternative synthesis processes are also possible. Based on [271].

PA6 has a melting temperature of approximately 222 °C, a glass transition temperature between 50 °C and 60 °C and a density of 1.14 g/cm^3. With 3 % moisture absorption at 50 % relative humidity and even 9.5 % moisture absorption in water, PA6 can uptake particularly high amounts of water. [271] Water acts as a plasticizer in PA, decreasing the glass transition temperature by about 50 °C between completely dry and maximum water absorption [272]. In addition, the shear modulus and modulus of elasticity depend on the temperature, as typical for thermoplastics. [272] These dependencies on climatic and environmental conditions exacerbate the work of engineers when designing with PA6 and characterizing the material behavior of it.

In this specific work, the PA6 matrix material used was the TECHNYLSTAR XS 1352 BL NATURAL with a master batch. It was generously provided by the company DOMO Chemicals GmbH, Leuna, Germany.

3.1.2 Polycarbonate

In order to take into account the differences that the choice of matrix system has, in particular on the structure of the plastificate, polycarbonate is sometimes used as a reference system in this thesis. The polycarbonate material is a Makrolon 2405 PC provided by Covestro AG, Leverkusen, Germany.

3.1.3 Carbon fibers

Carbon fibers are a long and thin (usually about 5 µm - 7 µm in diameter) material composed mostly of carbon atoms. They owe their high strength and Young's modulus to their graphite-like arranged carbon structure in planes that get aligned parallel to the fiber direction during manufacturing. When producing carbon fibers, the raw material, also called precursor, is in most cases polyacrylonitrile (PAN), or more rarely rayon or pitch, which all exhibit long strings of carbon bound molecules. After thermally stabilizing the long strands of precursor by heating the fibers in air at 200 °C - 300 °C, the so-called carbonizing step follows. While heating the fibers for several minutes at a temperature between 1,000 °C and 3,000 °C in an atmosphere without oxygen, the non-carbon atoms leave the fibers. After treating the surface to slight oxidization, the fibers are finally coated in a material compatible with matrix material. This process is also called sizing. [273]

For this work, high-tenacity (HT) carbon fibers PX 35 with sizing for PA6 were acquired from ZOLTEK Corporation, St. Louis, United States. In the respective data sheet, the manufacturer states a carbon content of 95 %, a density of 1.81 g/cm^3, a diameter of 7.2 µm, a Young's modulus of 242 GPa, a tensile strength of 4,137 MPa and an elongation at break of 1.7 %. The tex number is 3750.

3.1.4 Glass fibers

The glass fibers were provided by Johns Manville Corp, Denver, United States. The type is StarRov ® 895 2400 with a tex number of 2400. However, they will only play a subordinate role in this work, which is why no further mechanical parameters will be given.

Clarification of material use per method

It shall be made clear at this point that the main investigation material in this work is carbon fiber reinforced polyamide 6. Only in individual cases, which are broken down in more detail below, is another fiber or matrix material used for comparison. In the case of the plastificate examinations, in order to better understand the peculiarities of the CF-PA6 material, such as the increased porosity, GF-PA6 and GF-polycarbonate are also included. In the fiber volume content investigations, only CF-PA6 is used, as there is hardly any potential for developing methods for material combinations with glass fibers, since the FVC determination there works reasonably reliably even in a non-destructive manner. Specimens from Plate 2 and Plate 3 are used, which are shown in Figure 3.4. The investigations are published in [188]. As far as fiber orientation determination is concerned, CF-PA6, GF-PA6 and hybrid CF-GF-PA6 are investigated. Specimens from Plate 3 (CF-PA6), Plate 4 (GF-PA6) and a hybrid CF-GF-PA6 plate with an identical cutting pattern, which is not shown explicitly in this dissertation, were used. The method for GF-PA6 has already been implemented by Bertram and Pinter [79], the slight adaptations that are necessary for the other two material types are explained in the respective Section 3.6 in the Methods. With regard to tensor interpolation, only CF-PA6 is initially considered, as the various methods were assessed on this basis. Only specimens from Plate 1 (Figure 3.4) were used. These results are published in [17]. The determined orientation tensors of CF-PA6, GF-PA6 and hybrid CF-GF-PA6 and their interpolation were considered for the use in Mori-Tanaka and Halpin-Tsai and the respective comparison with the experimental values. The samples and plates correspond to those mentioned in the context of fiber orientation determination. These results are published in [274]. For the GAN, only CF-PA6 samples were used again, namely from Plate 3 and another pure CF-PA6 plate, which is not printed, with an identical cutting pattern, but with the plastificate inlay rotated by 180°. This decision was due to the fact that in the course of the preparation of this work, the discovered skewed flow front was identified as a process error and not as a desirable process property to be

depicted. By using a plate with an inverted plastificate insert, the training data set could be somewhat neutralized in this respect. These investigations were published in [275].

3.1.5 The LFT-D manufacturing process

The long fiber thermoplastic direct (LFT-D) process is depicted schematically in Figure 2.1. The polyamide 6 granulate is compounded in an extruder along with the master batch at temperatures between 260 °C - 280 °C. In a second, mixing extruder with a twin screw, the fiber rovings are introduced to the polymer melt uncut and are chopped up irregularly by the shearing motion of the extruder. The resulting initial charge, also called plastificate in this work, of which two kinds can be seen on top in Figure 3.2(a) and Figure 3.2(b), is ejected through a specific nozzle. Subsequently, it is directly inserted in a press (Dieffenbacher), preventing a reheating of semi-finished products, which is necessary in other known LFT production processes.

In the press, the plastificate is transformed into a plate or part (depending on the inserted mold) and simultaneously cooled down (cf. Figure 3.3) in a so-called compression molding process. The press has a temperature of about 80 °C - 90 °C. The mold for the plates commonly used in this work has dimensions of 400 mm $\times$ 400 mm and the height of the plates was fixed at 3 mm using spacers. An image of both front and back of these plates can be seen on the left in Figure 3.3. The insertion area of the plastificate at the left is clearly visible.

It shall be noted that both the amount and geometry of the plastificates as well as the insertion position in the mold severely influence the mold filling behavior. This in turn influences the macroscopic quality of the component, but in particular also the microstructure and therefore anistropies and weak points in the mechanical properties. These investigations are not part of this work and are carried out by process developers and process simulants. However, the aim of this work is to be able to draw conclusions about the behavior in the process and in subsequent mechanical tests through targeted characterization

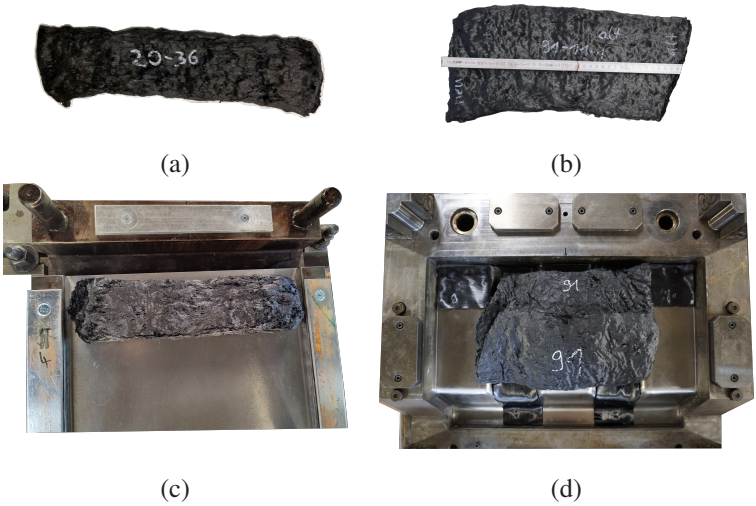


Figure 3.2: Images of (a) an initial charge/plastificate (340 mm × 75 mm × 35 mm) for the plate production with its (c) insertion position in the 400 mm × 400 mm mold, as well as (b) an initial charge/plastificate (275 mm × 175 mm × 29 mm) for the "MaiQFast" part (automotive underbody segment with stiffening ribs) and its (d) insertion position in the "MaiQFast" part mold (two overlapping inserted plastificates in the middle, and three tapes, e.g., on the upper side, for additional continuous reinforcement). The scale from image (b) does not apply to all images, so the corresponding dimensions are given in this caption.

and quantitative reproduction of the microstructure in the plates. For the plates used in this work, one single, longer plastificate of about 340 mm length, 75 mm width and 35 mm height was inserted at the left of the 400 mm × 400 mm mold (see left side of Figure 3.2). It shall be added that the dimensions of the plastificates given in the caption of Figure 3.2 are theoretical values, from which the actual plastificate geometries sometimes deviated significantly, especially the wider ones used in the MaiQFast production.

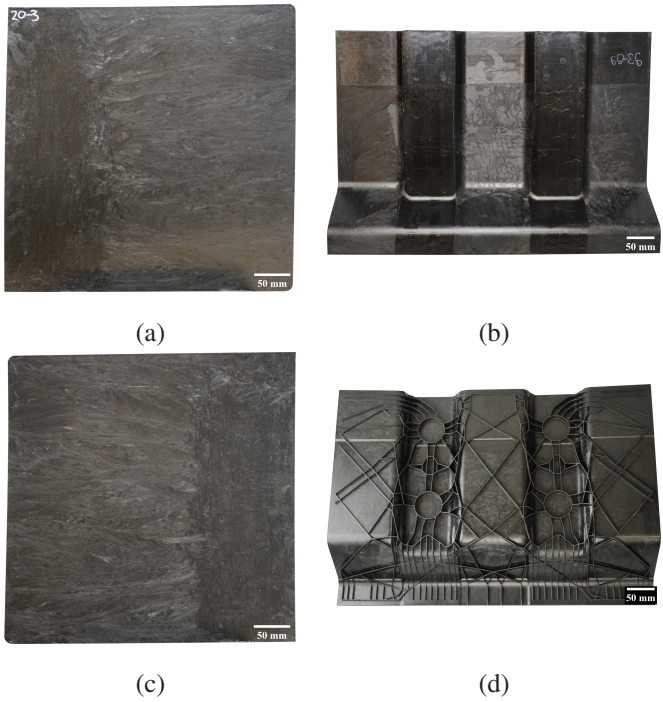
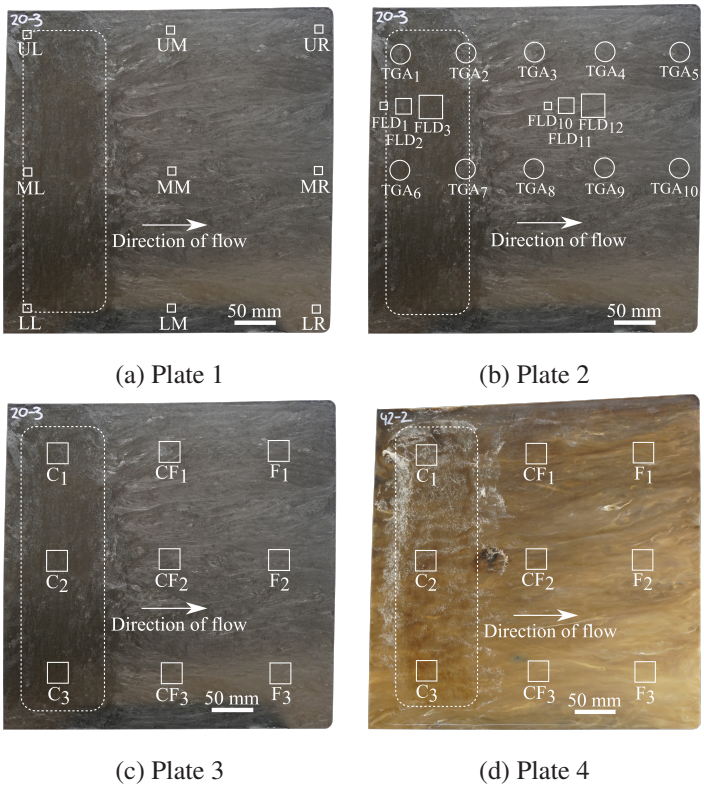


Figure 3.3: Images of the front (a) and back side (c) of a 400 mm × 400 mm × 3 mm CF plate used in this work after the press process and the front (b) and back side (d) of an additionally continuously reinforced "MaiQFast" part with dimensions of 600 mm × 400 mm after the press process.

3.1.6 Preparation of specimens

In order to examine the material in the CT device, smaller specimens were cut from the material through waterjet cutting. The size and position of the specimens was of particular interest. Regarding the position of the specimens, the differences between the area where the initial charge is inserted (often labeled C), the transition area (CF) and the area dominated by the material flow (F) were of particular interest. Therefore, in most cases, three specimens were taken from each of these areas as can be seen in Figure 3.4 (a), (c) and (d).



(a) Plate 1 (b) Plate 2

(c) Plate 3 (d) Plate 4

Figure 3.4: Different cutting plans to extract specimens of different sizes and at different locations by waterjet cutting superimposed on an image of a CF plate ((a), (b) and (c)) and GF plate (d).

Considering the dimensions it was evident that the samples should be small, so that a high resolution in the CT images could be reached, but large enough, so that not only very local effects are depicted. Several considerations influenced the choice of specimen size for a particular application/evaluation, so that partly different specimen sizes were used for different methods (cf. Figure 3.4). As far as the first question regarding the necessary resolution is concerned: 3 voxels - 4 voxels across the fiber diameter, i.e. a resolution of 1 µm - 2 µm, would be necessary for single fiber detection. This is not compatible with a

reasonably justifiable specimen size and is also not inevitable, as algorithms that do not rely on single fiber segmentation are possible. The next aspect of a representation of the microstructure that is not too localized is more difficult to grasp. Therefore, it seemed sensible that after cutting a specimen to size, the stochastic fiber length distribution in the plate resulting from the process should still be mapped as representatively as possible. To investigate the effect of specimen size on the FLD, three specimens measuring 10 mm $\times$ 10 mm, 20 mm $\times$ 20 mm and 30 mm $\times$ 30 mm were taken from Plate 2, both in the charge area and in the flow area (see Figure 3.4). The fiber lengths of these samples were determined experimentally and can be seen in Section 4.4. While there were occasional fiber lengths of significantly more than 10 mm, a peak at about 0.5 mm, a median length of about 0.48 mm and an average (arithmetic mean) length based on number of about 1.07 mm clearly indicate the occurrence of the clear majority of fibers in the range below 2 mm (see Figure 4.15). Based on this consideration, an initial specimen size with a side length of 10 mm could be justified for the development phase of the tensor interpolation problem, because only very few fibers would be decisively cut off (see Plate 1 in Figure 3.4). For all subsequent investigations, however, larger samples with side lengths of 25 mm (see Plate 3 and Plate 4 in Figure 3.4) were chosen for several reasons, which are briefly summarized hereinafter. Firstly, quite extreme cases of anisotropic tensors with very different local orientations were deliberately desired for the development of the interpolation methods, as this would ensure that the method would work in the worst case scenario. This was not the case for all further evaluations, where a realistic representation of the microstructures to be expected in such a plate was required. Furthermore, the average fiber lengths of the differently sized test samples increased from the smallest to the middle sample in both cases, but did not show a definitive trend from the middle to the largest sample (see Figure 4.16). Thus, a saturation of the effect of the sample size on the fiber length distribution between a side length of 20 mm to 30 mm was assumed. In addition, the fiber volume contents of the test samples were also determined via acid based dissolution (cf. Section 3.4.1) and they increased with increasing sample size as well (cf. Section 4.3.1). The available crucibles for the thermogravimetric analysis with the

commercial system LECO TGA801, as an alternative determination of FVC through pyrolysis, had a diameter of 25 mm. For all these reasons, a side length of 25 mm was subsequently considered appropriate.

In this work, apart from the purely CF reinforced plates (cf. Figure 3.4 (a), (b) and (c)) and the GF reinforced plate (cf. Figure 3.4 (d)), a hybrid reinforced plate with both carbon and glass fibers was also considered sporadically; especially in the context of reliably extracting the two separate fiber orientations in CT data of a plate like this.

3.2 CT scans

The CT device, the process of creating CT scans and the parameters used for them are explained briefly below.

3.2.1 Device

Most CT scans in this work were done with the institute's (Institute for Applied Materials at Karlsruhe Institute of Technology) own CT device. It is a YXLON-CT (Yxlon International CT GmbH, Hattingen, Germany) precision µCT system with a µ-focus X-ray transmission tube with tungsten target and a PerkinElmer (Waltham, MA, USA) Y.XRD1620 flat-panel, quadratic 2048 pixel (px) $\times$ 2048 pixel (px) detector.

During a repair of the in-house CT, another device was used at the wbk Institute of Production Science at KIT. This device was a Zeiss Metrotom 800 cone beam µCT system with the flat panel detector PaxScan2520V with 1536 px $\times$ 1920 px, so a rectangular detector contrarily to the other one used.

3.2.2 Procedure

After mounting the specimen(s) and installing them in the beam path, appropriate parameters were selected for the scan (see Section 3.2.3). After the scan procedure is done, the respective 2D projections generated by rotating the sample in the beam path were reconstructed to a volumetric image applying the Feldkamp cone-beam algorithm [154]. Subsequently, the reconstructed scans were processed in VG Studio Max 3.4.2.

3.2.3 Parameter settings

Different parameters were chosen for different scans used in this work. The parameters of some of the most important scans (the four main plates and specimens referenced in Figure 3.4) are summarized in the following Table 3.1.

Table 3.1: Scan parameters of the different plates. The rectangular specimens of Plate 2 were scanned on the Zeiss device, all others on the Yxlon device.

Parameter	Unit	Plate 1	Plate 2	Plate 3	Plate 4
Voltage	kV	150	100	110	125
Current	mA	0.25	0.16	0.13	0.12
Voxel size	µm/voxel	8.57	25.98	17.39	19.17
Linebinning parameter	-	2	-	2	2
Number of projections	-	1950	1450	2220	2100
Exposure/Integration time	ms	500	1000	800	1000

3.3 Investigations of the initial charge/plastificate

It appears obvious that the microstructure of the final plates is particularly dependent on the initial structure of the inserted plastificate. For this reason, a CT image-based investigation of the microstructure of the plastificate was initially of interest in order to understand the overall flow and material behavior in the pressing process.

3.3.1 Initial orientation state

First of all, the initial fiber orientation in the plastificate appears to be particularly decisive for the fiber orientation and microstructure of the plate. It could basically be determined using the same procedure for determining the fiber orientation tensors as for the samples from the plates, which is presented in Section 3.6 and which is based on the principles explained in Section 2.2.3.2. Therefore, CT images were acquired from plastificates, which can be seen for CF-PA6 in Figure 3.5 and for GF-PA6 in Figure 3.6.

The two swirls in the center of the plastificate can be clearly seen in image (a) of both figures. However, no real distinction between fiber and matrix is possible, especially with the CF plastificate, but also with the GF plastificate. This is due to the poor resolution, which is required by the size of the plastificate, and the fact that there is a lot of air in the material mixture, which has such a clear difference in gray value that the small difference between fiber and matrix is hardly recognizable. A classic fiber orientation analysis is therefore hardly possible. However, the gradient-based method can still be used and simply determine the gradients between air and material. This would then determine the overall material orientation. One can make the reasonable assumption that the fibers move along the matrix flow and should therefore be oriented in a similar or the same way as the overall material.

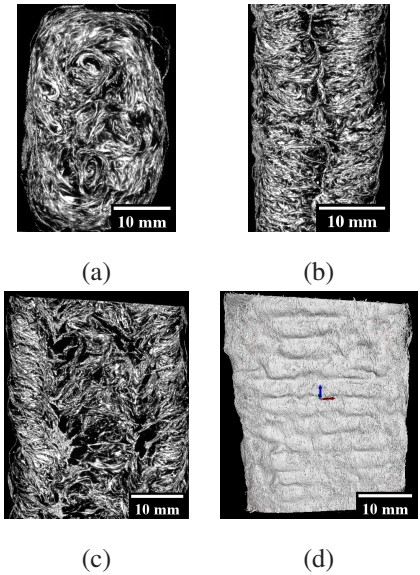


Figure 3.5: Upper (a), right (b), front (c) 2D section view and 3D view (d) of a CF-PA6 plastificate. It was done with the Zeiss Metrotom 800 cone beam µCT system at wbk Institute of Production Science at KIT. The scan had a resolution of 73 µm/voxel and was also used in the author's publication [276].

Accordingly, grids were placed over the plastificates (see Figure 3.7) and the material orientation tensors of the cubes were evaluated applying the method used to determine the fiber orientation tensors (see Section 3.6).

3.3.2 Porosity and volume determination

Fiber orientation measurements in the plate revealed that the flow front in the process appears to be slightly skewed. After ruling out various causes, the somewhat irregular geometry of the plastificates and their porosity distribution were taken into consideration.

Therefore, the images were first loaded into Python (read in as .raw image file) and then converted to numpy arrays with the correct dimensions. Subsequently,

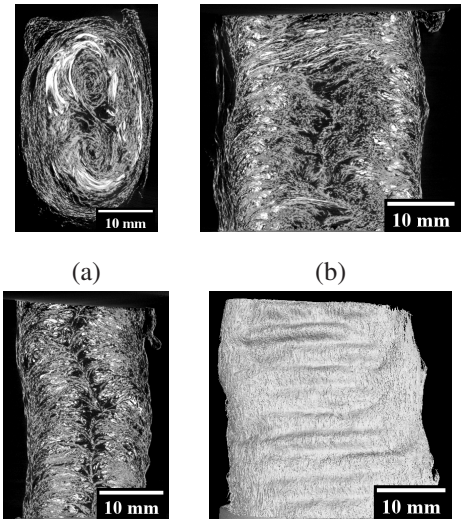


Figure 3.6: Upper (a), right (b), front (c) 2D section view and 3D view (d) of a GF-PA6 plastificate. It was done with the institute's own YXLON CT device. The scan had a resolution of 77.5 µm/voxel.

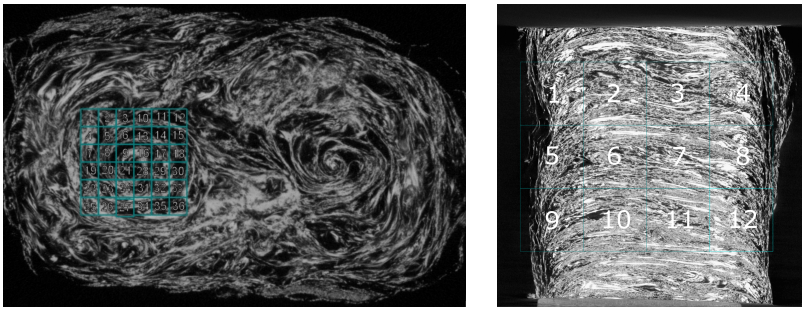


Figure 3.7: (a) Image of a CF plastificate scan section with a superimposed grid of 36 3.33 mm × 3.33 mm × 3.33 mm cubes around the left swirl. (b) Image of a GF plastificate scan section (different plane than in (a)) with a superimposed grid of 12 20 mm × 20 mm × 20 mm cubes along the extrusion direction.

the image was made binary with the Otsu threshold applied to each slice. Afterwards, the segmentation of the plastificate area and the pores inside this area had to be implemented. This is difficult insofar as the area of the plastificate that is penetrated by pores cannot be delimited by thresholding methods, as air is on the outside and inside, and also cannot be delimited by region growing approaches or similar due to its branched shape. The solution is morphological approaches. These apply a structuring element, which can have any shape or size, usually a square or circle, to an input image and generate an output image. These have the advantage that they cannot only remove noise but isolate individual elements and especially join disparate elements in an image, which is needed to capture the plastificate as a whole. The most basic and well-known morphological operations are dilation and erosion, which were both used in this application and are explained in the following. Numpy [277], the OpenCV cv2 [278] library and Matplotlib [279] were used for these operations and the creation of plots.

Dilation

The image I is convolved with a kernel X. The kernel is set with the command *cv.getStructuringElement* and is chosen to be of elliptic shape in this case. The ellipsis is defined with the function *cv.MORPH_ELLIPSE* as a circle with radius of the kernel size 15 - again in this particular work. The center of the kernel is typically the anchor point of the kernel, which is slid across the image. For the dilation operation (*cv.dilate*), the anchor point pixel value is replaced by the maximal pixel value captured by the kernel:

$$(I \oplus X)(x, y) = \max\{I(x + s, y + t) + X(s, t) \,|\, (s, t) \in D_x\}, \qquad (3.1)$$

with D_x as the definition domain of the structuring element. It is in that sense similar to a maximum filter with the difference of the important choice of

special structuring elements, changing the outcome decisively. Dilation causes the bright regions within an image to grow (cf. Figure 3.8), hence the name.

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

Figure 3.8: (a) Binarized slice of the original scan. (b) Same slice after applying dilation. The entire plastificate has become white and the borders are dilated. (c) Same slice as (a) but after applying erosion. The entire image has nearly become black. (d) Slice with successive application of dilation and subsequent erosion. The innards of the plastificate are still completely white, but the extended borders are reduced by the subsequent erosion. Segmentation of the plastificate is possible.

Erosion

Erosion (*cv.erode*) is like the antagonist of dilation computing a local minimum across the area of the given kernel:

$$(I \ominus X)(x,y) = \min\{I(x+s, y+t) - X(s,t) \mid (s,t) \in D_x\}. \text{[280, 281]} \quad (3.2)$$

It can be seen in Figure 3.8 that by applying dilation to the scan image, the pores can be filled in. However, the edge also expands beyond the edge of the actual plastificate. A subsequent erosion operation can reverse this expansion of the edge. The result is an image which (apart from small errors at the edge) has segmented the plastificate quite well.

Consequently, the difference between the image with the completely white plastificate (d) and the original image (a) can be used to generate an image in which only the pores are white. By counting all pore pixels and all pixels that are assigned to the plastificate and dividing them by each other, the proportion of porosity per slice is obtained. In addition, by dividing the pixels assigned to the material by the proportion of the total pixels of the scan, the proportion of area of the plastificate can be determined. Both of these aspects are shared in the results and then discussed.

3.4 Determination of fiber volume content

In the following, the procedure of the experimental and then the two computational methods for determining the fiber volume content is presented. Parts of this have already been published in a paper [188], which is marked accordingly.

3.4.1 Chemical dissolution [1]

In order to chemically remove the matrix, approximately 50 ml of concentrated sulfuric acid is added to the samples in an Erlenmeier flask which is then placed on a hotplate (cf. Figure 3.9).

This is followed by heating until smoke is produced. Subsequently, the samples have to react for one hour at this temperature. After the samples have cooled down ("lukewarm"), they are mixed with approximately 25 ml of a 35 % hydrogen peroxide solution and heated until the solution becomes clear and no more gas bubbles rise.

[1] This section is extracted from the author's publication [188] with only slight linguistic changes.

(a) (b)

Figure 3.9: (a) Drop of a specimen in sulfuric acid in Erlenmeier flask and (b) dispersion of the detached fibers. Courtesy of FIBRE Bremen. Extracted from Blarr et al. [188].

The remaining fibers are put in a new specimen cup and can be used for the fiber volume content determination via weighting and for the fiber length distribution determination.

The investigations were done at FIBRE (Bremen). For Plate 2, directly calculated FVC values by FIBRE were taken as result, for the specimens of Plate 3, the calculation of the FVC via the fiber mass content and the respective densities was performed afterwards.

3.4.2 Computational methods

This section explains the two methods for determining the FVC using image evaluation methods after the problems with existing methods have been addressed again.

3.4.2.1 Application and shortcomings of common techniques [2]

The through-thickness fiber concentration analysis by Gandhi et al. was mentioned in the State of the Art [5]. However, it was impossible to apply this procedure to the µCT scans in this work as the first step of choosing the midpoint threshold is not feasible for the histograms of the CFRP scans as will be

[2] This section is extracted from the author's publication [188] with only slight linguistic changes.

elucidated in the next section (Section 3.4.2.2) and can be seen in Figure 3.12 (only one peak is visible). Conventional automatic thresholding methods were tested as comparison to the methods introduced in this work, which can be seen in Table A.1 and Table A.2 in the Appendix (Section 9). Therefore, the two common automatic thresholding methods Otsu (*opencv* [278]) and mean (*scikit image* [282]) were applied, once without filtering the image before and once with the best-performing filter option of our self-implemented method (medianBlur with a kernel size of 15 for Plate 2 and 23 for Plate 3, respectively) applied beforehand. The minimum threshold by *scikit image* did not even compile with a "RuntimeError: Unable to find two maxima in histogram", which confirms the previous findings. While the two thresholding procedures that worked were applied in Python supporting the subsequent further processing of the values, the exact same threshold and filtering methods can be applied in ImageJ as well. The results were far away from the experimental values. It is noticeable that the results of the pure threshold methods deviate on average by almost 100 % relatively compared to the experimental results. It is particularly striking that the calculated FVC values are almost the same for each sample. A purely constant shift of the threshold value therefore does not appear to make sense, as this would also not cover the differences between the samples. The use of the median blur at least significantly improved the results with the Otsu threshold. It seems that in the case of low contrast CT images of composites with high fiber volume content, simple thresholding is insufficient for fiber segmentation, which is supported by literature [283].

3.4.2.2 "Average or above" (AOA) thresholding [3]

The novel thresholding method is realized in Python 3.8.7 with the help of the SimpleITK [284–286], the OpenCV [278] and the NumPy [277] libraries among others. The scans of the samples are generated as 16 bit 3D images in the .raw and .mhd file format. For all further steps those scans were converted

[3] This section is extracted from the author's publication [188] with only slight linguistic changes.

into 8 bit. Dark slices at the borders resulting from the specimens not being exactly even-surfaced and further image errors were cut. Each loaded scan was converted into a 3D array. In the following, every slice is handled separately; so it was iterated through the thickness of the samples and worked on 2D images. At first, a filter was applied to reduce the noise. The filters tested include the "normal" blur filter, the median blur filter, the Gaussian blur filter and the bilateral filter each with various kernel sizes. The kernel size defines the dimension of the window that is slid across the image and in which the filter-specific calculation is performed. The performance of the filters was judged afterwards by comparing the resulting calculated fiber volume contents with the experimental values. The results of the specimens of the Plate 2 can be seen in Table A.3, Table A.4 and Table A.5 and the results of the specimens of the Plate 3 in Table A.6 and Table A.7 in the Appendix (Section 9). The median filter with resolution-adapted kernel size performed the best. It works by creating a kernel of pixels around a central pixel. The values are sorted and the central pixel gets replaced by the median value. From the then noise-reduced image, a threshold value was determined by using the Otsu algorithm. The Otsu algorithm separates an image in two sections by maximizing the inter-class variance of the gray-level intensities between those sections:

$$\sigma_{\mathrm{B}}^2(t) = \omega_1(t) \cdot \omega_2(t) \left(\mu_1(t) - \mu_2(t)\right)^2 . \tag{3.3}$$

t is the gray value of the threshold being searched for and i is the run variable with $i \in [0, L]$. ω_1 and ω_2 are the probabilities of the two sections with

$$\omega_1(t) = \sum_{i=0}^{t-1} p(i) \quad \text{and} \quad \omega_2(t) = \sum_{i=t}^{L-1} p(i), \tag{3.4}$$

while $p(i)$ represents the probability for each gray level intensity.

$\mu_1(t) \cdot \omega_1(t)$ represents the mean intensity value of the first section (and accordingly for the second section) with

$$\mu_1(t) = \sum_{i=0}^{t-1} \frac{i \cdot p(i)}{\omega_1(t)} \quad \text{and} \quad \mu_2(t) = \sum_{i=t}^{L-1} \frac{i \cdot p(i)}{\omega_2(t)}. \tag{3.5}$$

After calculating the Otsu threshold T for each slice, it is plotted over the thickness. In Figure 3.10, the plot for specimen $F_{1,\mathrm{carbon}}$ can be seen as an example.

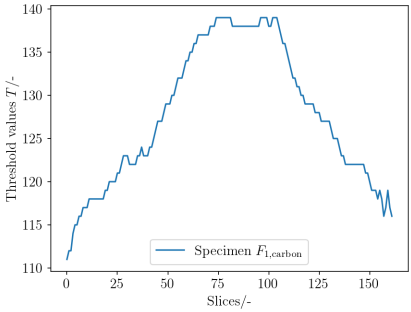
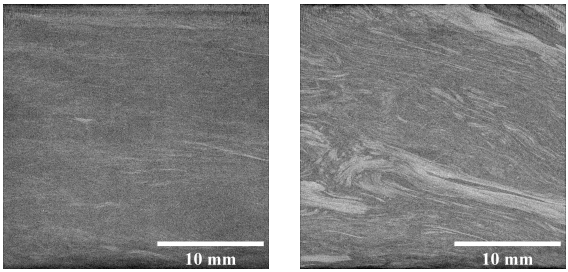


Figure 3.10: Example of the threshold values from sample $F_{1,\mathrm{carbon}}$. Based on Blarr et al. [188].

Considering the course of the threshold values over the height of the sample, it is noticeable that the threshold values are the highest in the center and drop to a much lower level at the borders of the specimen. Inspecting the scans slice per slice, it is noticeable that there are less fibers visible in those border areas compared to the central layers of the sample as can be seen in Figure 3.11.

This phenomenon is due to the material flow in the compression molding process. The process-induced difference between outer layers and the center of the part is called shell-core effect in injection molding vocabulary and can be detected considering both fiber volume content as well as fiber orientation [5, 17]. However, this would mean that there are more lighter gray values representing the fibers and less darker ones representing the matrix in the center



(a) (b)

Figure 3.11: First slice (a) and center slice (b) of the scan of the specimen $F_{1,\mathrm{carbon}}$. The low amount of fibers and fiber bundles at the border (a) and the contrary high amount in the center (b) is clearly visible. Extracted from Blarr et al. [188].

layers, but the threshold value distinguishing between the two peaks should not shift. This therefore cannot be the main reason for the course of the threshold values over the thickness.

A second possible explanation would be that of the beam hardening effect, a common phenomenon in computed tomography. The further the beam penetrates the material, the higher the average energy of the photons, as the low energy photons get scattered easily. However, uncorrected images typically show increasing gray values towards the center, hence the rotation axis of the CT. Consequently, this effect would be contrary to the one observed. Additionally, multiple specimens were scanned occasionally, which would superimpose this effect on multiple samples.

Considering the histograms, one understands the issue more. In Figure 3.12, exemplary histograms of specimen $F_{1,\mathrm{carbon}}$ are given.

In Figure 3.12(a), the histogram of the entire specimen is shown, in Figure 3.12(b), one can see the histogram of one single slice rather towards the border of the specimen and in Figure 3.12(c), the histogram of a single slice in the center of the specimen is given. In the entire histogram in Figure 3.12(a), it is apparent that there are not two peaks as expected. To the contrary, all voxels seem to show gray values roughly fitting to one single normal distribution. This

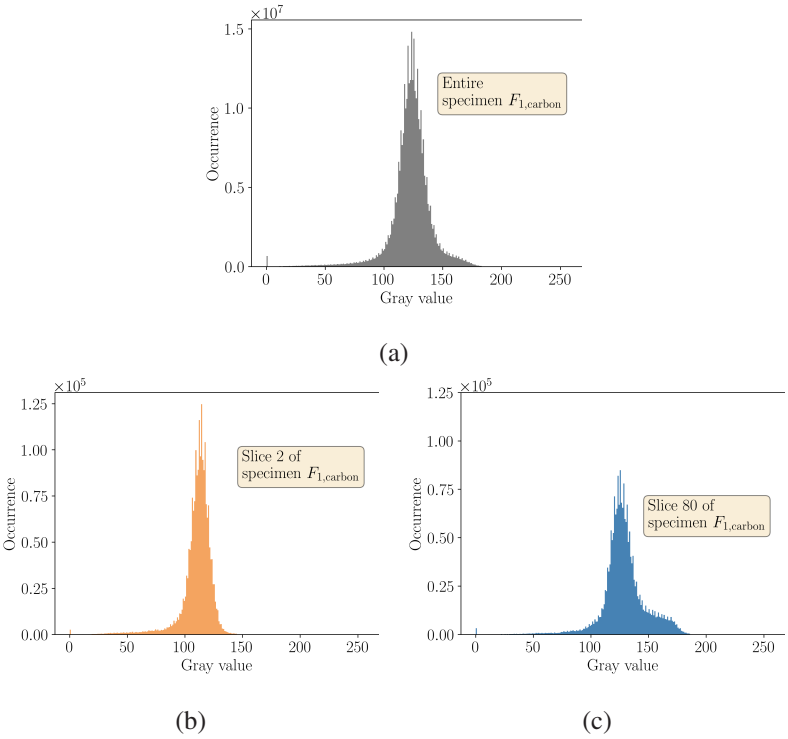


Figure 3.12: Comparison between the histograms of the entire specimen $F_{1,\mathrm{carbon}}$ (a), a border (b) and a middle slice (c) (after having applied the median filter). $(b) \subset (a) \wedge (c) \subset (a)$. Based on Blarr et al. [188].

is due to the bad contrast between carbon fiber and polymer in the CT, which has already been mentioned before and can be seen, e.g., in Figure 3.16(b), induced by the closeness of the densities. Additionally, much higher resolutions would be necessary to at least come close to resolving single carbon fibers, which is not given with the resolutions of these scans. Both of these facts lead to very noisy images. The Otsu thresholding, respectively any thresholding method for that matter, therefore can not work the way it is supposed to, but calculates some kind of median value of the entire gray value distribution. Looking at the histogram of the border slice (Figure 3.12(b)), the peak intensity is slightly shifted to the left

and there is a small shoulder visible at the left of the distribution. Comparing it with the histogram of the center slice (Figure 3.12(c)), the distribution is shifted to the right in this case and there is a pronounced shoulder at the right of the distribution. This leads to the rise of the threshold values towards the center of the scans. As these values are more correct than the low threshold values calculated at the borders, there is the need of a non-constant adaption of the threshold values up until the center of the specimens.

Therefore, a two-stage procedure is implemented.

1. The first stage consists of the previously described approach. A median blur filter is applied to each slice. Afterwards, the threshold value is determined and saved as $\boldsymbol{T}_{\mathrm{Otsu}}[i]$, with i being the corresponding slice.
2. At the beginning of the second stage, the average threshold value

$$T_{\mathrm{average}} = \overline{T} = \frac{\sum_{i=0}^{n+1} \boldsymbol{T}_{\mathrm{Otsu}}[i]}{n} \tag{3.6}$$

   is calculated. Following, a new array $\boldsymbol{T}_{\mathrm{new}}$ is declared. If $\boldsymbol{T}_{\mathrm{Otsu}}[i]$ is smaller than T_{average}, $\boldsymbol{T}_{\mathrm{new}}[i]$ will be set equal to T_{average}. Otherwise $\boldsymbol{T}_{\mathrm{new}}[i]$ will be set equal to $\boldsymbol{T}_{\mathrm{Otsu}}[i]$.

This procedure can be seen in the flowchart in Figure 3.13.

After that, a binary image is created from each slice i by using the threshold value $\boldsymbol{T}_{\mathrm{new}}[i]$. From these binary images, the FVC for each slice can be calculated by determining the percentage of the pixels with a non zero value. The comparison of $\boldsymbol{T}_{\mathrm{Otsu}}$ and $\boldsymbol{T}_{\mathrm{new}}$ can be seen in Figure 3.14.

The effects of this two-stage approach can be viewed in Figure 3.15. This empirical procedure was only one among multiple ones tested, but the one that made the most sense considering the non-constant adaption of threshold values reflecting the changes in the gray value distributions over the thickness.

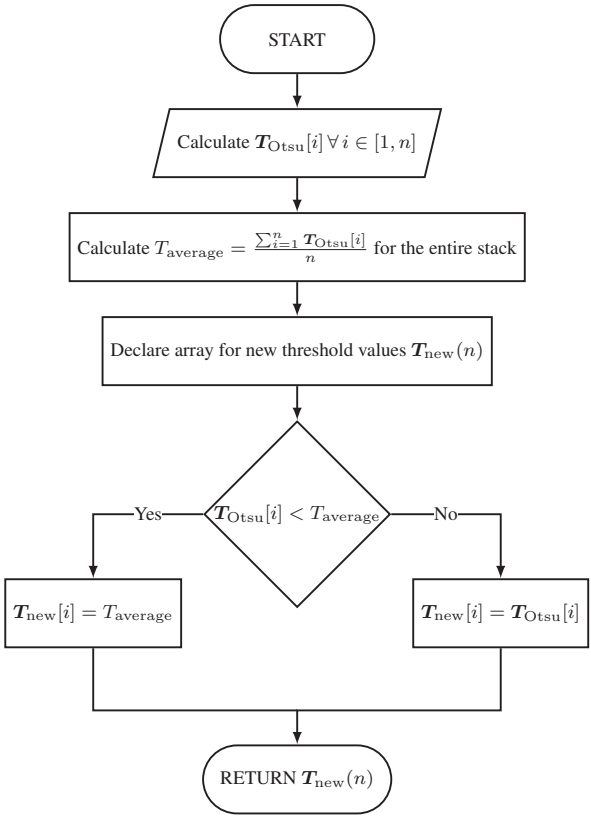


Figure 3.13: The process of the novel thresholding procedure. At first, the threshold value for each of the n images is calculated with the Otsu algorithm and stored in $\boldsymbol{T}_{\mathrm{Otsu}}$. Afterwards, the average threshold value T_{average} across all slices is determined. Then, the threshold value $\boldsymbol{T}_{\mathrm{Otsu}}[i]$ for each image i is compared to T_{average}. The larger of the two values is then stored in $\boldsymbol{T}_{\mathrm{new}}$ as $\boldsymbol{T}_{\mathrm{new}}[i]$. Extracted from Blarr et al. [188].

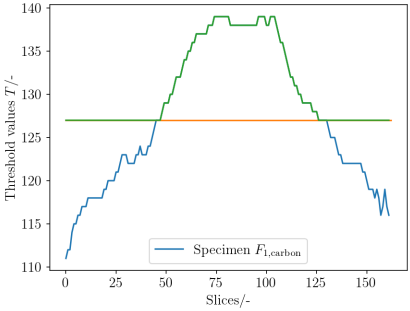


Figure 3.14: Example of the threshold values from the sample $F_{1,\mathrm{carbon}}$. The blue line represents $\boldsymbol{T}_{\mathrm{Otsu}}$, the orange line T_{average} and the green line represents $\boldsymbol{T}_{\mathrm{new}}$. Based on Blarr et al. [188].

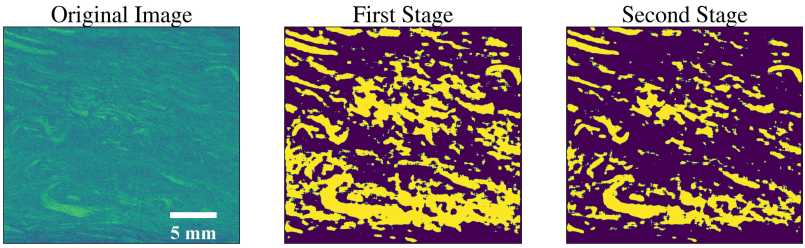


Figure 3.15: Comparison between the original image (a) and the binary image after the first (b) and second stage (c). Extracted from Blarr et al. [188].

3.4.2.3 Convolutional neural network

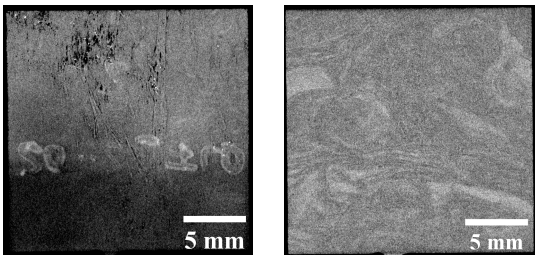
Input data and data processing [4] The convolutional neural network is implemented in Python 3.6.8 with the help of inter alia the NumPy, Scikit-image [282] and SimpleITK packages. Tensorflow [287] and Keras [288] were used as AI framework. The calculations were performed on CPUs provided by the bwHPC cluster 2. For the CNN, the 16 bit scans are loaded directly

[4] This section is extracted from the author's publication [188] with only slight linguistic changes.

into the Python script for further processing. In contrast to the thresholding method, the scans are handled as 3D arrays without a loop iterating through the slices. A comprehensive set of uniformization methods is applied so that the neural network's training algorithms solely train the network on the intended differences between the scan data. For the data loading, uniformization and augmentation, the Python libraries of SimpleITK, Keras and Numpy provide a large variety of useful methods. However, since the data is processed in 3D, a range of processing methods had to be custom-made. Those helper functions can be found in the respective Github repository as well (cf. link in [188] or at the end of this work in Chapter 9). The steps used to process the scans are as follows:

1. **Cutting**

   First, the scans were cropped individually to the actual core material volume to avoid noise at the edges of the scan volume (cf. Figure 3.16).The individual amount of cut back slices per specimen can be taken from Table 3.2.



(a) (b)

Figure 3.16: Exemplary slice near the top of the FLD_2 scan (a) showing marking and uneven surface conditions in contrast to (b) a slice of the same scan 30 layers deeper into the material. Partly extracted from Blarr et al. [188].

Table 3.2: Overview of the original scan data. The resolution is given in terms of the absolute amount of voxels in each dimension. The numbers below "Front/Back cut" refer to the amount of slices removed during data pre-processing. Based on Blarr et al. [188].

Scan	Destructively measured FVC	Original scan resolution	Front cut	Back cut
FLD_1	22.3 %	122 × 386 × 386	5	10
FLD_2	25.5 %	128 × 780 × 780	12	12
FLD_3	28.6 %	148 × 1168 × 1162	35	35
FLD_{10}	17.9 %	130 × 391 × 395	9	14
FLD_{11}	24.0 %	135 × 777 × 772	16	16
FLD_{12}	26.6 %	132 × 1164 × 1167	13	15
$F_{1,\mathrm{carbon}}$	23,1 %	168 × 1424 × 1425	0	0
$F_{2,\mathrm{carbon}}$	22,1 %	165 × 1421 × 1425	4	0
$F_{3,\mathrm{carbon}}$	23,1 %	165 × 1416 × 1428	0	0
$CF_{1,\mathrm{carbon}}$	25,6 %	171 × 1403 × 1415	1	5
$CF_{2,\mathrm{carbon}}$	22,3 %	161 × 1422 × 1421	4	0
$CF_{3,\mathrm{carbon}}$	22,8 %	165 × 1406 × 1415	0	0
$C_{1,\mathrm{carbon}}$	26,4 %	165 × 1409 × 1421	2	4
$C_{2,\mathrm{carbon}}$	23,1 %	155 × 1414 × 1421	0	4
$C_{3,\mathrm{carbon}}$	23,8 %	161 × 1406 × 1425	0	4

2. **Resizing**

   All data used to train the CNN should be of the same shape so that one input size of the network can be established. Furthermore, reshaping all arrays into cubes adds an additional possible axis to rotate the data by without changing its shape. That allows for an additional augmentation step and thus doubles the amount of input data after augmentation.
   To reshape the cuboid scans into cubes, the *transform()*-method from the *scikit-image* library was used. The target size of the cubes was constrained

by the computing power available at the bwHPC cluster. Iterative trials showed that it was capable of executing the script stably up to a cube size of $100 \times 100 \times 100$ voxels, which was thus selected as array dimension.

3. **Augmentation**

   Neural networks require extensive amounts of data to improve their training process. Since only a small amount of scans are available (14), multiple stages of 3D-image augmentations were used to enlarge the input data set.

   In the first step, every scan was rotated by $90°$ and added to the data set with the same FVC as its original. Since the cuboid shape of the transformed data allows for rotation about three independent axes, the process was repeated for the remaining two orthogonal axes. Solely rotating by multiples of $90°$ ensures that no data is lost at the edges by leaving the scope of the arrays. Furthermore, it is computationally much more efficient than a rotation by a random angle since only the array indexes need to be interchanged.

   After multiplying the data set by a factor of 4 by adding rotations of the original scans, all scans are then flipped in a second step. Similarly to the first step, copies of the original scans are mirrored at one plane and then added to the data set with the same FVC as their originals. The process is repeated for the two remaining normal planes, further multiplying the data set by a factor of 8.

   Overall, by combining three rotations and three mirroring steps, the amount of input data can be multiplied by a factor of $(1 + n_{\text{rotations}}) \cdot 2^{n_{\text{reflections}}} = (1 + 3) \cdot 2^3 = 32$. Therefore, the 14 original scans multiply to a data set of 448 samples. More combinations are possible but they lead to exact duplicates of arrays which can be obtained using the method above, i.e. two consecutive $90°$ rotations about one axis equal two reflections about two different planes.

4. **Split**

   Before being fed into the neural network, the data, consisting of CT scan arrays coupled with their respective, experimentally (destructively) observed FVC values, are split into a *training* and a *validation set*. The larger training set is used for the initial tuning of the network's parameters similar to [289].

   In between each training epoch, the validation set is used to verify the model's performance on unseen data, which prevents overfitting and allows for an estimation of the model's ability to generalize beyond the training set. More details on the training process are discussed in the *Network training process* paragraph below.

Network architecture [5] A special characteristic of the network implemented in this work is the direct input of a 3D scan along with a singular scalar value representing its corresponding FVC making it a mixed network. However, the output of the network is only the predicted FVC as a singular value between 0 and 1 for a given scan.

For image processing tasks, the conventional type of neural network is a CNN.

The model architecture in this work consists of several layers of convolutions, pooling, dense, and dropout layers. Overall, the neural network can be divided into two stages: The feature extraction stage, where the convolution is happening and the subsequent feature processing stage, where the extracted features are mapped to a corresponding output.

The input layer takes in a 3D tensor of depth, width, and height, representing the CT-scan data, as a single channel since the CT-scans are in gray scale. Furthermore, the single scalar value for the FVC is passed along.

[5] This section is extracted from the author's publication [188] with only slight linguistic changes.

The subsequent convolutional layer that extracts features from the input data is followed by a max-pooling layer that downsamples the output of the convolutional layers to reduce the dimensionality of the data and capture the most important features. In this case, a $2 \times 2 \times 2$ max-pooling layer follows the convolution, where out of the $2^3 = 8$ voxels only the largest value is passed on to the next layer. This way, an $87.5\,\%$ reduction of data is achieved without a major loss of relevant information since for feature extraction, the precise location of certain features is less relevant. Furthermore, the strongest features are enhanced more as only the highest value inside the kernel area is passed on.

After the convolutional and max-pooling layers, the output is passed through several dense layers with ReLU activation function. In the case of this study, where FVC percentages are evaluated, any negative values are implausible and get filtered out automatically this way. These layers enable the model to learn complex relationships between the input and output. Dropout layers are added after each dense layer to prevent overfitting.

The further one advances into the network during the analysis, the less relevant spacial information becomes, since the ultimate objective is to compute a single scalar, that describes the entirety of a scan. A Global Average Pooling Layer transforms the 3D output of the dense layers into a single-row vector. Its dimensionality corresponds to the amount of feature maps which the feature extraction stage feeds forward. This technique and the choice of Global Average vs. Global Max Pooling was inspired by Zunair et al. [290] and improved the performance of the CNN.

Finally, the output layer is a dense layer with a sigmoid activation function. Its output represents the predicted FVC of the CT scan.

The final network architecture can be seen in Figure 3.17. The parameters that define the shape of all layers used were determined using parameter sweeps, which are discussed in the *Network optimization* paragraph below. The graphic was made with the help of the latex code published in [291].

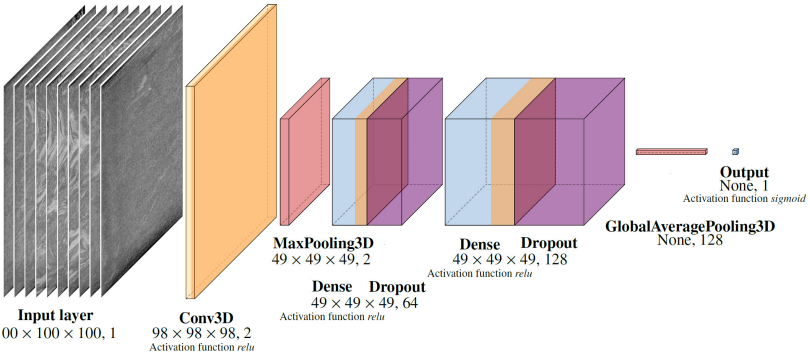


Figure 3.17: Final architecture of the CNN. Extracted from Blarr et al. [188] and based on [291].

Network training process [6] The CNN described so far is initialized with random weights in all of its layers. Therefore, the initial predictions for the scans of the training set will also be random and are unlikely to show any causal relation to their actual FVC values. In order to tune the weights in a way that enables the network to make reasonable predictions, a training algorithm is used. The training can be divided into two main steps: model compilation and model fitting. During the model compilation step, the configuration parameters are set up. The chosen loss function for this task is Binary Cross Entropy (BCE). The BCE loss, also referred to as log loss or negative log probability, is defined as follows

$$l\left(y, \hat{y}\right) = -\Big(y \log\left(\hat{y}\right) + (1-y) \log\left(1-\hat{y}\right)\Big), \tag{3.7}$$

with y being the true term (0 or 1) and $\hat{y}$ being the predicted probability (between 0 and 1) [292, 293]. In the perfect case of the model exactly predicting 0 or 1 correctly, the loss amounts to zero. However, in this work the true term y is not only zero or one, but a continuous value in between. While this is not what the BCE loss was originally intended for, it worked better than other popular loss

[6] This section is extracted from the author's publication [188] with only slight linguistic changes.

functions that are not intended for binary input (and output) such as the mean square error (MSE) or mean absolute error (MAE) or regular cross entropy. Literature also suggests that it works in practice [294, 295], so the BCE loss was used in this work. It shall be noted that BCE is asymmetric in the case of the ground truth not being a binary value and the minimum loss is not zero anymore. As an example for $y = 0.24$, which is roughly the average FVC in this case, the minimum loss for $\hat{y} = y = 0.24$ amounts to 0.5511 (cf. Figure 3.18). This will be relevant when judging the loss plot in the Results (cf. Section 4.3.3).

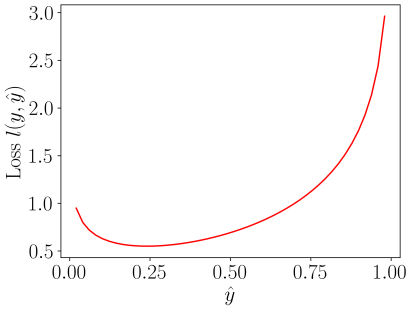


Figure 3.18: Course of the BCE loss over prediction values $\hat{y}$ between 0 and 1 for a true value of $y = 0.24$. Extracted from Blarr et al. [188].

The optimizer selected for this task is ADAM. It is an adaptive learning rate optimization algorithm that adjusts the learning rate dynamically during training, which helps to converge to an optimal solution faster. Because of its computational efficiency and little memory requirement [201], it is a popular choice for training deep neural networks due to its efficiency and effectiveness in updating the parameters of the model. Additionally, the mean squared error (MSE) is used as a metric to evaluate the performance of the model during training. It measures the average squared difference between the predicted and true values. Therefore, it provides insight into the overall accuracy of the predictions of the model along the training process.

After model compilation, the model is fitted to the training data using the *model.fit()* function from Tensorflow. This method implements the general process of machine learning using the parameters set in the compilation process: Using the weights provided at that stage in the network, all training scans are passed through the network. The resulting scalars (the predictions) are compared to the actual FVC values of the respective specimens. The loss function, which was specified earlier, takes both the prediction and the real value as its arguments and computes a loss score. Subsequently, the selected optimizer adjusts the weight in the layers according to the performance of the score, before the entire process is repeated by the set amount of epochs specified in the beginning. Figure 2.19 describes this process visually.

Alongside the training set, the validation set is being evaluated with the same model and loss function simultaneously. Finally, a loss plot, which can be seen in Figure 4.13, is generated using the training and validation loss values. The plot shows both as a function of the number of epochs. This visualization helps to monitor the convergence and performance of the model during training, where a decreasing loss indicates that the model is learning and improving over time.

Furthermore, this method also takes over the task of shuffling the input data and splitting it into training and validation sets, for which a ratio of 2/3 to 1/3 was set. The number of training epochs is set to 40, indicating the number of times the entire training data set is passed through the model during training. Once the model is trained, it is used to predict the fiber volume content for the test data. The deviation between the predicted values and the true labels is then plotted, as visible in Figure 4.14.

Network optimization [7] The CNN requires a lot of parameters to be defined. It was started with values provided in similar literature [187, 289], which were adapted to the problem at hand. Most parameters were defined by so-called

[7] This section is extracted from the author's publication [188] with only slight linguistic changes.

parameter sweeps. Instead of single values, the parameters were provided with a list of values, which were looped through. By changing two parameter values at once, one can find the combination with the best performance. As an example, the amount of filters in a 3D convolutional layer is selectable in Keras. The best amount of connected dense-dropout layers was also unclear. Hence, lists for the amount of filters in the convolutional layer $n_{\mathrm{filters}} = \{2, 4, 8, 16, 32\}$ as exponential values to the basis 2 in order to cover a greater field and a list for the amount of dense-dropout layers $n_{\mathrm{dd-layers}} = \{1, 2, 3, 4, 5\}$ was given to the network instead of singular values of these parameters. These lists are then iterated through and training is carried out for each combination, i.e. each new network. The standard average deviation for the results of each network is finally compared and the combination of parameters that leads to the best FVC prediction is used.

A similar procedure was followed for the parameters of the dropout rate in the dropout layer, the pool size in the max-pooling layer and the kernel size in the convolutional layer. Analogously, multiple initial learning rates, loss functions and optimizers were tested until the author arrived at the presented architecture in Figure 3.17.

3.5 Determination of fiber length distributions

In the following, it is explained how the fiber length distribution of individual samples was determined experimentally.

Experimental method

Owing to the process-related stochastic scattering of the fiber lengths, a determination of the fiber length distribution is of particular interest in the case of LFT materials. As it was impossible to separate individual fibers in the CT images of the CF-PA6 scans, only an experimental determination was carried

out at the FIBRE. This fiber length analysis is conducted after the resolution of the surrounding matrix.

As the dispersion of the resulting fibers was not ensured adequately for the pyrolysis results, only the FLD of the specimens investigated through chemical dissolution of Plate 2 are taken into account here.

Therefore, distilled water and detergent are added to the carbon fibers obtained by acid-based dissolution in a sample vessel. They are then evenly distributed throughout the sample vessel without breaking them. After placing a cover slip on it, it was placed in the so-called "FiVer" scanner (cf. Section 2.2.2.1). Several scans of the fibers in the liquid are then performed. These images are loaded into the "FibreShape" software, which performs the analysis of the fiber lengths. The .csv-files obtained after completion were visualized by the authors in the Results in Section 4.4. These investigations were also done at FIBRE (Bremen).

3.6 Determination of fiber orientation tensors

In order to determine fiber orientation tensors, the specimens cut out of plates/parts through waterjet cutting, were installed on the rotary table of the CT device. After adjusting the specific scan parameters, the CT scan takes place. The resulting 2D projections were reconstructed with the Feldkamp cone-beam algorithm [154] in VG Studio Max in the Versions 3.3.2, 3.4.2 and 2023.2.1. After loading the reconstructed version in the same software, the registration of the object, hence the alignment of the (often straight) object borders with the coordinate system of the viewpoint is performed. In the case of multiple specimens in one scan, this was repeated for each separate object. These were then cut out as a ROI and exported separately as .raw-files. Along with respective sizes of dimensions, these can be loaded in the software FIJI/ImageJ, where an adjustment of brightness takes place and subsequently an approximate gray value threshold differentiating between fiber and matrix is determined. While

there are pre-implemented automatic thresholds available in ImageJ (Otsu, Mean, Moments, Median, etc.), sometimes the manual determination of the threshold made more sense in the sensitive case of CFRP. The resulting stacks of 2D images are saved as .mhd-files which allow for further processing without additional information. This is followed by the actual orientation determination. The C++ code by Bertram and Pinter, which was published in Composight [79], is used for this evaluation. Its basic functionality based on the structure tensor was explained in the State of the Art in Section 2.2.3.2. In practice, the executable (.exe-) version of the code was used, which was accessed via console. Apart from the .mhd image file and the threshold value determined before, the width σ of the Gaussian blur applied in combination with the derivative and the second blurring parameter ρ, also referred to as the mask size, have to be specified. σ was usually chosen to 0.2, whereas ρ was typically eight or smaller, but always larger than σ. After running, a vector-valued image as another .mhd file with three channels is output storing the information of orientation in each coordinate direction. This .mhd-file is finally read into MATLAB R2020b as a fourdimensional matrix with the *mha_read_volume* function available on *Mathworks File Exchange* in the *Read Medical Data 3D* toolkit by Dirk-Jan Kroon from 2010 [296]. In MATLAB, the orientation tensors of second- and fourth-order are determined along with many other evaluations, which will be explained in the following and of which the results will be shown in Section 4.5. The entire procedure is summarized in the flowchart in Figure 3.19.

FOT calculation

After importing the vector-valued image (.mhd file resulting from the C++ code by Bertram and Pinter [79] in Figure 3.19), the following procedure to calculate the FOT implemented by Schöttl [25] was used. The components of the voxels can be accessed as follows:

```
image(c, x, y, z),
```

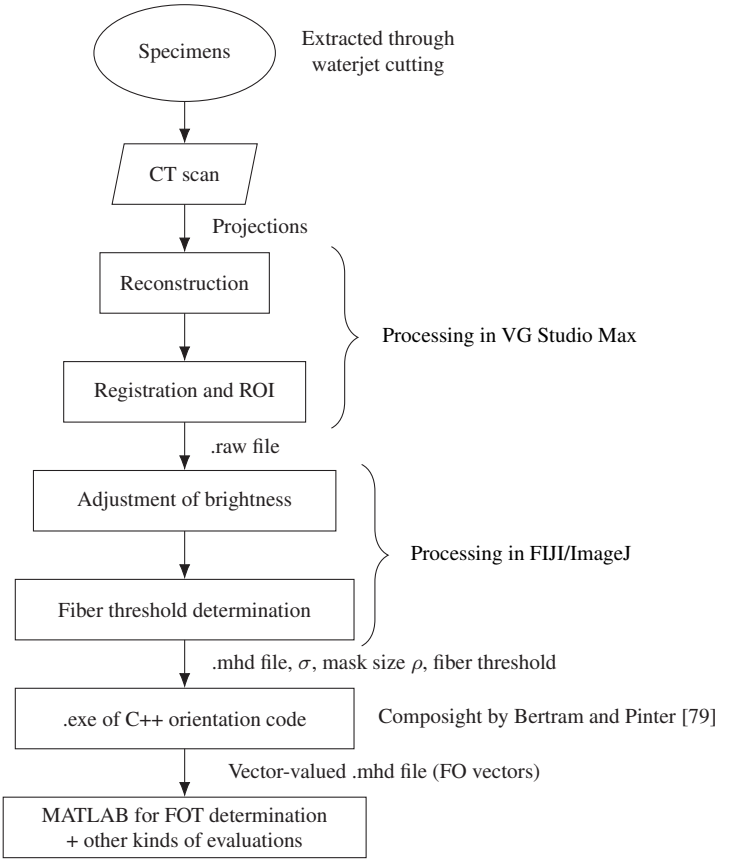


Figure 3.19: Flowchart of the typical procedure for determining the fiber orientation tensors.

with `c` being the component of the orientation vector at the position `x, y, z`. Hence, the orientation tensor of second-order can be determined using the sum function and element-by-element multiplication (using the .* Operator in MATLAB) of two channels of the image over all voxels. As an example, to calculate the A_{11} component, the multiplication of the voxel values of the first channel with itself is needed, for A_{12} the first and second channel are used, etc.):

```
OT2(n)=sum(sum(sum(image(i,:,:,:).*image(j,:,:,:)))).
```

Of the nine components, only six are calculated due to the tensor symmetries and saved in the following order:

```
[A_11 A_12 A_13 A_22 A_23 A_33].
```

In order to guarantee the trace condition (cf. Equation (3.15)), the FOT is normalized as such:

```
OT2=OT2/(sum([OT2(1) OT2(4) OT2(6)])).
```

The FOT of fourth-order is calculated analogously and saved in the following order:

```
A_3333 A_3332 A_3322 A_3222 A_2222 A_3331 A_3321 A_3221
A_2221 A_3311 A_3211 A_2211 A_3111 A_2111 A_1111.
```
[52]

Visualization via pseudo color image

In order to visualize the planar orientation of the fibers in the slices, the fibers are colored according to their angle resulting in a pseudo color image. This procedure was implemented by Schöttl as well [25]. Therefore, the angle between the x-axis and the voxel vector is calculated through the arctangent as

```
phi(:,:,:)  = atan(img(2,:,:,:)./img(1,:,:,:))  + pi/2.
```

This angle becomes the H value in the HSV (hue, saturation, and value) color space, which is illustrated below (Figure 3.20).

The further S and V values are both set to one, implying full saturation and brightness. The colored slices are finally saved as .tiff-stack. [25] This kind of visualization is especially helpful to detect flaws in the angle determination.

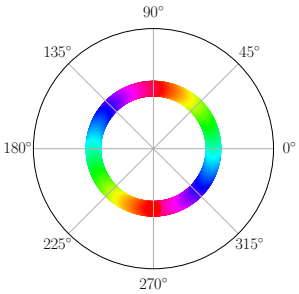


Figure 3.20: Visualization of the HSV color space with respect to the fiber orientation symmetry ($\boldsymbol{p} \widehat{=} -\boldsymbol{p}$). Extracted from [274].

Further depictions of orientation information

In order to judge the behavior of the orientation across the thickness of the specimens (respectively the plates) and not only obtain one single tensor per specimen, the determined main components, A_{11}, A_{22} and A_{33}, were plotted for each slice progressively. This allows for process-related conclusions.

Furthermore, while the FOT were used in the Mori-Tanaka homogenization, a distribution of angle occurrences was required for Halpin-Tsai. The occuring angles are partitioned into $n_{\text{centers}} = 20$ centers with an interval arc length of $n_i \in [i \cdot \pi/20, (i+1) \cdot \pi/20)$ with $i = \{x \in \mathbb{N}_0 : x < 20\}$. It is subsequently counted how often angles fitting in each specific angle group appear in the specimen in order to obtain a discrete fiber orientation histogram.

Challenges for CFRP and hybrid reinforced material

While the orientation determination worked without problems for GFRP, the CF-PA6 CT scans showed much more salt and pepper noise, which led to some orientation recognition inconsistencies that were conspicuous in the HSV images. Hence, compared to the description of Pinter et al. [52] an additional

median filter with a kernel size of 10 was applied to the images of the carbon fiber reinforced specimens only.

A further challenge was the orientation determination for hybrid GF-CF reinforced material as the thresholding is more difficult for four different constituents in the CT scan (air, polymer, GF, CF). Therefore, a multiple thresholding procedure was used in order to separate the different fiber types. The glass fibers were thresholded first as they have better contrast towards the matrix. After the resulting image stack was subtracted from the original one, the air is thresholded. Subsequently, the resulting image stack is again subtracted from the original stack. Air then appears completely black and the glass fibers completely white. Afterwards, a twofold limited threshold was applied, cutting off air (black), glass (white) and matrix, which has the closest gray value to the carbon fiber one. Finally, the original image stack was fed to the C++ code with the glass fiber threshold evaluating the glass fiber FOT. The final stack following the procedure just described was subsequently given to the C++ code along with the carbon fiber threshold to acquire the carbon fiber FOT.

3.7 Interpolation of fiber orientation tensors

The motivation to find a tensor interpolation method suitable for the interpolation of fiber orientation tensors was already given in Section 2.2.3.3. In the following sections, three methods, that were implemented and evaluated in this thesis, to obtain full-field FOT information, are explained. The sections are extracted from the author's publication [17].

3.7.1 General notes on the interpolation methods [8]

In general, all interpolation methods were mainly implemented in Python 3.8.

SPD tensors can be visualized as tensor glyphs [116, 117]. This method will be used in this work as it constitutes a descriptive and interpretable way of assessing the success of the different implemented interpolation methods. The visualization was implemented in Matlab R2020b with the help of the “plotDTI” function of the fanDTasia ToolBox by Barmpoutis et al. [118].

The overall idea of all three implemented interpolation methods is to get FOT values for the 160 positions in the plate that are missing, from calculations with the measured nine FOT at the given grid positions. Determining values between a set of measured values, here the set $\mathcal{T}_{\mathrm{m}} = \{\textbf{UL, UM, UR, ML, MM, MR, LL, LM, LR}\}$ (respectively denoting “Upper Left", "Upper Middle", "Upper Right", "Middle Left", ..., "Lower Left", etc.) of measured FOT computationally, based on the set of measured values, describes a classical interpolation problem. An interpolation scheme $\bar{\phi}$ is defined as a mapping f, which connects its arguments, on the one hand a set of $N \geq 1$ discrete values ϕ_i and on the other hand their associated weights $w_i \in [0; 1]$:

$$\bar{\phi} = f(\phi_i, w_i). \tag{3.8}$$

As a weight function, multiple options are conceivable with the possibly simplest being Shepard’s inverse distance weighting (IDW) [297]

$$w_i(\boldsymbol{x}) = \frac{1}{d(\boldsymbol{x}, \boldsymbol{x}_i)^p}, \tag{3.9}$$

[8] This section is extracted from the author’s publication [17] with only slight linguistic changes.

as an explicit approach with $\boldsymbol{x}$ denoting an arbitrary point that shall be interpolated, $\boldsymbol{x}_i$ being a known interpolating point and d being the given distance from the known point $\boldsymbol{x}_i$ to the unknown point $\boldsymbol{x}$. p is a positive real number, called the "power parameter". Weight decreases as distance increases from the interpolated points. Greater values of p assign greater influence to values closest to the interpolated point, which results in nearly constant interpolated values for large values of p.

3.7.2 Component averaging [9]

Recalling the definition of second-order orientation tensors as described by Kanatani [89] as well as Advani and Tucker [43],

$$\boldsymbol{A} = \int_{\mathcal{S}} \Psi(\boldsymbol{p}) \boldsymbol{p} \otimes \boldsymbol{p} \, dp, \tag{3.10}$$

with $\mathcal{S}$ being the unit sphere and dp the surface element on it, as well as $\boldsymbol{p}$ being the unit vector for the direction of the fibers, it appears that $\boldsymbol{A}$ is linear in $\Psi(\boldsymbol{p})$. Assuming the surface can be divided into two equally sized areas $\mathcal{S}_1$ and $\mathcal{S}_2$ with two distribution functions $\Psi_1(\boldsymbol{p})$ and $\Psi_2(\boldsymbol{p})$ and $\Psi_i(\boldsymbol{n}) = \frac{1}{2}(\Psi_1(\boldsymbol{n}) + \Psi_2(\boldsymbol{n}))$ holds, this means that $\boldsymbol{A}_i = \frac{1}{2}(\boldsymbol{A}_1 + \boldsymbol{A}_2)$ is exact, as integration is a linear functional and as an integral domain can always be divided into sub-intervals. This further implies that a direct averaging of the orientation function is equivalent to an averaging of the components of the orientation tensors. Thus, the algorithm multiplies the components of each measured FOT by a weight that depends on its distance from the tensor being calculated. As mentioned before, Shepard's inverse distance weighting method [297] is used as weight function in all three methods with $p = 2$:

[9] This section is extracted from the author's publication [17] with only slight linguistic changes.

$$w_i = \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{x}\|^p} \frac{1}{\sum_j \|\boldsymbol{x}_j - \boldsymbol{x}\|^{-p}}. \tag{3.11}$$

Compared to Equation (3.9), Equation (3.11) features a necessary normalization factor.

3.7.3 Decomposition method [10]

The method, which uses spectral decomposition of tensors, is shown schematically in Figure 3.21 and is described in the following.

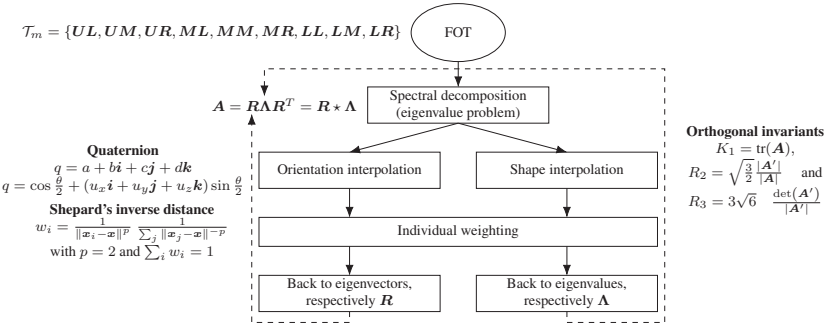


Figure 3.21: Overview of the concept of the implemented decomposition method. Based on the graphic in the author's publication [17].

For the chosen decomposition approach, the shape and orientation of the tensors are to be interpolated separately. Therefore, the well-known spectral decomposition resulting from the eigenvalue problem is used:

$$\boldsymbol{A} = \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{\boldsymbol{T}} = \boldsymbol{R} \star \boldsymbol{\Lambda}. \tag{3.12}$$

[10] This section is extracted from the author's publication [17] with only slight linguistic changes.

$\boldsymbol{\Lambda}$ denotes the tensor containing the eigenvalues on the principal diagonal and $\boldsymbol{R}$ is defined as the orthogonal rotation matrix consisting of the normalized eigenvectors.

Orientation The rotation matrix $\boldsymbol{R}$ can be interpreted as a rotation around a rotation axis and can therefore be transformed into a quaternion as described in the State of the Art:

$$
\begin{aligned}
q &= a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}, \\
q &= \cos\frac{\theta}{2} + (u_x\boldsymbol{i} + u_y\boldsymbol{j} + u_z\boldsymbol{k})\sin\frac{\theta}{2},
\end{aligned}
$$

with rotation axis: $\boldsymbol{u} = (u_x, u_y, u_z)^T$ and rotation angle θ.

The quaternion is calculated from the given rotation matrix $\boldsymbol{R}$ via:

$$
\begin{aligned}
&t = \mathrm{tr}(\boldsymbol{R}), \quad r = \sqrt{1+t} \quad \text{and} \quad a = \frac{r}{2} \quad \text{with} \\
&b = \mathrm{sgn}(R_z y - R_y z)\left|\frac{1}{2}\sqrt{1 + R_x x - R_y y - R_z z}\right|, \\
&c = \mathrm{sgn}(R_x z - R_z x)\left|\frac{1}{2}\sqrt{1 - R_x x + R_y y - R_z z}\right|, \\
&d = \mathrm{sgn}(R_y x - R_x y)\left|\frac{1}{2}\sqrt{1 - R_x x - R_y y + R_z z}\right|.
\end{aligned}
$$

This is followed by the actual interpolation: $q_{ges} = \sum_i w_i q_i$ with weights: $\sum_i w_i = 1$ and the retransformation of q_{ges} in $\boldsymbol{R}$:

$$\boldsymbol{R} = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{bmatrix}. \tag{3.13}$$

Shape For the interpolation of the shape, three linear independent invariants are formed of each tensor and interpolated separately. Of the already mentioned K- and R-invariants [124] K_1, R_2, and R_3 will be used (comparable to [123]):

$$K_1 = \mathrm{tr}(\boldsymbol{A}), \qquad R_2 = \sqrt{\frac{3}{2}}\frac{|\boldsymbol{A}'|}{|\boldsymbol{A}|} \quad \text{and} \quad R_3 = 3\sqrt{6}\,\frac{\det(\boldsymbol{A}')}{|\boldsymbol{A}'|}. \tag{3.14}$$

A' is the deviatoric part of **A**.

Even though K_1 and R_2 are not orthogonal [124], the use of K_1 can be justified by ensuring that the trace of the orientation tensor is one. It is not necessarily essential to have orthogonal invariants for this specific case of application.

The invariants are then interpolated individually: $K_{1,\mathrm{ges}} = \sum_i w_i K_{1i}$, $R_{2,\mathrm{ges}} = \sum_i w_i R_{2i}$ and $R_{3,\mathrm{ges}} = \sum_i w_i R_{3i}$. The weights stay the same: $\sum_i w_i = 1$.

From the interpolated invariants, the following formula was used to calculate the associated eigenvalues (cf. [123]):

For $i = 1, 2, 3$ holds: $\lambda_i = \frac{1}{3}K_1 + \frac{2K_1R_2}{3\sqrt{3-2R_2^2}}\cos\left(\frac{\cos^{-1}(R_3)+P_i}{3}\right)$,
with $P_i = 0, 2\pi, -2\pi$

With these eigenvalues, $\boldsymbol{\Lambda}$ can then be created again.

3.7.4 Artificial neural network[11]

The artificial neural network used in this study is based on the idea and implementation of Sabiston et al. [133]. Just like for the other two methods, the goal of the neural network is to determine a FOT for each specified point of the 160 missing positions within the plate. For the ANN, the nine x, y (and z) coordinates of the given FOT were normalized (divided by 14 since there are 13 rows and columns of FOT). These $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ represent the input data. The output data for the ANN are the respective components of the nine given orientation tensors of second-order. Since these components are already between -1 and 1, this data does not need to be normalized. The coordinates were read in as one .csv file as input and the components separated by A_{11}, A_{33}, A_{12}, A_{13} and A_{23} as five separate .csv files as output.

There are only five independent components instead of the usual six independent ones for symmetric tensors, since orientation tensors are subject to an additional condition that the trace of the tensor must add up to 1:

$$\sum_{i=1}^{3} A_{ii} = 1. \tag{3.15}$$

Therefore, only A_{11} and A_{33} were fed into the network as output parameters and A_{22} was determined via $A_{22} = 1 - A_{11} - A_{33}$. The choice was made explicitly, according to the findings of Sabiston et al. [133], to use only one in-plane coordinate and the through-thickness coordinate (A_{22} and A_{33} would have worked analogously as well) in order to reduce the error and to satisfy equation (3.15). This is due to the two in-plane coordinates being significantly larger than the through-thickness coordinate, which in turn meant that A_{11} and A_{22} alone could get above 1 if they were both predicted.

[11] This section is extracted from the author's publication [17] with only slight linguistic changes.

The ANN consists of an input layer, where the normalized input coordinates of the $i = 9$ different points are given and two hidden layers with $n = 48$ neurons. The output of the first hidden layer is the input for the second hidden layer. In the output layer the five independent tensor components $A_{11}, A_{33}, A_{12}, A_{13}$ and A_{23} are predicted for the given $i = 9$ points. The structure of the ANN can be seen in Figure 3.22.

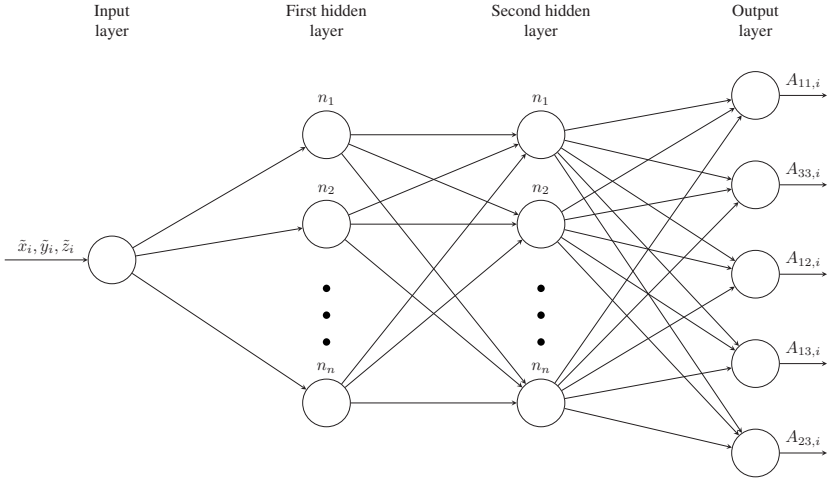


Figure 3.22: Schematic concept of the ANN used for FOT interpolation. Extracted from Blarr et al. [17]).

The optimizer used is SGD which is an iterative method for optimizing an objective function $Q(w)$ with suitable smoothness properties. Thereby, after choosing an initial vector of parameters w and a learning rate r_l, two steps are repeated until an approximate minimum is obtained: The samples in the training set are randomly shuffled and $w := w - r_l \nabla Q_i(w)$ is set for $i = 1, 2, ..., n$ (with Q_i being the summand function typically associated with the i-th observation in the data set (used for training)).

The loss function chosen is the MAE, which is defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} = \left| A_{jk(i)} - \hat{A}_{jk(i)} \right|, \tag{3.16}$$

with n being the number of samples, $A_{jk(i)}$ being the value of the orientation tensor component at the sample location and $\hat{A}_{jk(i)}$ being the predicted value of the orientation tensor component at that sample location. MAE was preferred as error metric over percentage error since many values (especially the off-axis and A_{33} components) are close to zero. Therefore percentage errors tend to become quite large. Additionally, outliers seem to be filtered out better by using MAE than by a quadratic error metric like root mean squared error which is more likely to result in overfitting and being biased towards outliers respectively. Additionally, a soft sign activation function is used in the model as it is able to calculate negative numbers and behaves differently in terms of saturation (compared to, e.g., the hyperbolic tangent) because of its smoother asymptotes (polynomial instead of exponential) [298]. However, this of course impacts the amount of epochs required for training as it does not saturate as quickly. The soft sign activation function is given as

$$o = \frac{i}{1 + |i|}, \tag{3.17}$$

where i is the input to the function and o is the output of the function. Furthermore, a bias was placed on the loss weights (w) of the outputs of the ANN in order to give more weight to the in-plane orientations. The biases are 0.4 for A_{11}, and 0.15 for all four other components, adding up to 1. The classical data validation split of 25% is used in the study. The high amount of 100,000 epochs, i.e., times the neural network iteratively trains the weights for each neuron to optimize the outputs from the given input steps, was chosen. While increasing the epochs normally reduces the error, it can also evoke overtraining and leads

to longer calculation times. All chosen parameters of the ANN are summarized in Table 3.3.

Table 3.3: Parameters of the neural network for the FOT interpolation. Extracted from Blarr et al. [17]).

Parameter	Value
Software	Keras 2.4.3 in Python 3.8.5
Optimizer	SGD
Loss function	MAE
Layers	2 hidden (plus 1 input and 1 output layer)
Neurons per layer	48
Training data	75 %
Validation set	25 %
Epochs	100,000

Once the model is trained, a .csv-file with all 160 normalized coordinates - apart from the nine the network was trained with - is given to the trained ANN, to predict the components of the missing FOT.

3.7.5 Simplified decomposition method for tensors of fourth-order

In order to subsequently test the quality of the fiber orientation tensor interpolation, the use of an averaged (from all measured and interpolated ones) fiber orientation tensor in Mori-Tanaka and of an averaged (again from all measured and interpolated ones) angular distribution in Halpin Tsai was planned. This provides stiffnesses for given angles, which can then be compared with experimental results. However, Mori-Tanaka requires a fourth-order fiber orientation tensor. The interpolation method for tensors of second-order could

have been used notwithstanding, and the final averaged tensor could have been converted into a fourth-order tensor by a closure approximation. Nevertheless, closures introduce a non-negligible inaccuracy into the method. Consequently, a direct interpolation of fourth-order tensors was developed instead. As the decomposition method proved to be particularly useful, this idea was also used for the interpolation of fourth-order tensors. However, a simplification was chosen in this case. The transformation to quaternions and orthogonal invariants was not considered as the transformation is not straightforward. Hence, the rotation matrix and the eigenvalues of the tensors of fourth-order are used directly. Therefore, the tensors were read into Python in Mandel notation and subsequently decomposed into eigenvalues (6×1) and the rotation matrix (6×6) using NumPy's linalg library [277]. Subsequently, the eigenvalues were sorted by magnitude. The fact that portraying a 3D tensor in 2D using Mandel notation and thus receiving a 6×6 matrix, allowed the author to apply the spectral decomposition. Afterwards, the distance-dependent weights applied to each measured tensor for each tensor position to be determined were also calculated with Shepard's inverse distance weighting [297] as explained before. Reassembling the newly weighted rotation matrix and eigenvalues leads to a then interpolated tensor for a specific point.

3.8 Microstructure generation through generative adversarial network

The methodology of image generation by the DCGAN is explained below. The computational resources and software are discussed first, followed by the preparation of the input data and the network architecture, and then various quantitative evaluation methods of the network and the generated images. Parts of this have already been published in a paper [275] and are marked accordingly.

3.8.1 Computational resources and software

As the generator and discriminator can consist of several million trainable parameters each, a high amount of computational power was inevitable. The training of the GAN was therefore performed on the Baden-Württemberg High Performance Computing (bwHPC) resources and in particular on the GPU x4 partition of bwUniCluster 2.0. Its hardware and architecture is shown in Table 3.4 below.

Table 3.4: Specification of the GPU x4 partition compute nodes on bwUniCluster 2.0. Based on [299].

GPU x4 partition resources	
Processors	Intel Xeon Gold 6230
Number of sockets	2
Processor frequency	2.1 GHz
Total number of cores	40
Main memory	384 GB
Accelerators	4x NVIDIA Tesla V100
Accelerator memory	32 GB
Interconnect	IB HDR

While the CNN in Section 3.4.2.3 was solely trained via the Jupyter web access (JupyterLab 3.6.2), this approach was only used for the GAN for less computationally expensive jobs as only a limited amount of resources is available in this case. For all jobs requiring either high random-access memory (RAM), multi graphics processing unit (GPU) usage (NVIDIA CUDA) or a longer computing time, direct remote access through the command prompt with the standard Secure Shell protocol (SSH) was used to access the bwUniCluster 2.0.

The entire code is written in Python using both native Python files (.py) and Jupyter notebooks (.ipynb). The FIJI application of ImageJ2 was helpful for

image visualization [300]. The following Python libraries and modules have been made use of in a virtual environment:

- PyTorch 1.10.1 and the Torchvision 0.11.2 library [301, 302]
- Numpy 1.19.5 [277]
- The OpenCV cv2 4.7.0.72 library for image editing [278]
- Matplotlib 3.3.4 for creating plots [279]
- TorchMetrics 0.8.2 for evaluating the image quality [303]
- Torchinfo 1.5.4 for structure information about the models [304]
- IPython 7.16.3 for live progress in Jupyter notebooks [305]
- tqdm 4.64.1 to show progress bars [306]
- Mahotas 1.4.13 to calculate Haralick features [307]
- SimpleITK 2.2.1 for image editing [284]

The Jupyter web access ran with Python 3.9.7, whereas the older Python 3.6.8 was installed on the cluster itself requiring some modifications when switching.

3.8.2 Image pre-processing

The scans of the nine specimens of Plate 3 were used along with the scans of nine specimens at identical positions of another CF-PA6 plate. The only difference of the latter, which is not pictured or numbered in Section 3.1.6 and Figure 3.4, was the insertion of the plastificate rotated through 180° to find possible changes in the fiber orientation drift. However, the overall fiber orientation and material behavior is identical. In addition, very similar scan parameters were used. The further processing of these 18 raw images is summarized in Figure 3.23 and will be described shortly in the following paragraphs.

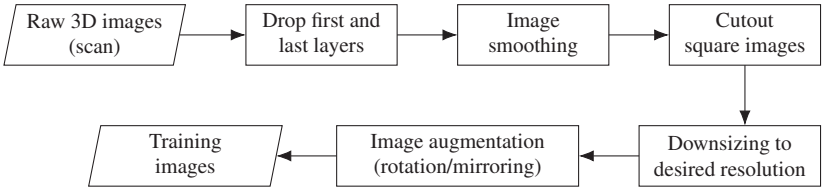


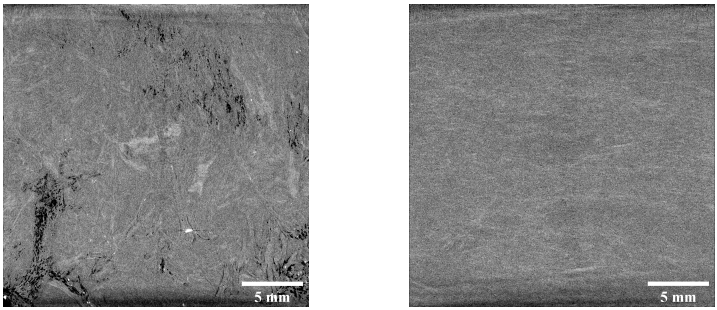
Figure 3.23: Schematic depiction of the image pre-processing procedure from CT scans to training images.

3.8.2.1 Dropping border layers

The 18 3D scans have slightly varying width, height and layers (thickness) with minimum values of width $\times$ height $\times$ layers = 1366 $\times$ 1346 $\times$ 155 voxel. They are sliced in thickness direction to obtain 2D images with the largest dimensions and most information per slice. For every of those raw images, the first and last 30 layers were dropped. They were especially prone to portraying artifacts or air-rich areas (cf. Figure 3.24(a)). While these outer layers close to the specimen surface also show below average fiber volume contents due to the manufacturing process (cf. Figure 3.24(b)), varying FVC in the images was actually desired. Variety in realistic (not artifact based) microstructure depictions allows for the network to actually reproduce diverse output. Fortunately, even with the cut of the first and last 30 layers, still enough variety of FVC was given both within one scan and between the different scans.

3.8.2.2 Cutout images

In order to avoid dark border regions in the 2D images, specific cutout sections were considered. To further increase the amount of training images from the available scans, it was decided to use multiple cutout sections. Therefore, four 1,024 px $\times$ 1,024 px regions of interest were cut out of every of the remaining layers, each with a given arbitrarily chosen offset of 128 px from the center in x- and y-direction (cf. Figure 3.25). While it can be chosen arbitrarily as well, the size of the cutout section had the advantage of being a multiple of the later



(a) An examplary first layer with surface roughness and air inclusions

(b) An exemplary ninth layer with low FVC

Figure 3.24: Two exemplary outer layers of different scans showing air-rich areas and a patchy surface (a) and a low FVC (b).

input size of 256 px $\times$ 256 px. Possibly less information is lost that way during the downsizing operation.

3.8.2.3 Downsizing and image smoothing

Considering downsizing, a balance between computation time and sufficient level of detail in both input and generated images was sought. After initially setting a size of 128 px $\times$ 128 px, finally, a size of 256 px $\times$ 256 px was chosen. The downsizing operation was performed with the recommended *INTER_AREA* interpolation from cv2. While this higher resolution allowed for the recognition of smaller fiber bundles and other details, it also conveyed a fair amount of salt-and-pepper noise. In order to avoid the generator learning this noise as desirable feature and applying it to the output images, the cutout images were blurred using a median filter before resizing them to the desired input resolution. A kernel size of 5 px $\times$ 5 px was chosen. Figure 3.26 exemplary shows an image of a CT scan slice with (a) and without (b) median filter.

The blurring effect is visible but still most details are recognizable. In fact the average signal-to-noise ratio (SNR) of all input images without median filter is

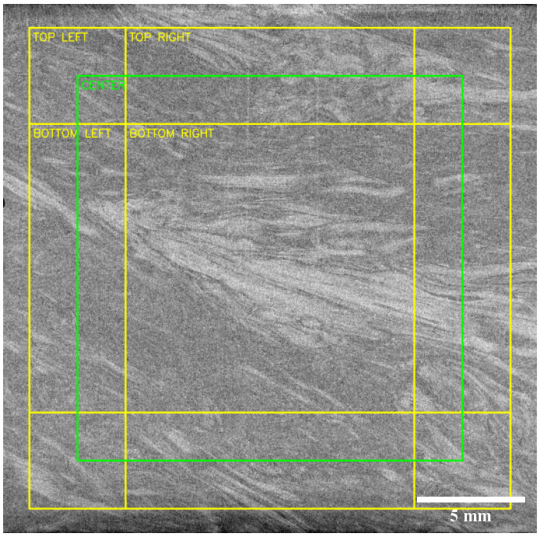
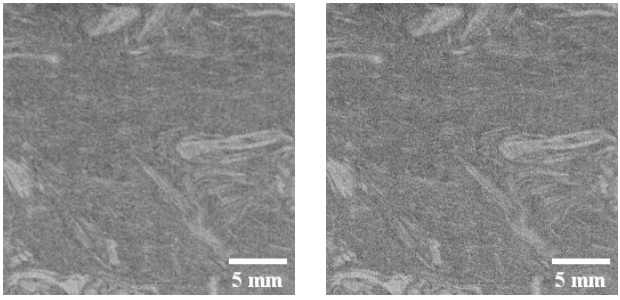


Figure 3.25: Exemplary layer with center (green) and offset (yellow) by $\pm$ 128 px in x- and y-direction cutout sections of 1.024 px $\times$ 1.024 px superimposed.



(a) (b)

Figure 3.26: An example image of a CT scan slice with (a) and without (b) median filter.

with a value of 5.27 lower than the average SNR of all input images with median filter of 5.90. The latter were hence used as input images for the training.

3.8.2.4 Image augmentation

As most neural networks, GANs require a large training data set to be able to detect reoccurring image features. Figures of $10^5 - 10^6$ images are suggested for modern, high-resolution GANs [308]. The high number of images is particularly relevant to reduce overfitting [309], which leads to the discriminator feedback becoming meaningless for the generator and finally the training diverging [308]. As reaching such a high amount of original training data is cumbersome and at times impossible, image augmentation has developed as the default solution to multiply the dataset [308, 309]. Augmentation refers to a deliberate, slight change to existing training images in order to create new training images that expand the original images within a meaningful scope to increase input dataset variety and improve training. Augmentation strategies for gray value images include but are not limited to rotation, flipping, translation, scaling and modifications of brightness and contrast [308]. In the case of the GAN presented in this thesis, every cutout image was rotated by 180° (corresponding to a point reflection) and additionally mirrored both horizontally and vertically, which can be seen in Figure 3.27.

A simultaneous mirroring at the horizontal and vertical axis was omitted due to being a duplicate to the 180° rotation. Furthermore, rotations by $\pm$ 90° were also left out to preserve the preferred fiber orientation resulting from the compression molding material flow. After all cutting and augmentation steps, this results in a total of 29,280 2D training images that are fed into the network as input.

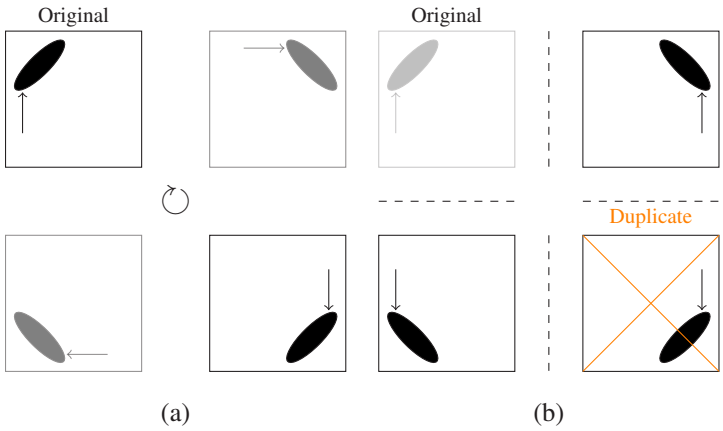


Figure 3.27: Visualization of the image augmentation through (a) rotation and (b) mirroring. The arrows indicate the orientation of the images.

3.8.3 Network architecture [12]

The code provided in the official PyTorch tutorial [310] served as a starting point for the DCGAN structure and was gradually adjusted to work with the given training data set. For larger image resolutions, the DCGAN structure by Milad Hasani [311] was found to work well. Inspired by this, a network structure for even larger resolutions was developed (cf. Figure 3.28). The generator takes a Gaussian sampled noise vector with 100 entries and uses a combination of transposed convolutional layers, batch normalization and ReLU activation to output images with a resolution of 256 px $\times$ 256 px. Tanh was used as final activation. In the discriminator, convolutional layers, batch normalization and Leaky ReLU activations were used. The latter was replaced by a Sigmoid activation in the final layer. The entire code can be found in the according GitHub repository linked in Chapter 9.

[12] This section is extracted from the author's publication [275] with only slight linguistic changes.

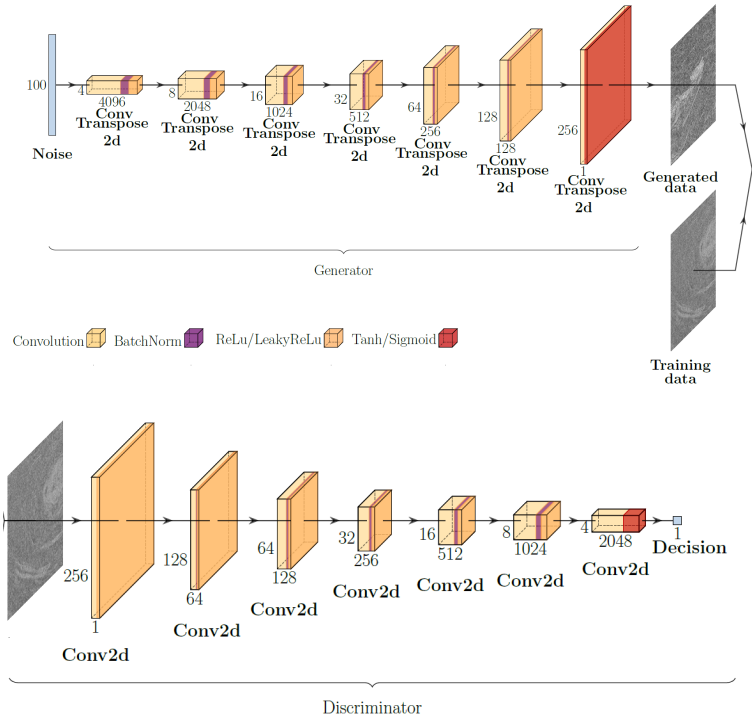


Figure 3.28: Graphic of the final DCGAN network architecture. Designed with the help of the latex code in the GitHub repository of Haris Iqbal [291]. Extracted from Blarr et al. [275].

In a first step, all images from the input data set are loaded as PyTorch tensors (one 3D array per image of x value, y value and gray value) and normalized to values in the range $[-1, 1]$ to avoid coefficients equal to 0. The image tensors are then shuffled and divided into individual batches. The last non-full batch is dropped, i. e. the images are not used for training. Unlike in the original DCGAN paper [250], no initial weights were defined as this was found to increase the likelihood of a stable training process for this particular configuration. The training loss is calculated with the ADAM optimizer based on the commonly used BCE. Setting the ADAM optimizer parameters to $(\beta_1, \beta_2) = (0.5, 0.999)$ was found to improve the stability of training just as described by

Radford et al. [250]. For multi-GPU parallelization and therefore a faster training, NVIDIA CUDA with the "DataParallel" method of PyTorch is used [310].

The actual training process follows this scheme: First, one batch of generated images is created based on random noise vectors and associated with the corresponding labels (0 for generated). These images are then fed to the discriminator together with one batch containing only real images from the training data set (labeled 1 for real). Based on the average of the calculated losses (following Equation (2.123)) on both batches, the discriminator biases are updated. As a next step, again Gaussian sampled random noise vectors are passed to the generator which outputs one batch of generated images. Based on the discriminator feedback on these images, the generator is then updated. Every epoch consists of multiple iterations and ends if the whole input data set was processed this way. It shall be mentioned that the losses were saved after every single iteration, whereas the Fréchet inception distance (FID) was only calculated after every epoch in order to avoid slowing down the training too much. Additionally, after every epoch a number of generated images based on fixed noise vectors was saved in order to visually analyze the training progress later on. Due to the computational effort needed, the image quality assessment was performed in a subsequent step.

3.8.4 Quantitative quality metrics [13]

Apart from the visual and hence qualitative evaluation of the generated images, multiple quantitative measures have been developed in recent years in order to assess the performance of a generative network [312]. Of those, the FID and nearest neighbor evaluations based on two different metrics were conducted that are outlined in the following.

[13] This section is extracted from the author's publication [275] with only slight linguistic changes.

3.8.4.1 Fréchet inception distance (FID) [14]

The Fréchet inception distance [313], as an advancement of the inception score [314], is a metric to determine the difference between feature vectors of generated samples and real training images. It is based on the Inception v3 Network [315] that is pre-trained on the ImageNet [316]. The FID compares the activations from the penultimate layer of the inception network of real $p_r(\boldsymbol{x})$ and generated $p_g(\boldsymbol{x})$ images [317]. The distributions of these real and generated images are thereby modeled as multi-dimensional Gaussians that are defined by their mean μ and covariance $cov(x, y)$. The distance is hence defined by

$$\begin{aligned} &d^2\big((\mu_\mathrm{r}, cov_\mathrm{r})(\mu_\mathrm{g}, cov_\mathrm{g})\big) = \\ &\quad |\mu_\mathrm{r} - \mu_\mathrm{g}|^2 + \mathrm{tr}\big(cov_\mathrm{r} + cov_\mathrm{g} - 2(cov_\mathrm{r} cov_\mathrm{g})^{1/2}\big). \end{aligned} \tag{3.18}$$

Thus, a lower FID value corresponds to a smaller distance between the two distributions of real and generated data. The FID values are dependent on the corresponding resize or compression operations and can even improve for higher compressed images (i.e. poorer resolution) [318]. The FID values must therefore be regarded as a benchmark for the quality of images created using the same network in different epochs and are of limited suitability for comparing different networks. The FID was calculated of 128 generated images per epoch.

3.8.4.2 Nearest neighbor procedures [15]

A question that arises quite naturally is whether a network reproduces only the training data, which involves overfitting. To address this, a k-nearest neighbor search is performed, which calculates the k nearest neighbors from the entire training data set to a given generated image. This is equivalent to finding the

[14] This section is extracted from the author's publication [275] with only slight linguistic changes.
[15] This section is extracted from the author's publication [275] with only slight linguistic changes.

images in the training data set that have the smallest distance to the generated image based on a suitable distance measure. In the following, we briefly introduce the Euclidean distance and the computationally more expensive structural similarity index measure (SSIM), as these serve as the basis of the k-nearest neighbor search in the results of this paper.

- Euclidean distance (ED)
  The Euclidean distance being one of the simplest ways to determine the similarity between two arrays and the most common distance measure for nearest neighbor search [319] can be calculated for two gray scale images $\boldsymbol{x} = (x_1, x_2, ..., x_{MN})$ and $\boldsymbol{y} = (y_1, y_2, ..., y_{MN})$ of size M px $\times$ N px as:

$$d_{\mathrm{E}}^2(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{MN} (x_k - y_k)^2 \tag{3.19}$$

  with the gray levels at the location (k, l) given as x_{kN+l} and y_{kN+l}. The smaller the calculated value, the higher the similarity. As there are major limitations like the non-consideration of spatial relationships, the nearest neighbors found through the calculation of the ED sometimes do not match human perception. Therefore, a second distance metric is considered. The ED is still calculated due to its computational simplicity.

- Structural similarity index measure (SSIM)
  Another option for the evaluation of nearest neighbors of GANs is the SSIM [312] introduced by Wang et al. in 2005 [319]. It focuses especially on factors that are also relevant to human perception through evaluating luminance L, contrast C and structure S. Those three aspects are separately mathematically defined through means μ_x and μ_y, standard deviations σ_x and σ_y and cross-correlation coefficient σ_{xy} of the two images $\boldsymbol{x}$ and $\boldsymbol{y}$. The detailed formulas can be found in the paper by Wang et al. [319]. In order to avoid instabilities for small values, constants C_i are added.

The product of these three quantities relatively weighted through three power parameters $\alpha > 0,\ \beta > 0$ and $\gamma > 0$ results in the $SSIM(\boldsymbol{x}, \boldsymbol{y})$. Choosing $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$ leads to the following SSIM formula:

$$SSIM(\boldsymbol{x}, \boldsymbol{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{3.20}$$

The values range between (-1, 1] with a value of one corresponding to optimal similarity, hence equality of the images. It shall be mentioned that the SSIM is usually not calculated globally but instead inside of a Gaussian window covering a local square patch which slides pixel by pixel across the entire image. In the so-called mean-SSIM or MSSIM all local values are summed up and divided by the total number of windows m in order to obtain one single quality measure. Conventionally, the MSSIM is often referred to as just the SSIM, which will be handled similarly in this work.

3.8.5 Overview of all chosen parameters [16]

All values of important parameters are summarized in the following list.

- Image resolution — $256\,\text{px} \times 256\,\text{px}$
- Learning rate r_1 — 0.0001
- ADAM optimizer — $(\beta_1, \beta_2) = (0.5, 0.999)$

[16] This section is extracted from the author's publication [275] with only slight linguistic changes.

- Duration of training 75 epochs

- Number of input images
  $$n_{\text{scans}} \times f_{\text{cut}} \times f_{\text{augmentation}} = n_{\text{final}}$$
  $$1,830 \times 4 \times 4 = 29,280$$

- Images per batch 128

- Number of iterations per epoch (the last non-full batch is dropped)
  $$29,280/128 = 228.75.$$

Some of the parameters such as the learning rate were chosen by incrementally increasing the learning rate in parameter sweeps and analyzing the respective result.

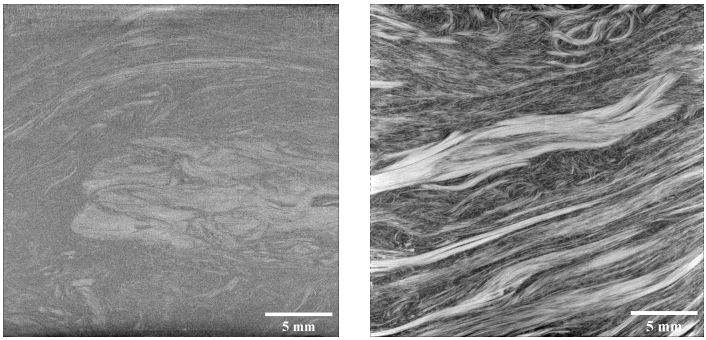
4 Results

This chapter first addresses microstructure characteristics that were noticeable in the CT scans. Then, in the same order as in the Methods chapter, the results of the plastificate examinations, the fiber volume contents and the newly developed methods for their evaluation and the fiber length distributions are presented. This is followed by the results of the fiber orientation distributions, the fiber orientation tensor interpolation and the image generation by the GAN.

4.1 Characteristics of microstructure

Looking at example images of scans in Figure 4.1, the significantly better image quality for the glass fiber reinforced specimens is visible. Owing to their larger diameter, even single fibers are recognizable in the matrix in Figure 4.1(b). However, this visual perception is challenging to convert into a quantitative metric. As an example: The signal-to-noise ratio of the example image in Figure 4.1(b) is with 2.95 lower than for the carbon fiber reinforced specimen in Figure 4.1(a) with 5.74. For this reason, different kinds of metrics were used for the evaluation of the images in the image generation section of this thesis (cf. Section 4.7).

Furthermore, there is a strong tendency towards fiber bundle formation, both in the CF and GF reinforced material. A particular effect on the material properties due to these quite large bundles is to be expected, as well as difficulties in using known material modeling approaches for their prediction. It is especially



(a) (b)

Figure 4.1: An image of a slice of the scan of specimen $CF_{1,\mathrm{carbon}}$ of Plate 3 (CF-PA6) (a) and of specimen $CF_{1,\mathrm{glass}}$ of Plate 4 (GF-PA6) (b).

interesting that these fiber bundles often include unimpregnated areas, as can be seen in Figure 4.2.

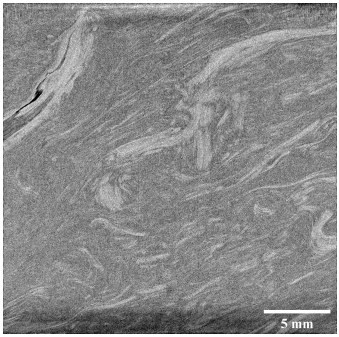


Figure 4.2: Example of an unimpregnated fiber bundle (blackish area in whitish fiber bundle area in the upper left corner) of specimen $F_{3,\mathrm{carbon}}$ of Plate 3 (CF-PA6).

These matrix-free areas or at least areas with poor fiber-matrix bonding are often the starting point for failure of the sample in tests, e.g., tensile tests, with matrix-fiber bonding strength being one of the two major critical failure modes

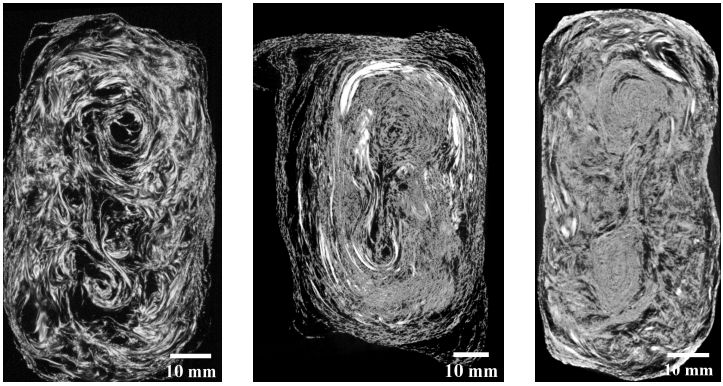
of DicoFRP (together with fiber breakage) [5]. It can also be assumed that they have an influence on the modulus of elasticity even in the linear-elastic range.

Further microstructure characteristics associated with specific microstructure quantities like FVC of FOD will be discussed in the respective sections on those quantities (cf. Section 4.3 and Section 4.5).

4.2 Plastificate investigations

In order to enhance the understanding of the process and the modeling thereof, scans of the plastificates, as the basis of the later plates, were conducted. The material orientation in two planes was investigated as well as the porosity of the initial charges.

As can be seen in Figure 4.3, the material constituents influence the shape, porosity and behavior of the plastificate.



(a) (b) (c)

Figure 4.3: Sections of a scan of a CF-PA6 (a), a GF-PA6 (b) and a GF-PC (c) plastificate for comparison.

While the combination of PA6 and CF (a) shows many air inclusions in the interior, the GF-PA6 (b) appears to have a somewhat protruding, loose surface layer. If, on the other hand, a different matrix material, polycarbonate, is selected, the plastificate (c) appears much more compact overall. A less compact plastificate shape, as occurs with the material used in this work, could also complicate the assessment of the influence on the finished plate.

As expected, it is not possible to differentiate between fiber and matrix in the first case, but this also appeared to be difficult with the glass fibers, which is why the orientation must be regarded as material orientation and not fiber orientation in the following. This is based on the reasonable consideration that the fibers align themselves along the matrix material, which has already been described in the Methods (cf. Section 3.3).

4.2.1 Initial orientation

With y being the extrusion direction, the orientation in the x-z plane was first of interest. In the 3D CT image, a quarter of the plastificate was therefore segmented into 15 cubes with dimensions of 10 mm $\times$ 10 mm $\times$ 10 mm (repeated at different y-positions). Evaluating the orientation with the structure tensor approach and plotting the resulting orientation tensors as glyphs, led to the result in Figure 4.4. The results of the quarter was mirrored vertically, horizontally and both vertically and horizontally at the same time to cover the remaining three quarters with the tensor glyphs as well assuming symmetric behavior. This assumption was made based on own observations and literature [320–322].

It is visible that the tensors in the corner are rather oriented in extrusion direction. In general, the tensors tend to get more isotropic towards the middle of the plastificate. Other than that, the two swirls of the twin screw extruder that are visible in the CT scan are reproduced by the glyphs quite well. Zooming in on this swirl area in Figure 4.5 shows that in more detail.

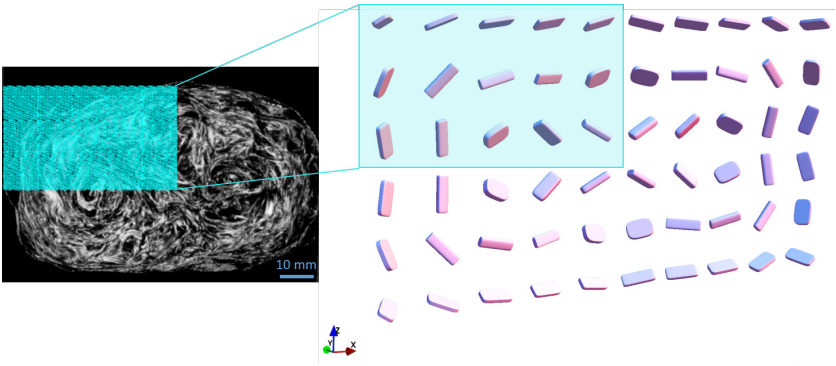


Figure 4.4: Glyphs of fiber orientation tensors evaluated in a quarter of a CF-PA6 plastificate and mirrored. The chosen grid consists of 3 × 5 cubes with dimensions of 10 mm × 10 mm × 10 mm. Glyph plots by Louis Schreyer in collaboration for joint proceeding [276].

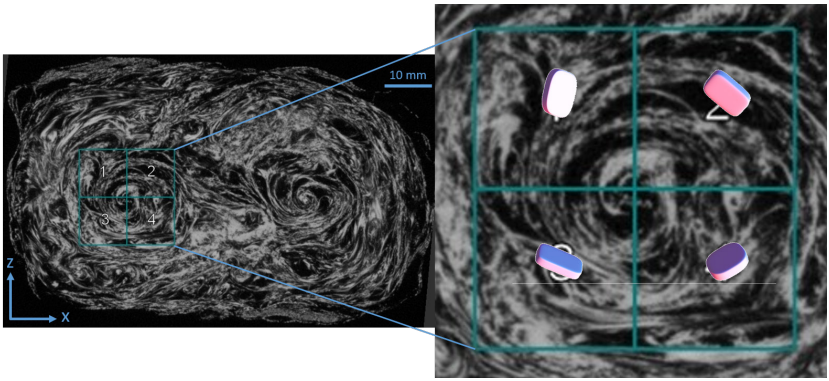


Figure 4.5: Glyphs of fiber orientation tensors evaluated around the twin screw extruder swirl of a CF-PA6 plastificate. The chosen grid consists of 2 × 2 cubes with dimensions of 10 mm × 10 mm × 10 mm. Glyph plots by Louis Schreyer in collaboration for joint proceeding [276].

In order to analyze that in more detail, a finer 6 × 6 grid with cubes of a side length of 3.33 mm was evaluated, which can be seen in Figure 4.6.

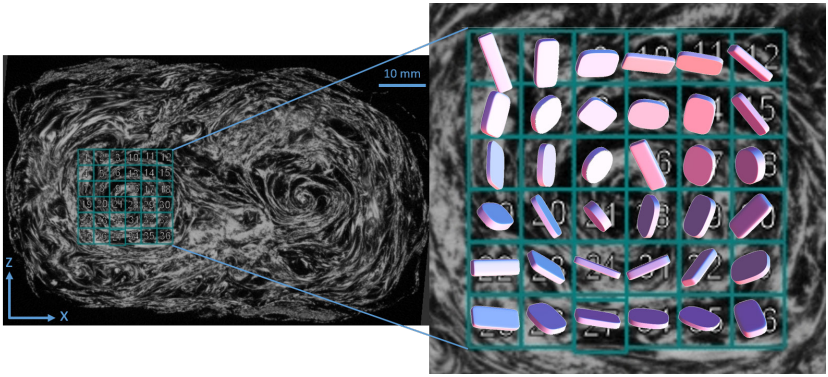
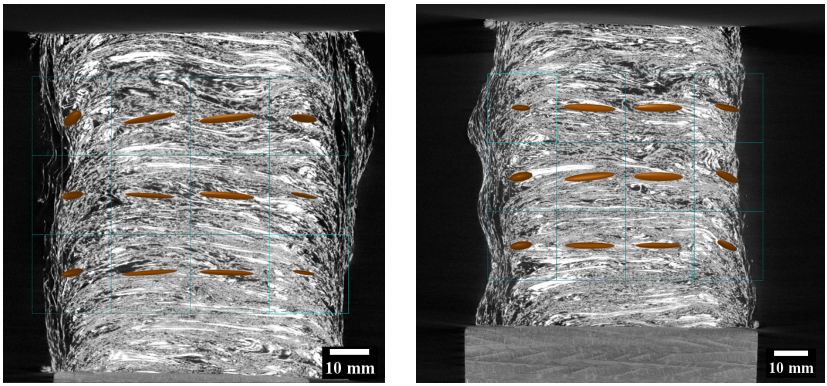


Figure 4.6: Glyphs of fiber orientation tensors evaluated around the twin screw extruder swirl of a CF-PA6 plastificate. The chosen grid consists of 6 × 6 cubes with dimensions of 3.33 mm × 3.33 mm × 3.33 mm. Glyph plots by Louis Schreyer in collaboration for joint proceeding [276].

The tolerably consistent circular orientation enabled a simpler reconstruction of the initial orientation state based on geometric considerations, which will be elucidated in the Discussion (Section 5).

To evaluate the orientation behavior across the extrusion direction y, slicing in the x-y plane was necessary. At mid-level z-position a grid of 3 × 4 cubes of 20 mm × 20 mm × 20 mm. The result can be seen in Figure 4.7.

Both outer layers (considering the x direction) have higher values concerning the 33 component of the tensor, hence are oriented more towards the z-direction, which coincides with the findings of the evaluation in the other plane before. While they also show a slightly higher 22-component, thus are more oriented in y-direction (extrusion direction), the overall tensor behavior is very 11-component (x-direction) dominant. This again simplifies a geometric reconstruction, compared to similar studies where authors found a more significant orientation in y-direction, which will also be discussed in Section 5.



(a) (b)

Figure 4.7: Sections of a scan of two different parts of the same GF-PA6 plastificate with their orientation tensors plotted as tensor glyphs superimposed. Glyphs of fiber orientation tensors evaluated in the x-y plane of a GF-PA6 plastificate. The chosen grid consists of 3×4 cubes with dimensions of $20\,\text{mm} \times 20\,\text{mm} \times 20\,\text{mm}$.

4.2.2 Porosity and volume

Considering the porosity of the plastificates it was of particular interest to understand the influence of process parameters on it, as well as the course of the porosity across the extrusion direction of the plastificate.

In Figure 4.8, the binarized image (a), the detection of the plastificate area (b) and the thereof determined pores in the plastificate (c) by the algorithm described in the Methods (Section 3.3) are depicted.

While in the example image, the detection of the outer bounds of the plastificate worked well, it was more difficult for the polyamide 6 based plastificates that exhibited more fringed surfaces (compare Figure 4.3).

Evaluating the porosity across the extrusion direction and the average porosity in percent for different configurations of plastificates enables conclusions that coincided with process observations. In Figure 4.9, four porosity plots can be seen.

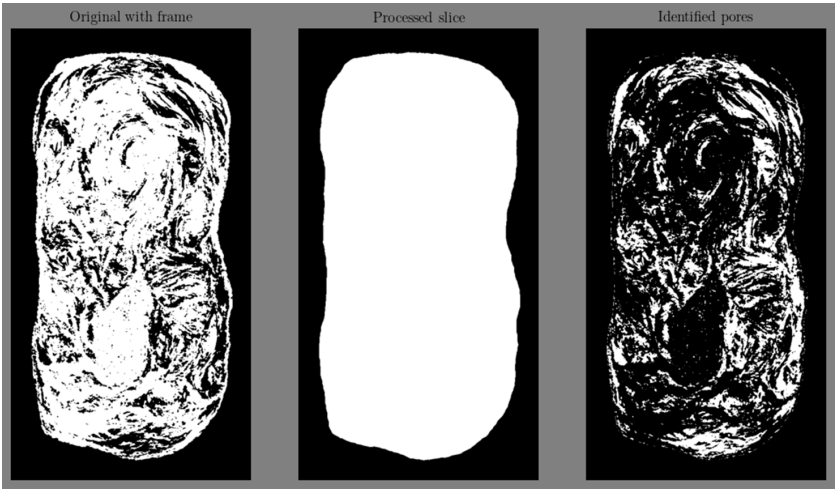


Figure 4.8: Example image of a slice of a binarized GF-PC scan at the left and the area that the algorithm identifies as the plastificate in white in the middle. At the right, all the detected pores are marked white.

Plots (a) and (b) show the porosity curves of two halves of the same GF-PA6 plastificate. In (a) you can see the end that came out of the twin-screw extruder nozzle first, i.e. the "front end", and in (b) the "back end", which was extruded last. It is noticeable that the average porosity (red dotted line) of the front end is significantly higher at 0.4310 than that of the back end at 0.3746. This is consistent with observations from process trials in which the plastificate appeared to expand (also known as lofting) when lying on the roller conveyor for a long time. This is clearly visible in plot (a), where towards the oldest end (high layer numbers) both the area occupied by the plastificate in the slice (on average 0.8001) and thus three-dimensionally the volume in the overall scan, as well as the porosity increases. The average area covered in plot (b) is slightly smaller at 0.7886. Incidentally, when calculating the porosity or the volume taken up, 20 slices were cut away at the edge of the 3D scans, as these had artifacts or already showed areas rich in air due to unevenness, etc. The area and volume comparisons are only valid because both plastificate halves were

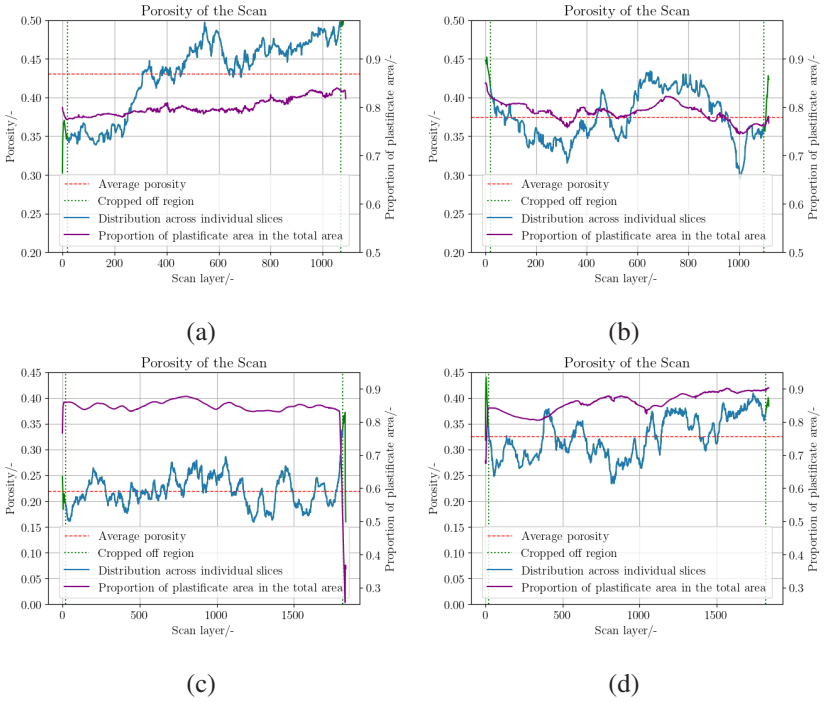


Figure 4.9: Course of the porosity of the "front end" (a) and "back end" (b) of a GF-PA6 plastificate. In addition, course of the porosity of two GF-PC plastificates, one manufactured with a screw speed of 45 rpm (c) and one with 100 rpm (d).

scanned in one scan and therefore have the same scan section and the same resolution.

Looking at the plots (c) and (d), the porosity curves of two separate GF-PC plastificates produced under different conditions can be seen. The plastificate shown in plot (c) was produced at a screw speed of 45 rpm, while the plastificate shown in plot (d) was produced at 100 rpm. The plastificate produced at slower speed shows a significantly lower mean porosity of 0.2196 compared to 0.3248 for the faster extruded plastificate. The low speed and porosity in (c) also ensures a relatively consistent porosity and volume profile across the slices,

while the higher speed in the production process in (d) results in an increase in porosity and volume towards the end.

4.3 Fiber volume content

In this section, the results of the experimental determination of FVCs of specimens are shown, as well as results of their computational determination from the scans of specimens with self-implemented, novel methods.

4.3.1 Chemical dissolution [1]

The nominal fiber volume content set during production was 25 %. Since the samples that were dissolved in acid of Plate 2 (cf. Figure 3.4) were of different sizes, the dependence of the fiber volume contents on the sample size can be shown. In fact, this effect seems to be clearly pronounced, as can be seen in Table 4.1: the fiber volume contents increase monotonically from Sample 1 to 3, as well as from 10 to 12. With Specimen 1 being the smallest (10 mm $\times$ 10 mm $\times$ 3 mm), Specimen 2 being the second largest (20 mm $\times$ 20 mm $\times$ 3 mm) and Specimen 3 being the largest (30 mm $\times$ 30 mm $\times$ 3 mm), it is noticeable that the larger the specimen, the larger the FVC. Moreover, the fiber volume content seems to be higher in the charge region (Specimen 1, 2, 3) than in the transition region (Specimen 10, 11, 12) - a finding that contradicts the results of the samples of Plate 3 (cf. Table 4.2). The results of the FVC in Table 4.2 of Plate 3 have been first published by Scheuring et al. [274]. While the average FVC of the three charge specimens (C_1, C_2, C_3) is equal to the global average with 23.57 %, it is indeed slightly higher than the average FVC of the three specimens of the flow region (F_1, F_2, F_3) with 22.74 %. The transition region

[1] This section is extracted from the author's publication [188] with only slight linguistic changes.

(specimens CF_1, CF_2, CF_3) shows the biggest average FVC with 24.42 % in this plate.

Table 4.1: FVC in % for specimens of Plate 2 (cf. Figure 3.4) determined through acid-based dissolution of the matrix. Extracted from Blarr et al. [188].

Specimen	FVC
FLD_1	22.3 %
FLD_2	25.5 %
FLD_3	28.6 %
Average	25.5 %
Standard deviation	2.6 %
FLD_{10}	17.9 %
FLD_{11}	24.0 %
FLD_{12}	26.6 %
Average	22.8 %
Standard deviation	3.6 %
Overall average	24.2 %
Overall standard deviation	3.4 %

4.3.2 "Average or above" (AOA) thresholding [2]

The scans, that the FVC had to be determined of, all showed *salt and pepper* noise. So the first step of the implemented thresholding method was the application of a filter. The choice of the filter type and kernel size was decided on after all other steps of the procedure were defined. A study was conducted, as to which filter and kernel size produces an FVC closest to the measured ones. Finally, the median blur led to the least average deviation between the

[2] This section is extracted from the author's publication [188] with only slight linguistic changes.

Table 4.2: FVC in % for specimens of Plate 3 (cf. Figure 3.4) determined through acid-based dissolution of the matrix. Data from Scheuring et al. [274]. Extracted from Blarr et al. [188].

Specimen	FVC
F_1	23.07 %
F_2	22.08 %
F_3	23.06 %
Average	22.74 %
Standard deviation	0.46 %
C_1	25.57 %
C_2	22.31 %
C_3	22.81 %
Average	23.56 %
Standard deviation	1.43 %
CF_1	26.36 %
CF_2	23.10 %
CF_3	23.81 %
Average	24.42 %
Standard deviation	24.42 %
Overall average	23.57 %
Overall standard deviation	1.37 %

calculated and the measured results. It is noticeable, that there are differences of the best kernel size between the two different plates. For the specimens of Plate 2, consisting of FLD_1 to FLD_{12}, a median blur filter and a kernel size of 15 $\times$ 15 px showed the least average deviation, as well as the least maximum deviation. The results can be seen in Table A.3, Table A.4 and Table A.5 in the Appendix. For the specimens of Plate 3, consisting of C_1 to F_3, a median blur and a kernel size of 23 pixel $\times$ 23 pixel had the least average deviation, as well as the least maximum deviation, which is shown in Table A.6 and Table A.7. In

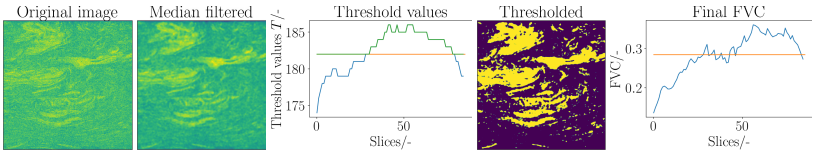


Figure 4.10: Illustration showing the whole process of FVC determination by AOA thresholding using the example scan of sample FLD_3. (a) Shown is a middle slice of the original 3D CT image read into Python. (b) The same slice is shown after applying the median filter (kernel size in this case 15). (c) The Otsu threshold of each slice was determined and plotted over the slices. The calculated thresholds are shown in blue, the average in orange and the value that is ultimately applied in green. The average value is used if the actual threshold of the slice is below the average value, otherwise the calculated value above it is used (cf. Figure 3.14). (d) The slice is shown with the threshold applied. (e) Finally, the fiber volume content determined by the method is plotted over the slices. The decreasing volume content at the edges is clearly visible. The mean value over all slices in orange is the final determined value, which is again compared with the experimentally determined value. Based on Blarr et al. [188].

the tables with the results of Plate 3, only the tested kernel sizes that seemed the most relevant of some filters are listed. Comparing the dimensions of the images of the different plates, to be seen in Table 3.2, it shows that the average image size of Plate 3 is larger than that of Plate 2, hence the image resolution is higher for Plate 3 (cf. Table 3.1). Apparently the needed kernel size of the median blur is dependent on the dimensions and the resolution of the scan used. The effect of the median blur is visible comparing the images (a) and (b) in Figure 4.10. The determined binary image after applying the AOA thresholding method of this particular slice is shown in (d).

However, not all final results of the FVC are convincing (compare Table A.4, column "medianBlur(...,15), and Table A.6, column "medianBlur(...,23)). While most samples show decent results, when not including outliers, with a relative average deviation of 1.81 % in the first measurement series and 3.42 % in the second measurement series, there are samples with a much larger deviation. Individual samples show relative deviations of up to 116.09 % (FLD_{10}). The reason for those differences of the performances of the algorithm are not entirely clear. FLD_{10} was a small specimen and a scan with low resolution, which is

not the best combination in general. This condition arose because it was desired to have the same resolution for all samples of one plate and the low resolution was necessary in order to fit the biggest specimens inside of the beam path. Hence, the image quality was insufficient but that was also the case for FLD_1, which did not deviate that massively. Additionally, FLD_{10} showed a much lower experimentally measured FVC and minor measurement uncertainties were assumed to lead to that value. For these reasons FLD_{10} was left out of the error calculations of the AOA thresholding and was not used as training data in the CNN either. However, the other two outliers C_2 and F_2 were included. Neither any visual deviation nor deviations in the histograms, brightness levels or other measures used for image comparison in these two scans could be detected, which is why they are included in the absolute average deviations and were also used as training data for the CNN. That way, the absolute average deviation of the final AOA thresholding with medianBlur filter of 15 and 23, respectively, amounts to about 2.7 %. The final results are depicted in Table 4.3.

Table 4.3: Predictions of final AOA thresholding method (compare green columns in Table A.4 and in Table A.6) and of final CNN structure. The values for the CNN are averaged predictions for the original and all augmented 3D image versions with the same FVC. Extracted from Blarr et al. [188].

Specimen	**Exp. values**	**AOA thresholding**		**CNN**	
	FVC	FVC	Absolute deviation	FVC	Absolute deviation
FLD_1	22.30 %	22.97 %	0.67 %	26.04 %	3.74 %
FLD_2	25.50 %	24.18 %	1.32 %	28.62 %	3.12 %
FLD_3	28.60 %	28.5 %	0.1 %	28.41 %	0.19 %
FLD_{11}	24 %	23.91 %	0.09 %	25.24 %	1.24 %
FLD_{12}	26.60 %	26.64 %	0.04 %	25.6 %	1 %
C_1	23.07 %	24.07 %	1 %	23.24 %	0.17 %
C_2	22.08 %	42.2 %	20.12 %	22.85 %	0.76 %
C_3	23.06 %	23.04 %	0.02 %	22.49 %	0.57 %
CF_1	25.57 %	26.6 %	1.03 %	23.12 %	2.46 %
CF_2	22.31 %	23.53 %	1.22 %	22.35 %	0.04 %
CF_3	22.81 %	22.53 %	0.28 %	22.80 %	0.01 %
F_1	26.36 %	25.48 %	0.88 %	21.98 %	4.37 %
F_2	23.10 %	32.76 %	9.66 %	24.78 %	1.68 %
F_3	23.81 %	25.11 %	1.3 %	22.68 %	1.12 %
Mean	**24.23 %**	**26.54 %**		**24.3 %**	
Abs. aver. dev.			**2.7 %**		**1.46 %**

The final results of the FVC determined by AOA thresholding after applying the median filter as described above are also plotted in Figure 4.11. The two deviating values of $C_{2,\mathrm{carbon}}$ and $F_{2,\mathrm{carbon}}$ can be clearly detected.

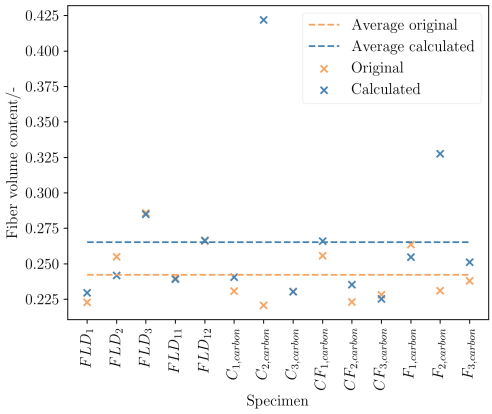


Figure 4.11: Original (orange) and calculated with the novel thresholding technique (blue) values of FVC for the fourteen specimens, as well as measured averaged FVC (orange dashed) and calculated averaged FVC (blue dashed). Based on Blarr et al. [188].

Considering the progress of the fiber content across a specimen by the example of specimen F_2, a clear non-monotonous course can be seen in Figure 4.12. The lower values at the borders of the sample and the increase of the FVC towards the center have been expected. This behavior appears due to the compression molding process, which will be further elaborated on in the Discussion.

4.3.3 Convolutional neural network [3]

As shown in Figure 4.13, using 40 epochs as training of the CNN is sufficient since the model's validation and training losses converge after around 15 epochs and there are no significant improvements beyond that point. The training loss

[3] This section is extracted from the author's publication [188] with only slight linguistic changes.

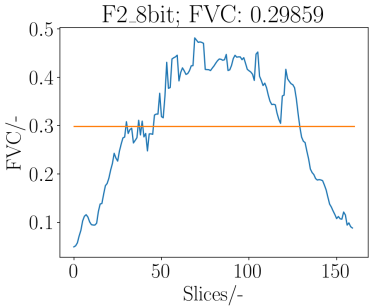


Figure 4.12: Exemplary progression of FVC over thickness of the specimen considering the example of specimen F_2. Typical increase of FVC towards the center of the specimen visible. Extracted from Blarr et al. [188].

value after 40 epochs amounts to 0.5539 and the validation loss to 0.5535. The initially unusual fact that the loss in the training data is greater than in the validation data is discussed in more detail in Section 5.2.3. Considering the shift of the minimum attainable loss value briefly discussed in the Methods (Section 3), this can be considered a successful training process.

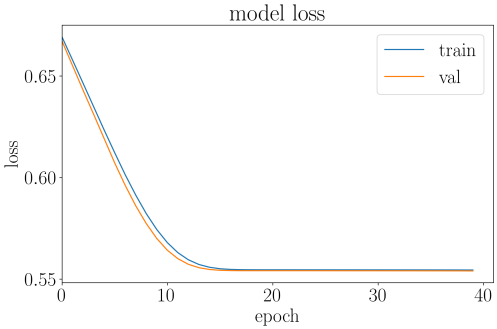


Figure 4.13: Final graph showing the losses for the testing and validation data sets during the training process as a function of the epoch. Extracted from Blarr et al. [188].

The performance of the CNN is further assessed in Figure 4.14. Of the 448 3D input scans, 299 were used for training and 149 for validation in accordance with the two thirds split. Figure 4.14 shows the mean values of the predicted FVC of

all validation scans with the same experimentally measured FVC (14 different ones) (including original and augmented scans). Thus, for five of the points, the mean value was determined from ten predicted values and for the remaining nine from eleven, resulting in 149 validation scans. The standard deviation was so small, with a maximum value at one point of 0.0009 and a mean value over the 14 points of 0.0005, that no boxplot was used, but the same format as in Figure 4.11 with the thresholding results. The network thus calculates stable consistent values for a given scan with the same FVC, regardless of whether it was augmented or not. This indicates a good geometric independence of the network. In addition, several runs of the CNN training were made with the same settings and the results also hardly differed, which indicates a low intrinsic uncertainty of the CNN. The biggest variation between runs was rather at which epoch the loss plot reached its minimum, which could vary by 5-10 epochs.

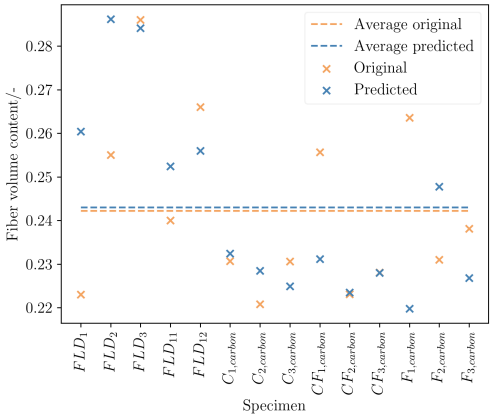


Figure 4.14: Original (orange) and calculated with the CNN (blue) values of FVC for the fourteen specimens, as well as measured averaged FVC (orange dashed) and calculated averaged FVC (blue dashed). Extracted from Blarr et al. [188].

In the best universal case, hence the final network depicted and described in Section 3.4.2.3, an absolute average deviation of 1.46 % was achieved (cf. Table 4.3).

Beside the amount of epochs used to train the network, a number of variables were tuned to improve the prediction's accuracy. The augmentation process, e.g., has multiple steps of flipping and rotating images. Experiments with using less or more augmentation steps showed that using the most amount of steps showed the best results. Changing which scans were fed into the program from the start had a large impact and by down-selecting systematically, an absolute average deviation below $0.9\ \%$, so a performance increase by about $40\ \%$ was achieved. However, since the goal of this network is to provide a universal method for CF-PA6 sample scans, all data was used instead, in favor of general applicability to different scans at a later time. FLD_{10}, which has already shown to be an outlier for the thresholding method, poses an exception in this case. Its implementation caused a remarkable decrease in performance in all cases, for which an identifiable rationale is lacking and thus this scan had to be removed from the entire data set, as mentioned before.

4.4 Fiber length distributions

In Figure 4.15, the length distributions of the samples FLD_1 to FLD_3 and FLD_{10} to FLD_{12} of Plate 2 (cf. Figure 3.4(b)) can be seen.

The graph was truncated at a fiber length of 6.5 mm, in the knowledge that occasional fibers up to a length of 32 mm do occur. In general, fibers appear in an extremely wide range of lengths, but the majority of the fibers are in the region between 0 mm and 1 mm. Especially the three samples from the charge area, i.e. FLD_1 to FLD_3 show strong peaks in the range between 0.3 mm and 0.5 mm approximately. Of all the fibers measured from the six samples, more than 58,000 individual fibers were measured, which together have an average value (in relation to the total number) of 1.07 mm. However, the median is 0.48 mm. Here, it is already noticeable that the choice of the mean value determination can have a significant influence on the result of a possible modeling of the material. The aspect ratio is included in, for example, the Mori-Tanaka and Halpin-Tsai homogenization approaches, and the resulting

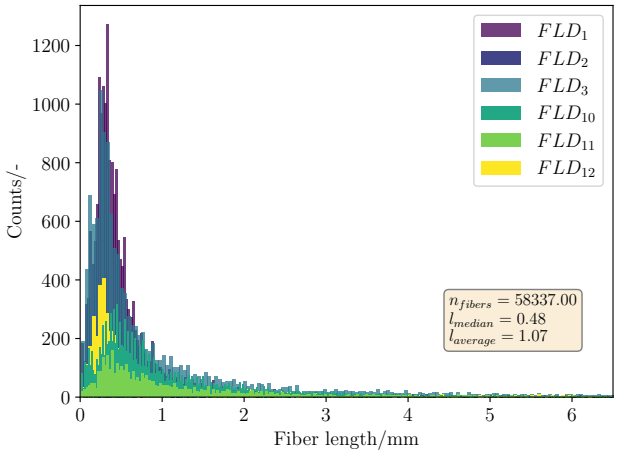


Figure 4.15: Histogram of amount of fibers of a specific length in mm per specimen, including FLD_1 to FLD_3 and FLD_{10} to FLD_{12}.

mechanical properties are therefore dependent on the mean fiber length. In relation to the individual sample, the average and median values can be seen again in Figure 4.16.

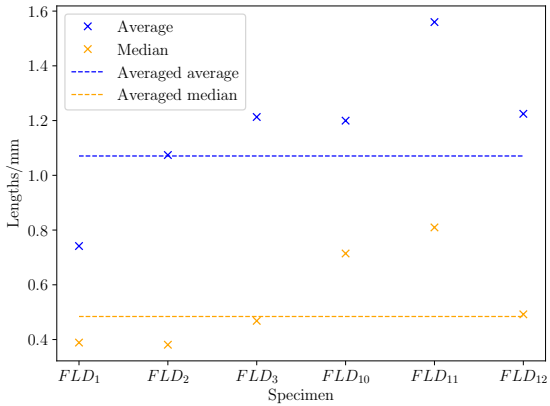


Figure 4.16: Average (blue) and median (orange) fiber lengths per specimen (FLD_1 to FLD_3 and FLD_{10} to FLD_{12}) and averaged across all specimens (dashed blue and orange).

It is noticeable that the mean values of the samples in the charge area (FLD_1 to FLD_3) are on average lower than those of the samples in the flow area (FLD_{10} to FLD_{12}). In addition, the fiber lengths increase significantly from the smallest (FLD_1 and FLD_{10}) to the next larger sample (FLD_2 and FLD_{11}) when looking at the average. It is assumed that for a sample area of 10 mm $\times$ 10 mm, very long fibers are already cut off, which is hardly likely to be the case for 20 mm $\times$ 20 mm. From the medium (FLD_2 and FLD_{11}) to the largest samples (FLD_3 and FLD_{12}), the average fiber length increases only slightly, or even decreases in the case of the flow samples. Therefore, specimen dimensions of 25 mm $\times$ 25 mm were chosen for the specimens of Plate 3, since it may be assumed that the influence of sample size on the fiber length distribution should be saturated approximately in the range between the middle and largest samples. However, median and average again behave differently. Thus, the sample size effect cannot be clearly shown, but the difference between flow and charge area can be seen for both mean values.

4.5 Fiber orientation

In the following section, all results relating to fiber orientations are presented. First, validation results of the application of the structure tensor based method for fiber orientation evaluation for the material of this thesis are shown. Subsequently, special features of the evaluated fiber orientation tensors over the thickness (z-axis) are addressed. The behavior of the fiber orientation tensors considering the x-y-plane will be taken up in the following Section 4.6 dealing with the interpolation of FOT.

4.5.1 Validation of the application of the structure tensor method

The validation of methods for fiber orientation determination is limited, as it cannot be measured directly by experiment. Pinter et al. [81] created artificial microstructures with given properties and used them to validate their structure tensor approach (also applied in this work). In this respect, the method has already been validated for similar microstructures. However, an exact validation is not possible for the material in this work for the reason that the generation of a microstructure that truly corresponds to the CF-PA6 with its mixture of bundles and single fibers is currently not possible or implementable. There are two further possibilities for validation, one qualitative and one indirect, quantitative. The qualitative approach will be discussed here first. After applying the structure tensor algorithm, a vector-valued image (3 channels) is created that maps the proportion of the orientation in the three spatial directions in each pixel, respectively a large number of orientation vectors that can be derived from this. The detected orientations of the algorithm can in turn be made visible by false colors in the HSV space and placed on the original slices. If the visual impression of the orientation corresponds well with the detected colors/angles, the detection appears to have been successful as shown in Figure 4.17.

Figure 4.17-1(a) shows an exemplary slice from the center of a CF-PA6 scan (in this case sample CF2 from Plate 3, cf. Figure 3.4), Figure 4.17-2(a) of a GF-PA6 scan (CF2 from Plate 4) and Figure 4.17-3(a) of a hybrid reinforced CF-GF scan (also CF2, i.e. the middle sample, analogous slice pattern to Plate 3 and Plate 4). Below are the same images with the false color representation of the orientation detected by the algorithm, where Figure 4.17-3(b) shows the orientation of the glass fibers and Figure 4.17-3(c) that of the carbon fibers of the hybrid reinforced sample. If one compares the colors in 2(b) and 3(b) with the resulting angles in the HSV color circle in subgraph 4, the recognition of the algorithm corresponds very well to the optical impression of the angles in the CT scan. Especially in the purely CF reinforced case, however, it becomes clear that in areas of quite uniform fiber bundle orientation, slight speckle patterns

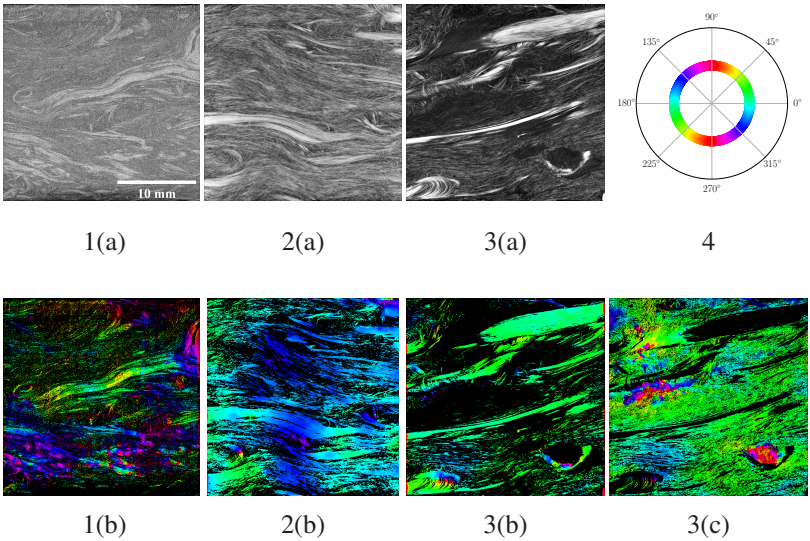
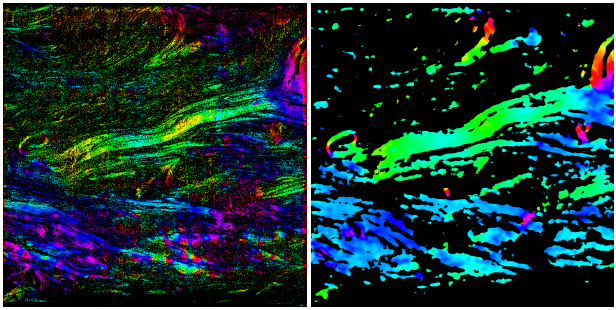


Figure 4.17: One slice out of the middle of the CT scan of the CF2 stack of the CF reinforced plate (1(a)), of the CF2 stack of the GF reinforced plate (2(a)) and of the hybrid CF-GF reinforced plate (3(a)). 4: Color-angle correlation in the HSV ("Hue Value Saturation") color space. 1(b): Detected fiber orientations in the slice presented in 1(a) through the code by Pinter et al. [81] visualized in the HSV color space. 2(b): Detected fiber orientations in the slice presented in 2(a) visualized in the HSV color space. 3(b): Detected fiber orientations of the glass fibers in the slice presented in 3(a) visualized in the HSV color space. 3(c): Detected fiber orientations of the carbon fibers in the slice presented in 3(a) visualized in the HSV color space. Extracted from publication [274], co-authored by the author.

of other colors and thus orientations occur that are not comprehensible. These are due to the greater image noise. An attempt was made to counteract these incorrect detections by applying a median filter of kernel size 10 to the CT scans (before orientation determination) of the purely CF-enhanced samples only. The effect of the filter known for salt and pepper noise reduction is visible in Figure 4.18.

It can be clearly seen that there are almost no more randomly distributed deviating color spots in otherwise homogeneously oriented bundles. However, the detection of very fine structures is also no longer possible. Small bundles



(a) (b)

Figure 4.18: Effect of the median blur applied to a CT scan slice of a CF-PA6 specimen. HSV color image without median filter visible in (a) and with median filter in (b).

and individual fibers are no longer included in the orientation detection at all. Since the detected fiber orientation in the CF-reinforced plates is dominated by the bundles anyway and, as will be shown later, this corresponds well with the mechanical tests, it can be assumed that this provides the more accurate solution.

The other quantitative, indirect type of validation mentioned above is the use of the extracted orientation information in mechanical homogenization models. The resulting stiffnesses or strengths can then be compared with those measured experimentally. As these experimentally measured stiffnesses were carried out with samples taken from the plates at different angles, the accuracy of the stiffness values for different orientations can be used to draw conclusions about the quality of the orientation information used in the modeling. However, since tensor interpolation was used for the orientation information of the modeling, which will not be described until the next section, the procedure and the result will be explained in more detail in Section 4.6.4 and debated in the Discussion in Section 5.4.4.5.

4.5.2 Characteristics of the FOD across the z-axis

The mere display of fiber orientation tensors does not appear to be useful for obtaining a comprehensive understanding of the behavior of the fiber orientation. The results over the plate (x-y plane) are therefore taken up in the visualization form of the tensor glyphs in the context of the fiber orientation interpolation in the next section. However, the behavior over the z-axis is thereby difficult to recognize, which is why characteristic curves of the three main components of the fiber orientation tensors of individual specimens were plotted in Figure 4.19 instead.

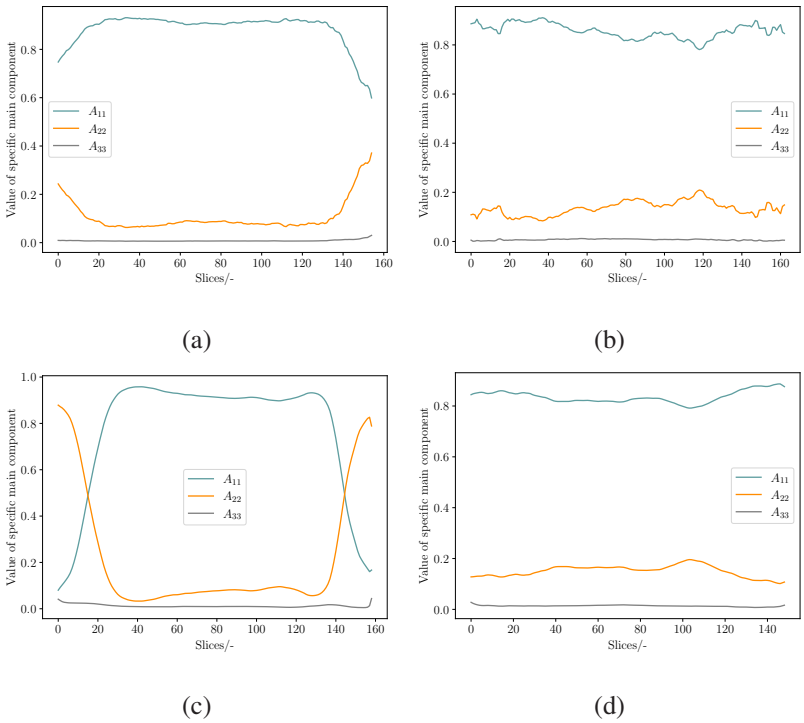


Figure 4.19: Main components over thickness (in slices) of specimens C2 (a) and F2 (b) of Plate 3 (CF-PA6) and of specimens C2 (c) and F3 (d) of Plate 4 (GF-PA6).

The two upper graphs show the results of CF-reinforced specimens, while the lower graphs show the results of GF-reinforced specimens. In addition, the two left-hand plots are each from a specimen in the batch area and the two right-hand plots are each from a specimen in the flow area of the plate. Clearly, an edge effect is visible in both samples in the charge region. The A_{22} component appears to increase at the edge of the specimen and the A_{11} component, which is dominant in the center of the specimen, decreases. In contrast, the curves of the main components in the flow area tend to be constant and are clearly A_{11}-dominant. The question of whether this effect can be called the "shell-core effect", because it is reminiscent of the phenomenon of orientation components changing across the thickness in injection molding, is addressed in the Discussion in Section 5.4. It is also discussed where this changed surface orientation could stem from and, particularly, why it only occurs in the charge area but not in the flow area.

4.6 Interpolation of fiber orientation tensors

The results of the three methods for the interpolation of fiber orientation tensors of second-order are presented below, in the form of the tensors generated in each case, as well as a quantitative evaluation by determining individual tensors that have been measured previously. The specimens of Plate 1 were used (cf. Figure 3.4). These results have already been published in the paper [17]. The results of the slightly modified decomposition method for tensors of fourth-order are presented in the following. However, the use of these results in homogenization methods and the comparison with the experimental stiffness values is included in the Discussion (Section 5.4.4.5).

4.6.1 Component averaging [4]

The set of measured orientation tensors $\mathcal{T}_{\mathrm{m}}$ via CT scan and subsequent calculation via structure tensor is represented by the blue tensor glyphs in Figure 4.20, the set of interpolated tensors $\mathcal{T}_{\mathrm{i}} = \{\boldsymbol{T}_{xy} \ \forall \ x \in 1, ..., 13 \cap y \in 1, ..., 13\}$ by the orange tensor glyphs.

The origin of the global coordinate system is located in the lower left corner of the plate. The original LFT charge covered almost the entire left side of the 400 mm $\times$ 400 mm mold with a width of about x = 90 mm (to the right), a length of about y = 350 mm (up) and a height of about z = 60 mm. Thus, when the press closes, one would expect a quasi 1D flow to the right. However, in Figure 4.20 a clear progression can be seen in the measured (blue) fiber orientation.

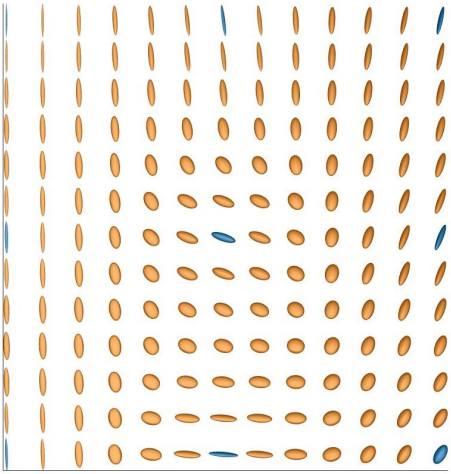


Figure 4.20: Visualization of interpolated (orange) and measured (blue) tensors when using the component averaging interpolation method described in this thesis. Extracted from Blarr et al. [17].

[4] This section is extracted from the author's publication [17] with only slight linguistic changes.

After a clear preferred direction in the left region resulting from the plastificate, i.e., from the last extrusion step in the LFT-D process, a turn to a rather dominant orientation in the x-axis seems to happen towards the middle of the plate (apart from the top region). At the right side another turn to an again more y-direction-dominant orientation happens (apart from the lowest tensor glyph, which in general seems to be more isotropic than the other tensors). Considering these measured tensors, the interpolated tensors should follow some kind of curve. In fact, the interpolation does not seem to cover this behavior smoothly but instead rather accomplished the orientation changes through rounding the tensors. Following the literature, this behavior was expected (cf. swelling effect in the State of Art (Section 2.2.3.3)) and can be confirmed.

In order to be able to approach quantitative error analyses and to better assess the interpolation behavior, one measured tensor of $\mathcal{T}_{\mathrm{m}}$ was omitted in each case and instead also determined with the interpolation method. The visualization results are shown in the nine pictures in Figure 4.21.

To obtain a quantitative error value, the Frobenius norm of the measured tensors and their respective interpolated substitutes was formed. The result of the difference between the Frobenius norm of the interpolated and the original tensor can be seen as an error map in Figure 4.22. The method seems to predict the $\boldsymbol{MM}$, $\boldsymbol{LM}$ and $\boldsymbol{LL}$ tensors the worst. It is difficult to judge whether the Frobenius norm is suitable as a quantitative assessment, but it will be discussed further in the Discussion.

Therefore, a third possibility of error analysis is considered, namely the direct component comparison between interpolated and measured tensors. Figure 4.23 shows this for the component averaging method.

The nine graphs in Figure 4.23 correlate to the nine different tensor components of a 3×3 tensor. Each graph shows the component value of the measured tensors in blue and of the interpolated tensors in orange on the y-axis for each of the nine measured tensors (depicted on the x-axis, starting with $\boldsymbol{UL}$). While the differences are largest for the components with the largest values and fluctuations

Figure 4.21: The graphic shows the measured (blue) and interpolated (orange) fiber orientation tensor glyphs when leaving one measured tensor out of the calculation and interpolating it instead with the component averaging method respectively. Extracted from Blarr et al. [17].

(mainly A_{11} and A_{22}), the character of the component averaging method is also clearly evident in this type of error analysis. Thus, the components of the interpolated tensors almost resemble a moving average of the components of the measured tensors.

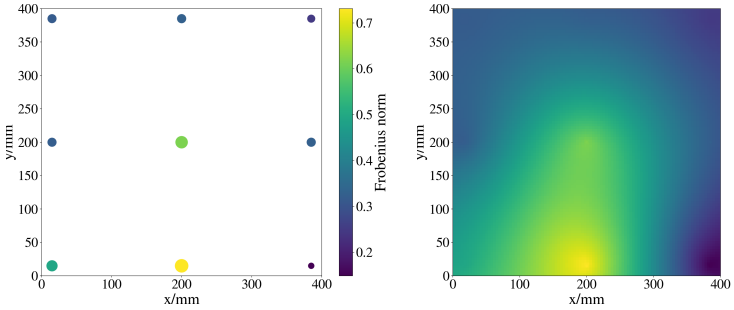


Figure 4.22: Visualization of the discontinuous (left image) and continuous (right image) error of the component averaging method across the plate of the interpolated tensor in comparison to the measured one when leaving this specific tensor out of the computation. Value determined via Frobenius norm. Extracted from Blarr et al. [17].

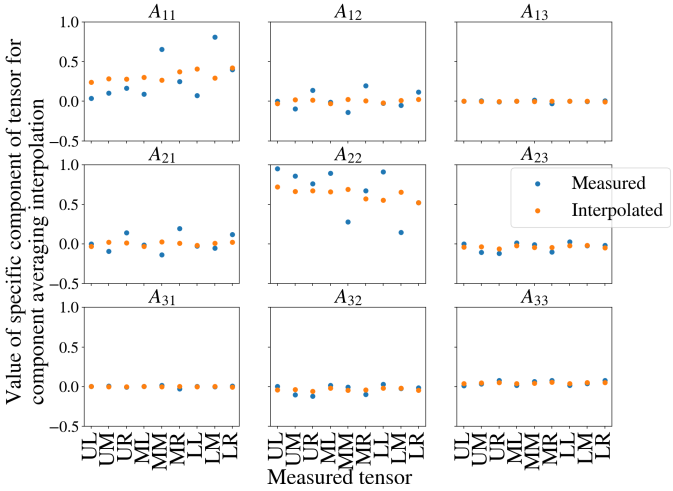


Figure 4.23: Comparison of each of the nine components of each of the nine measured tensors with the corresponding resulting component of the interpolated tensors for the component averaging method. Extracted from Blarr et al. [17].

4.6.2 Decomposition [5]

In Figure 4.24, the results of the interpolation with the quaternion-based decomposition approach can be seen.

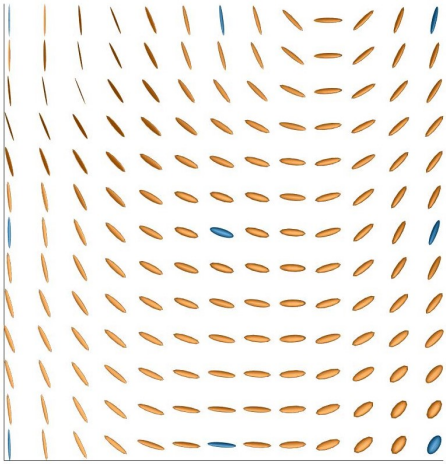


Figure 4.24: Visualization of interpolated (orange) and measured (blue) tensors when using the decomposition-based interpolation method described in this thesis. Extracted from Blarr et al. [17].

The before-mentioned progress of orientations can be visually traced as a clear curve. As for the interpolation method as such, the visual results are for the most part very appealing. Interpolation between the individual measured FOT is good and the transition between two adjacent tensors also appears reasonable. The anisotropy is not basically lost between two differently oriented tensors by "rounding the tensor". The rotation of two adjacent tensors occurs with small angles and therefore smoothly. The only exception to this can be seen at the upper right edge: The interpolated tensor $\boldsymbol{T}_{10,13}$ in the middle of $\boldsymbol{UM}$ and $\boldsymbol{UR}$ behaves somewhat strangely as far as the behavior of the row is concerned.

[5] This section is extracted from the author's publication [17] with only slight linguistic changes.

Instead of closing the estimated angle of 20° between $\boldsymbol{UM}$ and $\boldsymbol{UR}$ by a piece wise change of 10°, the tensor $\boldsymbol{T}_{10,13}$ is oriented in an angle deviating by around 80° compared to the measured ones. However, the tensor $\boldsymbol{MM}$ is for example also taken into account for the calculation of this tensor (just like all the other measured ones of the set $\mathcal{T}_m$), even if weighted less strongly than $\boldsymbol{UM}$ and $\boldsymbol{UR}$, which favors the big rotation of the tensors in the uppermost row considering the global orientation behavior. Furthermore, the behavior in this column looks much better than could be expected if the tensor had rotated in mathematically negative (clockwise) direction around the z-axis than the chosen positive (anti-clockwise) direction. When leaving measured tensors out of the "ground truth" and interpolating them instead, there are definite changes in the orientation course, which can be seen in Figure 4.25.

For example, the behavior of the afore-mentioned $\boldsymbol{T}_{10,13}$ changes significantly when $\boldsymbol{UM}$ or $\boldsymbol{UR}$ are omitted. In general, however, it must be stated that the orientation courses react very agilely and sensibly to the changes when individual tensors are omitted when using the decomposition method.

The quantitative evaluation based on the Frobenius norm is visualized in Figure 4.26. The rather poorer interpolation at the left and upper edges and the relatively good performance in the middle of the plate (and lower right) are noticeable.

Considering the component-wise deviations displayed in Figure 4.27, it is striking that they are considerably high in this specific case for this coordinate system. However, it is noticeable that major trends between the different tensors are mostly preserved by this method (cf., e.g., A_{11}).

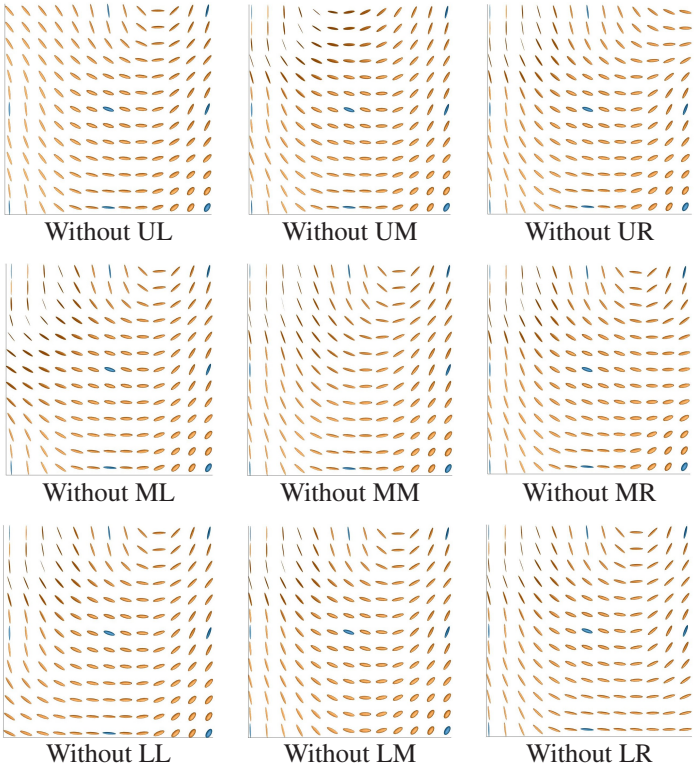


Figure 4.25: The graphic shows the measured (blue) and interpolated (orange) fiber orientation tensor glyphs when leaving one measured tensor out of the calculation and interpolating it instead with the decomposition method respectively. Extracted from Blarr et al. [17].

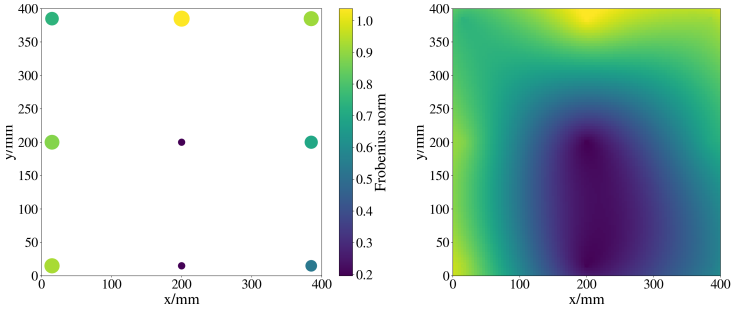


Figure 4.26: Visualization of the discontinuous (left image) and continuous (right image) error of the decomposition method across the plate of the interpolated tensor in comparison to the measured one when leaving this specific tensor out of the computation. Value determined via Frobenius norm. Extracted from Blarr et al. [17].

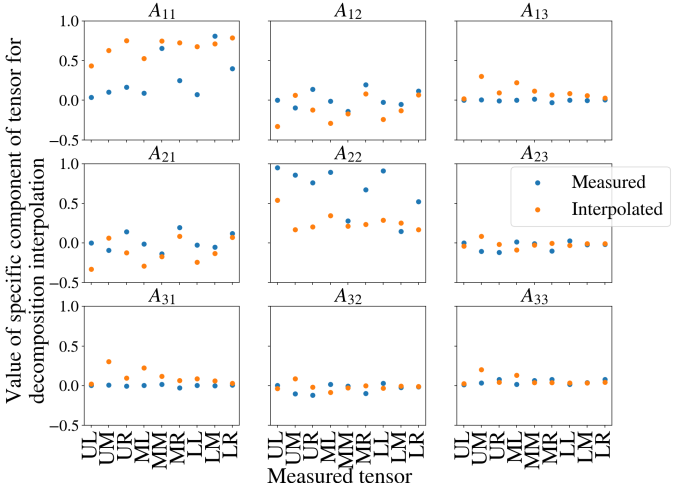


Figure 4.27: Comparison of each of the nine components of each of the nine measured tensors with the corresponding resulting component of the interpolated tensors for the decomposition method. Extracted from Blarr et al. [17].

4.6.3 Artificial neural network [6]

The results of the tensor field when interpolating with the neural network can be seen in Figure 4.28.

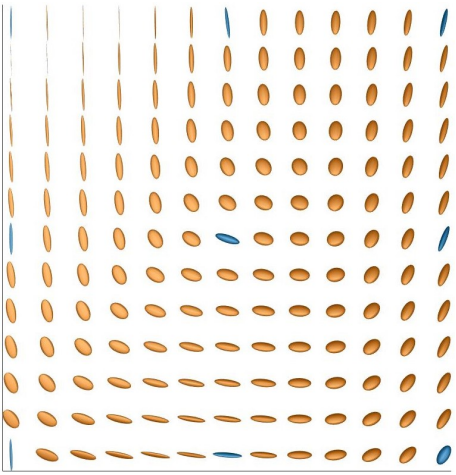


Figure 4.28: Visualization of interpolated (orange) and measured (blue) tensors when using the ANN-based interpolation method described in this paper. Extracted from Blarr et al. [17].

It strikes that when training this network with the measured tensors, it is able to produce both very anisotropic and very isotropic tensors at the points with missing tensors, compared to the other two main methods that rather dispensed one or the other. While some areas look smooth like, e.g., the upper and right area, there is non-monotonous interpolation behavior concerning, e.g., $\boldsymbol{MM}$ and $\boldsymbol{LL}$ where the surrounding tensors are much more isotropic than the measured one and also quite differently oriented (looking specifically at the tensor $\boldsymbol{LL}$). When training the network multiple times with the same input data, the results look very much alike, speaking for the robustness of the method.

[6] This section is extracted from the author's publication [17] with only slight linguistic changes.

However, looking at the plots of the tensor fields when leaving measured tensors out of the input data (Figure 4.29), some of the non-monotonous behavior shows again (cf., e.g., the fields without MM and without MR).

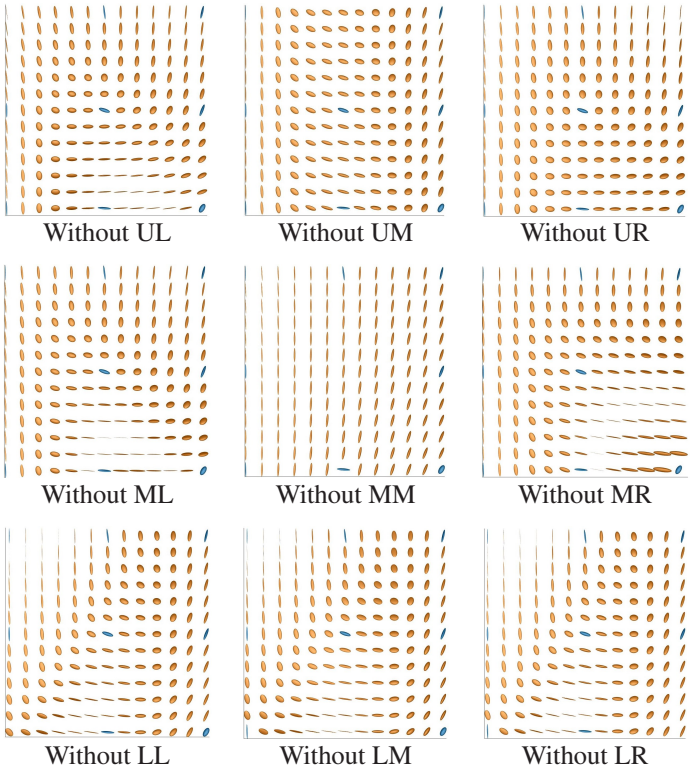


Figure 4.29: The graphic shows the measured (blue) and interpolated (orange) fiber orientation tensor glyphs when leaving one measured tensor out of the calculation and interpolating it instead with the neural network respectively. Extracted from Blarr et al. [17].

Consequently, the difference of the Frobenius norm for these two cases is significantly large and so is the error for the interpolation of the UM tensor, which is all depicted in Figure 4.30.

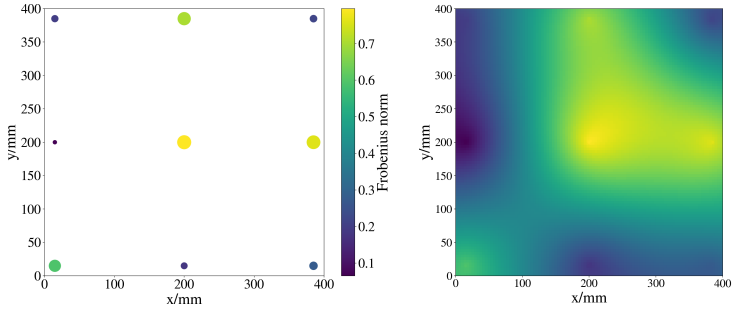


Figure 4.30: Visualization of the discontinuous (left image) and continuous (right image) error of the ANN method across the plate of the interpolated tensor in comparison to the measured one when leaving this specific tensor out of the computation. Value determined via Frobenius norm. Extracted from Blarr et al. [17].

Considering the component-wise errors in Figure 4.31, the results are still comparatively good. The largest deviations occur especially for the three mentioned tensors before. Just as was the case for the other two methods, the biggest deviations appear mostly for the A_{11} and A_{22} components. The comparison between the three methods regarding the differences between the measured and interpolated tensor components will be taken up again in the Discussion (cf. Section 5.4.4).

The quality of the neural network must also be considered with respect to the progression of an error measure over the number of epochs. As an error measure, the already mentioned MAE was used and the course over the epochs can be seen in Figure 4.32. As expected with the small number of training data, the network does not behave optimally. However, the graphs show both overfitting (A_{23}, A_{33}, A_{31}) and underfitting (A_{11}) trends. Thus, it is difficult to draw a general conclusion. Overfitting rather argues for using more training data or stopping at a lower number of epochs, apart from solutions that require more specific methods which are very dependent on the model. Underfitting can be combated by different measures depending on the cause of the underfitting; either more epochs (unlikely here) or more parameters of the model can help,

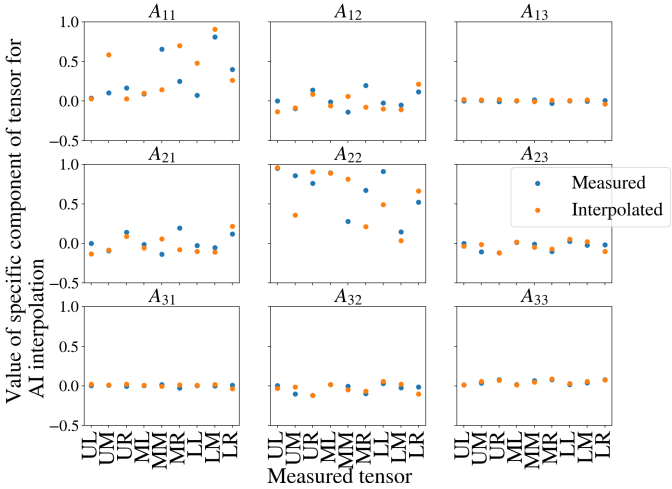


Figure 4.31: Comparison of each of the nine components of each of the nine measured tensors with the corresponding resulting component of the interpolated tensors for the ANN method. Extracted from Blarr et al. [17].

or a change to a completely different model. However, more training data can also help with underfitting, which is most likely in the case considered.

4.6.4 Interpolation of tensors of fourth-order through adapted decomposition method

With the simplified decomposition method for tensors of fourth-order explained in Section 3.7.5, analogous fields of fiber orientation tensors could be created. They are plotted in Figure 4.33 on the left side for the CF-PA6 Plate 3 (cf. Figure 3.4) (a), the GF-PA6 Plate 4 (cf. Figure 3.4) (b), the CF evaluation of the hybrid plate (c) and the GF evaluation of the hybrid plate (d). The different shape of the tensors of fourth-order compared to the ellipsoids of second-order can be clearly seen. Furthermore, it strikes that the measured tensors are deviating a lot less in their shape and orientation among themselves compared to the heterogeneous

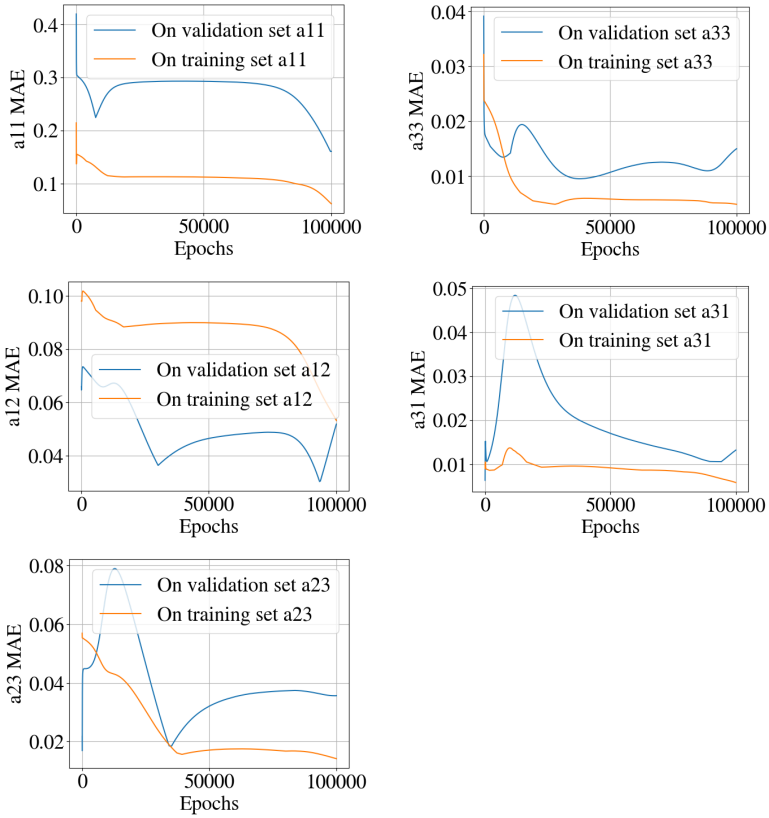


Figure 4.32: MAE over training epochs of the ANN per tensor component. Extracted from Blarr et al. [17].

initial state of the example in the sections before. This is due to the fact that the samples in the previous case were significantly smaller (10 mm compared to 25 mm side length) and thus the orientation was evaluated very locally. This made interpolation significantly more difficult and led to the critical points mentioned above. In these cases, however, the interpolation appears uniform and monotonic, as the output tensors are significantly more similar. The slight upward drift can also be seen in the four cases, although it varies slightly in

strength. In addition, there is partly also a slight, less pronounced downward drift at the right end of the plate in the lower part.

Averaging all 169 tensors into one averaged tensor gives a representation of the overall fiber behavior. These averaged tensors can be seen in Figure 4.33 in column 2. In general, they are similar in shape, with the averaged tensors representing carbon fibers being slightly more isotropic (rounder), which corresponds to observations in the CT scans. The light and thin carbon fibers behave more randomly, deviate in their angles a lot and are less strict aligned with the overall flow. Orientation-wise, all show the slight upwards drift - again varying in intensity.

For the third column in Figure 4.33, the orientation vectors detected in the nine scans were binned into 20 angle categories, each containing an interval of length $\pi/20$. The results were averaged and plotted as fiber orientation histogram. A shift towards slightly positive angle values can also be seen here. The distributions of the carbon fibers are also somewhat flatter (especially in the hybrid case), or show higher occurrences at larger angle values, which matches the somewhat more isotropic tensor glyphs.

The averaged tensor glyphs of fourth-order were subsequently used (without prior closing approximation) as input in the Mori-Tanaka homogenization, and the orientation histograms in the shear-lag modified Halpin-Tsai homogenization. When comparing the resulting stiffnesses with experimentally measured stiffnesses, the use of the 169 tensors as basis for the average resulted in a stiffness body closer to the experimental one than just using the nine measured ones. This will be discussed in detail in Chapter 5.

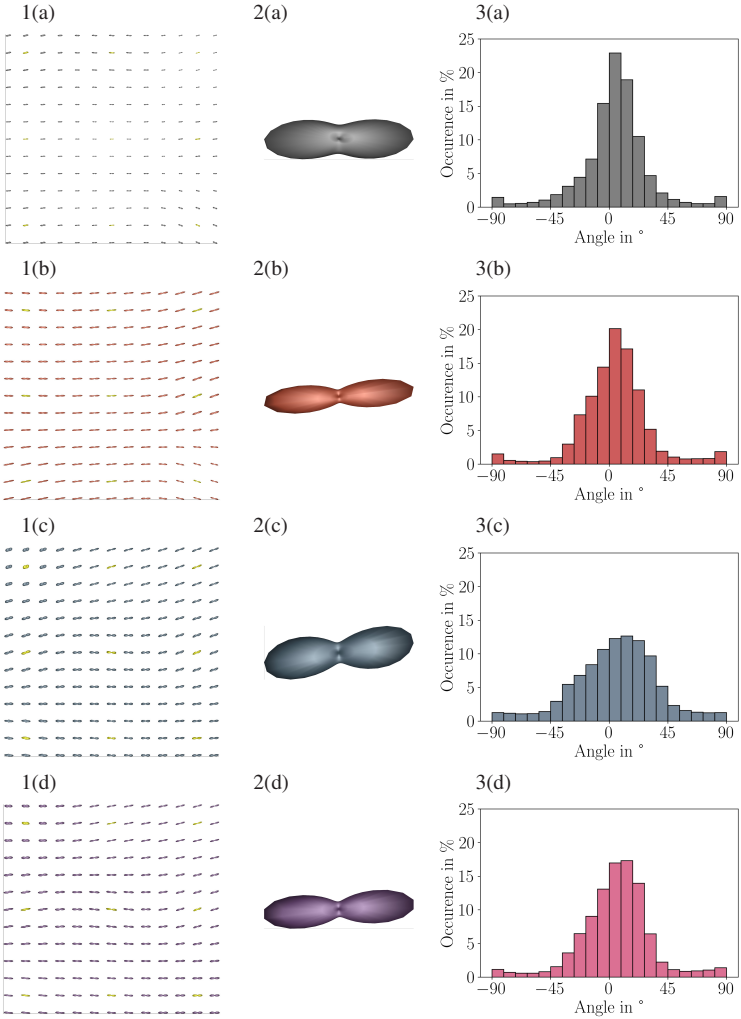


Figure 4.33: Field of tensors of fourth-order (1), averaged tensor glyph of all 169 tensors of fourth-order (2) and histogram of orientation distribution of the mean of all specimen in degrees for 20 groups (width of one column: $180°/20$), for the CF plate (a), the GF plate (b), the carbon fibers in the hybrid plate (c) and the glass fibers in the hybrid plate (d). Extracted from the publication [274], co-authored by the author.

4.7 Microstructure image generation through generative adversarial network

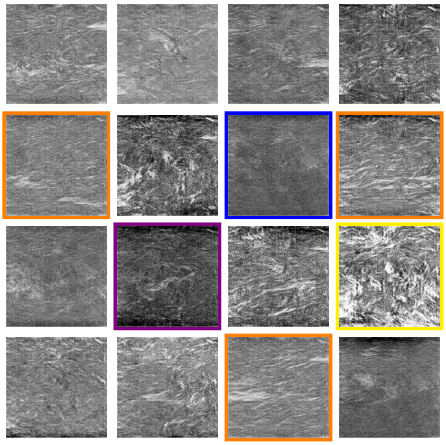
In the following, the results of the final DCGAN network are presented - from the actually generated images to quality evaluations of the network in the form of the loss plot to quality evaluations of the generated images in the form of FID calculations and nearest neighbor considerations with the two different metrics of ED and SSIM. The results of the following sections have all already been published in a paper [275].

4.7.1 Generated images [7]

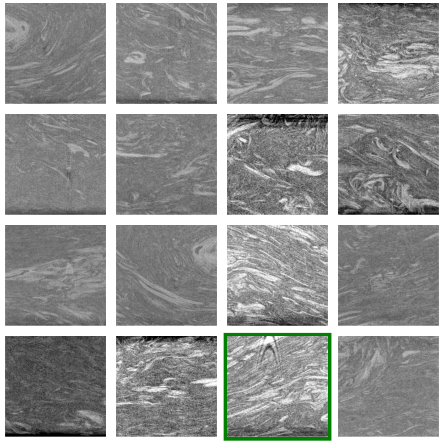
The DCGAN was trained for 75 epochs on 29,280 input images (cf. Section 3.8.2) with 128 images per batch corresponding to 228 iterations per epoch when dropping the last non-full batch (cf. Section 3.8.5). A comparison of randomly chosen real and generated images is shown in Figure 4.34.

It can be observed that every generated microstructure is different. Generated images look mostly realistic (cf. orange-rimmed images in Figure 4.34) including fibers and fiber bundles. The overall fiber orientation resulting from the flow in the compression molding process (cf. Section 4.5) is visible and the contrast and brightness varies between the images just as in the training images. There are, however, few images which show characteristics that are not represented in the input data set such as two dark border regions on opposite sides of the image (cf. violet-rimmed image in Figure 4.34). This phenomenon has to be a result of the combination of features from different images as in the original ones, at most one side displays this artifact related to the scan process. In general, the observed occurrence of dark border regions in the generated images can, for the most part, not be traced back to the DCGAN structure but is subject to the quality of the training data set. On the other hand, it should be

[7] This section is extracted from the author's publication [275] with only slight linguistic changes.



(a) Randomly picked generated images



(b) Random images from training data set

Figure 4.34: Comparison of (a) randomly picked generated images (after 75 training epochs) side by side with (b) a random selection of real images of the training data set. Selected images are highlighted in color as they are especially realistic looking (orange), show little to no fibers (blue) or an excessive amount of fibers and fiber bundles (yellow). Furthermore, some real and generated images contain artifacts such as dark image borders (violet) or striped patterns (green). Extracted from Blarr et al. [275].

positively emphasized that striped patterns (cf. green-rimmed image in Figure 4.34), which occur in the training data and represent image artifacts as well, were no longer found in the generated images. As they were not as common as the dark regions on the edge, these artifacts disappeared in the course of the training process. Furthermore, some images might show little to no fibers (cf. blue-rimmed images in Figure 4.34) or an excessive amount of fibers and fiber bundles in unrealistic orientations, hence strongly deviating from the general flow direction or showing extreme curvature (cf. yellow-rimmed image in Figure 4.34).

4.7.2 Loss plot and FID [8]

The loss value of the generator and discriminator network can be analyzed to get information on the stability of the training process. This plot is shown in Figure 4.35. It can be seen that both losses approach each other after only a few epochs. However, while the loss of the discriminator stays on a very small value for the rest of the training process, the generator loss increases slightly and oscillations grow bigger. The smallest possible loss values for both generator and discriminator would be zero, but they cannot simultaneously reach this value. Hence, ideally, both losses should converge to approximately the same value resulting in a balance of generator and discriminator or decrease monotonously on average towards different values indicating a stable training process. This is based on game theory and the aim to reach the Nash equilibrium. Even though the generator loss in Figure 4.35 does not decrease monotonously (not taking oscillations into account), the DCGAN in this paper still resulted in a stable training process and the generation of meaningful images. In fact, similar shapes of loss plots have been observed for other GANs as well [323–326]. The behavior of the loss plot and its influence on the quality of the generated images and the need for a different metric to assess them is elaborated on in the Discussion (Section 5.5).

[8] This section is extracted from the author's publication with only slight linguistic changes [275].

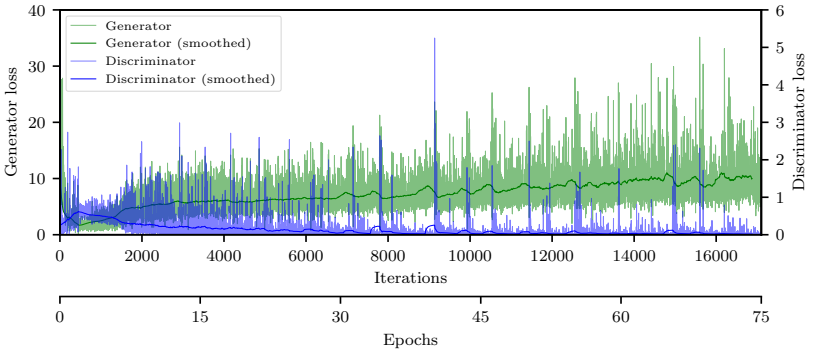


Figure 4.35: Plot of the generator and discriminator loss of the final network. The smoothed curves of the loss values are calculated as floating average over 228 iterations (corresponds to one epoch). While the discriminator loss increases merely visible at the very start, it decreases to then stay constantly at very low values indicating correct assessment of the images. The generator loss values increase contrarily after a small valley at the beginning but stay roughly at a constant loss range although oscillating heavily. Extracted from Blarr et al. [275].

In order to quantitatively judge the generated images even further, the associated FID value was calculated after every epoch and is depicted in Figure 4.36. It drops sharply during the first few epochs and then remains at this level without major fluctuations. The average of the last 45 epochs is approximately 150.6 with a standard deviation of about 5.7.

This coincides with the visual perception much better than the loss plot: Displaying one randomly chosen generated image based on a fixed noise vector after each epoch (cf. Figure 4.37 and Figure 4.38) shows that the images become slightly better over time and not worse, which will be discussed further in Section 5.5.

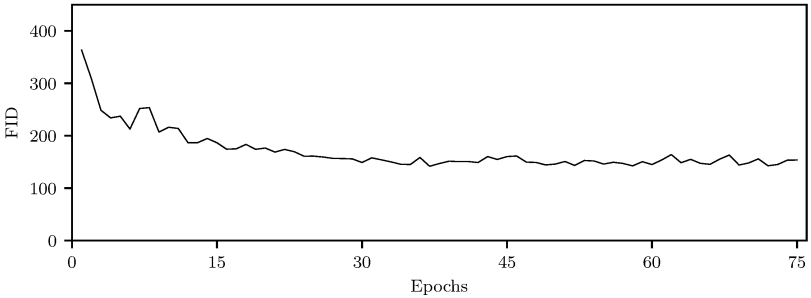
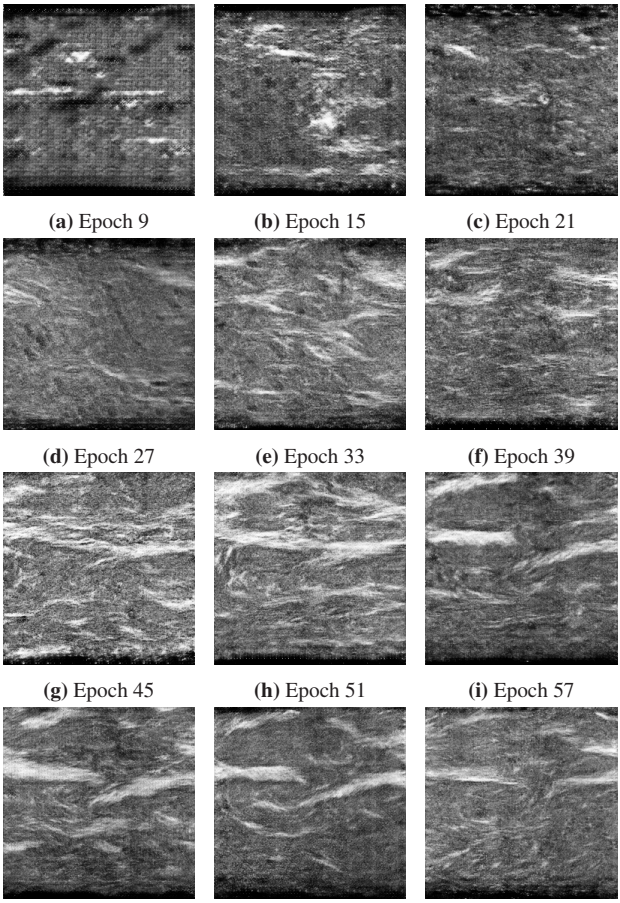


Figure 4.36: Plot of the FID of images generated by the final network. The values decrease up until about 30 epochs where the distance between the distributions of real and generated images stays more or less constant. The FID hence corresponds more to visual perception than the loss plot, at least in its initial course. Extracted from Blarr et al. [275].

4.7.3 Nearest neighbors [9]

The distance to the nearest neighbor for every image within the last batch of generated images (after 75 epochs) is depicted in Figure 4.39(a). The values are sorted in ascending order (from left to right). Both for ED and structural similarity index measure (SSIM), there is a wide distribution of images with a relatively similar distance to their nearest neighbor and no noticeable sharp steps in the plot. The range of possible SSIM values is (-1, 1], meaning that only about 2 % is covered in the plot. On the contrary, the ED can output values between 0 and 512 for normalized image tensors of size 256×256 px and entries in the range [-1,1] as used in this work (cf. Section 3.8.4.2, Equation (3.19)). Therefore, the calculated ED values cover a significantly larger proportion of the possible range (more than 16 %). However, these values cannot be compared directly since the distribution of distance values is not linear and differs for ED and SSIM. It can be noticed that for the ED the curve rises sharply on the right edge which leads to the assumption that some generated images show a less strong resemblance to even the closest image from the input data set.

[9] This section is extracted from the author's publication [275] with only slight linguistic changes.



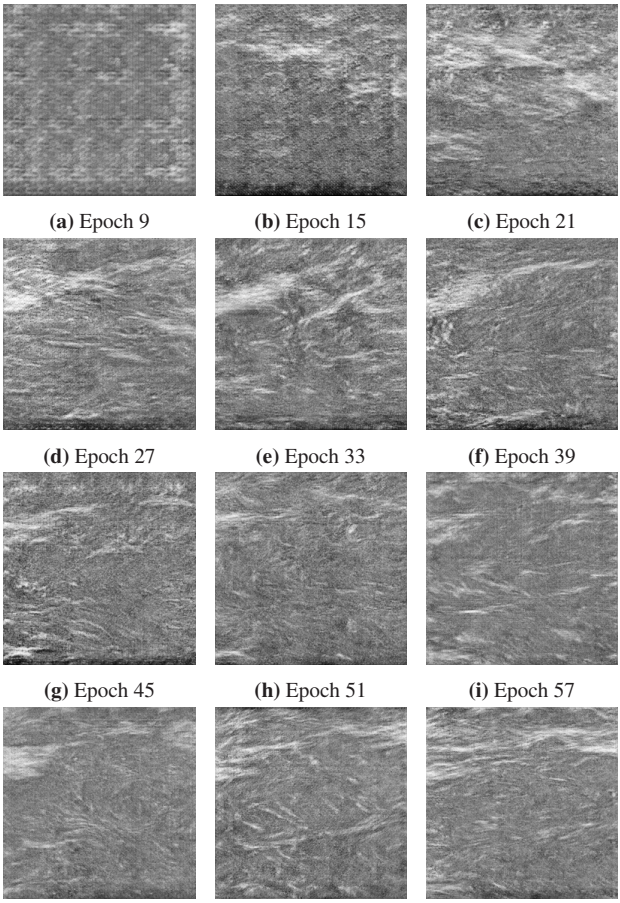
(a) Epoch 9 **(b)** Epoch 15 **(c)** Epoch 21

(d) Epoch 27 **(e)** Epoch 33 **(f)** Epoch 39

(g) Epoch 45 **(h)** Epoch 51 **(i)** Epoch 57

(j) Epoch 63 **(k)** Epoch 69 **(l)** Epoch 75

Figure 4.37: Evolution of a selected representative image. From epoch to epoch, new microstructure characteristics emerge and existing features are refined. Extracted from Blarr et al. [275].



(a) Epoch 9 **(b)** Epoch 15 **(c)** Epoch 21

(d) Epoch 27 **(e)** Epoch 33 **(f)** Epoch 39

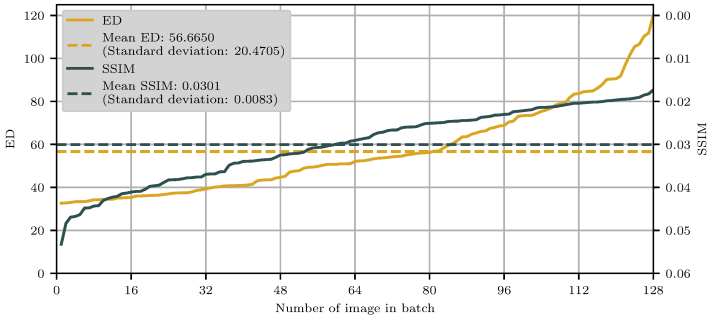
(g) Epoch 45 **(h)** Epoch 51 **(i)** Epoch 57

(j) Epoch 63 **(k)** Epoch 69 **(l)** Epoch 75

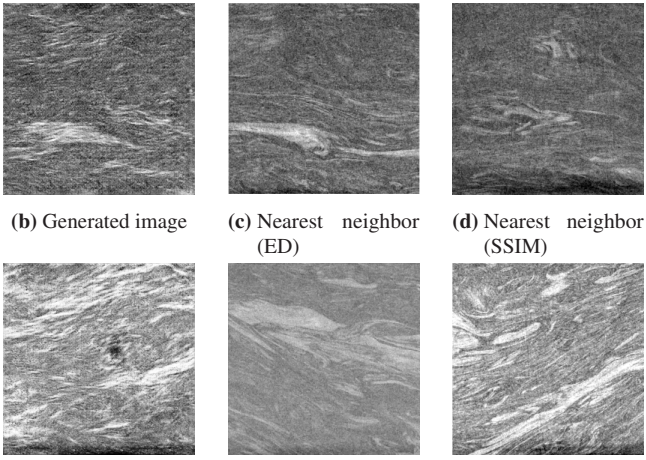
Figure 4.38: Evolution of a selected representative image. From epoch to epoch, new microstructure characteristics emerge and existing features are refined. Extracted from Blarr et al. [275].

On the other hand, the SSIM curve drops downwards (i. e. to higher values) at the left indicating that there are also few images that are very close to their nearest neighbor from the input data set based on the SSIM. For both ED and SSIM it is unclear, what kind of shape of the curve is desired. Both images of high quality (that means realistic looking images that are no copies of the input data set) and images with a close nearest neighbor as a consequence of copied image sections could score an equally low value. Furthermore, a consistently high quality throughout the entire test batch does not necessarily go hand in hand with an even distribution (i. e. a horizontal line). This is due to the differences in images from the input data set. A side by side comparison of two exemplary generated images Figure 4.39(b) and Figure 4.39(e) and their corresponding nearest neighbors for both ED and SSIM distance measure is shown in Figure 4.39. For example image (b), both methods find decent nearest neighbors. The nearest neighbor determined by ED even seems somewhat closer. However, in the case of image (e), the nearest neighbor that is found through the SSIM measurement fits much better. It appears that for images with clearly recognizable and circumscribed fiber bundles that also appear at the same place as in a training image, ED is a suitable measure. As soon as fibers are rearranged in an angle or shifted in respect to the input images or the amount of fibers in the entire image changes leading to large non-aligning areas, the SSIM was found to be the more robust measure.

In order to not only judge the closeness of a final generated image to the training data set, but also the evolution of their proximity throughout training, the smallest ED and highest SSIM value of every epoch for one fixed generated image was plotted. These results can be seen in Figure 4.40 and Figure 4.41 for the first image and Figure 4.42 and Figure 4.43 for the second one, corresponding to the respective image series in Figure 4.37 and Figure 4.38. In both cases the ED and SSIM curves decline at the beginning and either increase slightly in the end (ED and SSIM for the first image series, Figure 4.40 and Figure 4.41) or stay approximately constant (ED and SSIM for the second image series, Figure 4.42 and Figure 4.43). It is striking that low values, hence high proximity to the nearest neighbor in the training data set, appear for images that show small



(a) Distance between generated images from the last batch (after 75 epochs) and their nearest neighbor in the training data set based on ED or SSIM. The values are sorted in ascending order (left to right). Note that for the ED, a small value corresponds to high similarity whereas for the SSIM, with values ranging between -1 and 1, a value of 1 would mean perfect similarity (i. e. identical images).



(b) Generated image

(c) Nearest neighbor (ED)

(d) Nearest neighbor (SSIM)

(e) Generated image

(f) Nearest neighbor (ED)

(g) Nearest neighbor (SSIM)

Figure 4.39: Top: Plot of the ED and SSIM of the generated images from the last batch after 75 epochs and their respective nearest neighbor in the training data set. Bottom: Examples of two generated images ((b) and (e)) and their respective nearest neighbor image of the training data set, based on ED ((c) respectively (f)) or SSIM ((d) respectively (g)). Extracted from Blarr et al. [275].

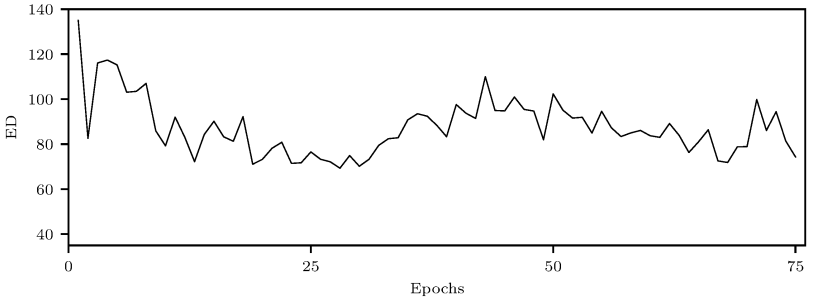


Figure 4.40: Plot of the smallest ED value between one chosen generated image (cf. Figure 4.37) and the training data set over the duration of training. Extracted from Blarr et al. [275].

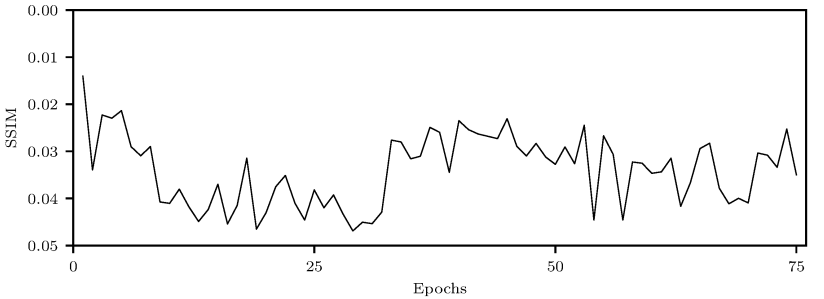


Figure 4.41: Plot of the highest SSIM value between one chosen generated image (cf. Figure 4.37) and the training data set over the duration of training (higher value corresponds to higher similarity). Extracted from Blarr et al. [275].

amounts of fibers and therefore small gray value fluctuations and a somewhat smooth and evenly distributed structure. Images of this kind appear in the training data set but more often images with widely distributed fibers and fiber bundles occur. Hence, generated images with characteristics close to the latter description should lead to at least equally close nearest neighbors.

Figure 4.42: Plot of the smallest ED value between one chosen generated image (cf. Figure 4.38) and the training data set over the duration of training. Extracted from Blarr et al. [275].

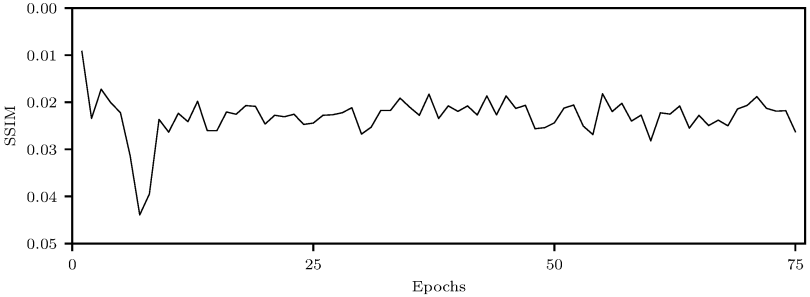


Figure 4.43: Plot of the highest SSIM value between one chosen generated image (cf. Figure 4.38) and the training data set over the duration of training (higher value corresponds to higher similarity). Extracted from Blarr et al. [275].

5 Discussion

In the following, the results of the plastificate investigations are discussed as well as the fiber volume contents, the methods for their evaluation, and the fiber lengths. The other sections deal with the results of the fiber orientation and the fiber orientation tensor interpolation as well as the microstructure generation by the GAN.

5.1 Investigations of plastificate

As a preliminary point, it should be noted that the plastificate investigations were more of a phenomenological investigation than a statistically verified one. In many cases, visual observations (material orientation around screw vortices, expansion of the plastificate) or results from another type of examination by colleagues in the International Research Training Group (IRTG) (volume detection via GOM Atos 3D scanner, weighting, density calculation) should be confirmed or new procedures were to be motivated. However, the plastificate investigations were included in this dissertation because the majority of research is still concerned with the characterization of the finished part after compression molding and not the plastificate.

The fairly evenly rounded structure around the screw vortices was expected and was already visually detectable from the scans. This double helix structure in the cross section was, e.g., described by McLeod et al. [322]. Furthermore, Radtke claims that there is a pre-orientation of the fibers in the plastificate strand as a result of the discharge from the twin-screw device as well in his dissertation

[321]. He confirms the observation in this work that in the outer layers, the fibers are arranged parallel to the extrusion direction, as they are aligned by the polymer adhering to the extruder walls and the discharge nozzle wall. Inside of the strand, the alignment of the fibers are predominantly influenced by the rotation of the screw and the feed of the material according to Radtke, which also coincides with the author's findings.

In contrast, the relatively constant alignment of the fibers perpendicular to the extrusion direction in the plastificate was more surprising. Radtke also determined pre-orientations in 2009 using computed tomography of a plastificate strand [321]. He examined five layers separately across the thickness. Even though the work unfortunately does not show a planar evaluation on the extrusion plane (x-y plane), which makes a direct comparison with the findings in this work more complicated, a plot with the frequency of occurrence of certain angles also shows a clear tendency towards 0°, but then drops quite linearly towards 90° in both positive and negative direction, so that angles not perpendicular to the extrusion direction also occur frequently. This behavior appears quite regularly regardless of the layer examined across the thickness. [321] This confirms findings of Tröster's earlier dissertation [320]. He also discovered the symmetry of the fiber orientation across the thickness of the plastificate and the surface layer oriented in extrusion direction observed in this work and by Radtke. In addition, he confirms the finding of the orientation being similar along the length of the plastificate, which was also shown in this work. In particular, he found the same course of angles as Radtke though, with its mirror-symmetric orientation regarding the length axis of the plastificate. Specifically, he shows a similar broad and flat frequency distribution with a low maximum value indicating a small degree of preferred orientation in extrusion direction. In fact, he schematically depicts the fibers being oriented in -60° (left of the middle length axis of the plastificate) and +60° (right of the middle length axis of the plastificate) to the extrusion direction. [320] This result of Tröster and Radtke is the only one that is clearly not congruent to the findings in this work. However, both Radtke and Tröster examined polypropylene with glass fibers. The one polycarbonate scan in Figure 4.3 alone shows how strongly

the matrix system influences the plastificate. Furthermore, the fiber volume content and the speed of extrusion certainly influence the orientation as well and multiple of these parameter combinations would need to be investigated in order to make a definite statement on this. Anyway, the evaluated orientation behavior of the polyamide plastificates in this work is consistent with the visual impression of the scans and physical plastificate sections.

Pre-orientation in the plastificate is crucial for correctly predicting the fiber orientation of the final part in process simulations. This non-uniform, local fiber orientation state, which is influenced by the extruder geometry, has so far typically either been neglected by assuming an isotropic or planar-isotropic orientation state or has been measured through image-processing of scans similar as in this work, which is then mapped to the finite element model of the charge used in the simulation [5]. The regularity of both the vortex orientation and the orientation in the extrusion direction in turn allowed considerations to generate these initial fiber orientation states in a different way instead of determining them each time in a time-consuming manner by CT scans. Schreyer therefore developed a novel approach to determine the initial fiber orientation state with the help of the tensors determined by the author and based on geometric assumptions [276]. Depending on the extruder type and the dimensions of the plastificate, the tool can generate a three-dimensional data set that outputs the mesh information together with the tensorial representation of the initial fiber orientation state. With the help of this simplified generation, the influence of different geometric variations of the plastificate, different extruder variations and different insertion positions of the plastificate in the press on the final product of the compression molding process can be estimated without any experimental effort. An example of such a generation (right) and the corresponding original state (left) of a CF-PA6 plastificate is shown in Figure 5.1. [276]

As far as the porosity and the expansion of the plastificates are concerned, observations and results of alternative methods were able to confirm the results of this work. When systematically measuring the volume via GOM Atos 3D scanner, the weight and calculating the consequential density of the old and

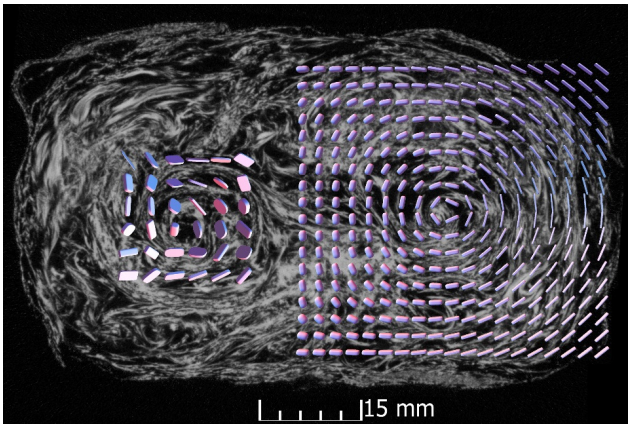


Figure 5.1: Section of a scan of a CF-PA6 plastificate with their measured orientation tensors plotted at the left swirl and the geometrically generated ones at the right swirl - superimposed as tensor glyphs. Extracted from the publication [276], co-authored by the author.

new ends of various plastificates, colleague Christoph Schelleis was able to determine a higher porosity (and thus a lower density) of the front end in each case. Using the identical method, the same correlation between higher porosity and higher screw speed was also demonstrated by Sven Löwe in his Bachelor's thesis [327]. He consistently showed that higher rotation speed led to smaller densities and therefore higher porosities both for high and low fiber volume content. He furthermore showed that higher fiber volume contents lead to lower densities and higher porosities compared to plastificates with lower fiber volume content. [327]

5.2 Determination of fiber volume content

The precise determination of fiber volume content is of particular interest for discontinuous fiber reinforced polymers. This allows both checking the set process parameters to achieve the desired fiber volume content and providing exact values for modeling.

Motivation for the development of recognition from CT images instead of the exclusive use of experimental methods was already given in the State of the Art (Section 2.2.1.3). In the following, the results of all methods used are evaluated and possible improvements are suggested.

5.2.1 Chemical method

As for the method itself, chemical acid-based dissolution of the matrix is the standard for determining the fiber volume content of CFRP, but it is more complex and labor-intensive than TGA [328, 329]. There are no comparative values and no better or similarly good methods - therefore the quality of the results cannot be checked. Nevertheless, the measurements are used as ground truth because, according to current knowledge, there is no superior method.

As far as the FVC results are concerned, initially the low values at the borders of the specimens and the higher values towards the middle of the thickness are striking. The so-called "shell-core effect" known from injection molding signifies changing fiber volume content and fiber orientation between shell and core layer of the plate [5, 17]. While the fiber orientation effect is only visible in the area of the initial charge (and will still be discussed in Section 5.4) and non-existent in the flow area, the effect of the compression molding process on the through-thickness FVC is clearly visible.

In general, the measured mean values (24.2 % for Plate 2 and 23.6 % for Plate 3) deviate from the target value of 25 %. It is difficult to answer why the true values are slightly below the values actually set via the process parameters. This is likely to be a measurement error, although the cause may be of various kinds. The calculation made for the process may not have been accurate enough or the feeding speed of the fibers may have deviated slightly from the calculated one. Fiber losses in the extruder are also conceivable.

Furthermore, the clear increase in values when using larger samples is surprising. Although the six samples of course do not provide sufficient statistical certainty,

the effect seems clearly pronounced. While an increase in mean fiber length with increasing specimen sizes due to the cutting off of longer fibers in small specimens makes sense, there is no obvious reason why the pure fiber content should be lower in small specimens. A literature search on this did not yield any results; while mechanical tests are often performed with different sample sizes, most authors determine FVC with only one sample size. Two possible explanations are discussed in the following. Firstly, the significance of the FVC value of a smaller sample is generally lower, as it is statistically more likely to catch a spot in the panel that happens to be particularly rich in fibers or matrix. In the cases measured, however, it would then have to be a very matrix-rich area in each case. Another possible explanation is the fact that fibers can be pulled out of the material at the cut surfaces when the sample is cut and removed from the plate. As the small samples have a higher surface-to-volume ratio, this effect is more significant for them. Now, the fibers could also get stuck in the specimen and be detached from the plate, potentially increasing the FVC. However, the probability that the larger part of the fibers is stuck in the plate is particularly high for small specimens, which in turn could lead to the shorter piece in the specimen being torn out. The probability that the longer part is not in the specimen is lower for larger specimens. The idea is illustrated graphically in Figure 5.2. Furthermore, a similar effect to the problem with incineration is conceivable as a possible cause: Larger specimens have to be exposed to the acid for longer to dissolve the matrix due to the poorer surface-to-volume ratio. However, since the same treatment time was selected for the smaller specimens, the matrix could be removed more quickly and the fibers subsequently attacked, as sulphuric acid is in fact oxidizing and reacts with graphite. This could also lead to the lower measured fiber volume content.

There is also a recognizable difference between the charge and flow areas. The three samples in the charge area of Plate 2, FLD_1 - FLD_3, have on average a higher FVC than the samples in the flow area, FLD_{10} - FLD_{12}, and C_1 - C_3 of Plate 3 also have on average more fibers than F_1 - F_3. However, the central samples of Plate 3, which have an even higher FVC than the charge samples, do not quite fit the picture. Since the differences in the mean values are on average

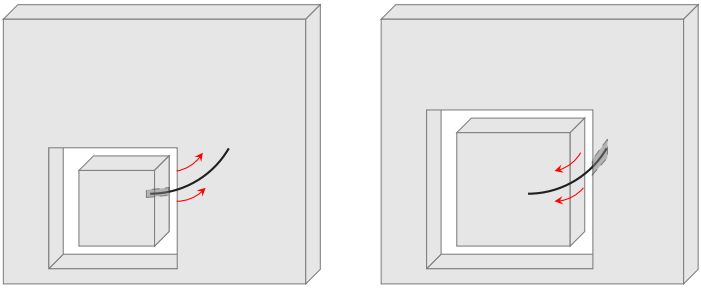
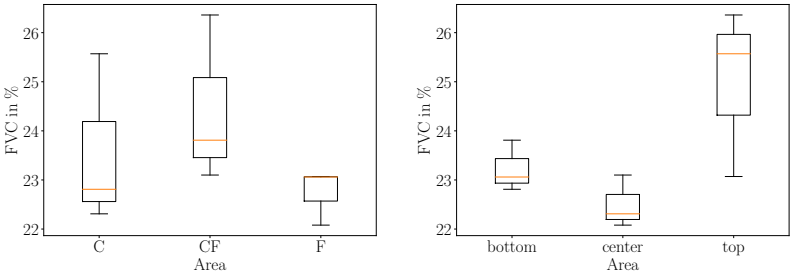


Figure 5.2: Schematic representation of the theory for the reduction of the fiber volume content in small specimens. Left: The fiber is less embedded in the small specimen than in the plate and is therefore torn out of the specimen. Right: The fiber is more embedded in the large specimen than in the plate and is therefore torn out of the plate.

less than 1 % between the different areas of Plate 3, the possibility that these differences are due to measurement errors or general plate fluctuations must also be taken into account. Overall, the number of samples is too small to be able to make valid statements. What is also noticeable is that the values in the center of all three columns, i.e. the values of samples C_2, CF_2 and F_2, are the smallest. Considering the flow of the plastificate depicted in Figure 5.5, it makes sense that most fibers are pushed towards primarily the upper and partly also the lower area of the plate. The results of Plate 3 are summarized comprehensibly as boxplots in Figure 5.3.

In his dissertation, Radtke carried out similar measurements of the fiber volume content in the different plate areas [321]. He found that greater fluctuations were observed in the results across the width of the plate than along the flow path. This observation is consistent with our results: The standard deviation within areas C and CF is large (see Figure 5.3). However, the fluctuations in the flow area over the height of the plate are small. In contrast, the FVC values along the flow path in the area of the 3-sample (bottom) and the 2-sample (center) are quite constant. Only along the 1-samples (top) is the deviation similar to that within the charge and transition region of the plate. The upper samples (top, 1-samples) generally show a high FVC. However, this corresponds very well with the skewed flow front discovered, which will be explained in the context of



(a) (b)

Figure 5.3: Boxplots for the fiber volume content results of Plate 3 categorized into charge, transition area and flow area (a) and top (1), center (2) and bottom (3) specimens (b). Data first published by Scheuring et al. [274].

the determined fiber orientations, according to which a slightly upward-pointing fiber orientation and thus flow front could also lead to an increased fiber volume content at the upper end of the plate. The flow area shows the smallest average FVC along the flow front; presumably not too many fibers arrive there.

5.2.2 "Average or above" (AOA) thresholding [1]

With regard to the two computational methods for determining fiber volume contents, the novel thresholding method AOA will be discussed first. It shall be noted here directly that this method was born out of necessity, so to speak, as conventional thresholding methods did not work, as can be seen from the Appendix, and alternative methods such as Gandhi's [5] were not possible.

The average of the FVC values determined by chemical dissolution was 24.2 % and the AOA method determined an average value of 26.5 %. The absolute mean deviation of 2.7 % is therefore greater than that of the CNN with about 1.5 % (cf. Table 4.3). The deviation was mainly increased by the two outliers, which

[1] This section is based on the author's publication [188].

were included in the result because there was no explanation as to why those scans should differ (similar histograms to other scans, similar visual perception, etc.). Without these two outliers, however, the average deviation of the AOA thresholding would only be 0.6 %, which is more than twice as good as the CNN. In addition, the results in Figure 4.11 of the AOA appear more reliable and plausible than the results in Figure 4.14 of the CNN. The fact that the average value of the CNN fits well is due to the average of the training data or is a coincidence. Sample by sample, the agreement is much worse than with AOA thresholding. So if the FVC of a new sample had to be determined, AOA thresholding could probably be used with more confidence than the CNN.

If the method is to be evaluated as such, the general validity of the approach must be critically examined. There is no guarantee that it will also work for alternative material systems or scans of a different quality. It is fundamentally based on the fact that the samples have fewer fiber bundles at the border of the sample (in relation to the thickness) and many fiber bundles in the middle of the sample. This changed the histograms per slice, which generally did not show two clear peaks. The meaningful determination of a threshold was therefore difficult with these changing histograms across the thickness. For all materials/scans that do not exhibit this peculiarity, the approach will not be of interest, at least not in exactly the same way. Regarding the adjustment of the threshold determination across the thickness, alternative/more complex methods could be used than the binary decision between the Otsu threshold of a single slice and the average Otsu threshold of the whole stack chosen in this work. Instead, another non-constant adjustment of the threshold could be developed. A mathematical determination of the variation of the histograms over the thickness and a corresponding continuous adjustment of the threshold values would be possible. Theoretically, it would still be possible to locally enhance the CT scans with the low resolution (i.e. single fibers and matrix mixed as mean gray value) with small CT scans of higher resolution in order to be able to better interpret the gray values. However, the aforementioned difficulty that an increase in stochastic noise occurs at higher resolutions makes segmentation even more difficult. In addition, the sample would then have to

be cut into smaller pieces so that the sample fills the entire field of view, as otherwise even more noise would occur. This in turn creates different boundary conditions if the FVC is still to be determined destructively and was no longer possible in the case of this work anyway.

The previous application of the filters had a non-negligible influence on the quality of the results. Nevertheless, it should be emphasized that the main factor is thresholding. When looking at Figure 3.15, for example, it becomes clear that although the median filter was applied to both images, the amount of fibers is still greatly overestimated by the normal Otsu threshold. The correct choice of filter is an additional finetuning factor. Considering that neural networks are essentially just a series of filters, the importance of filter selection should not be underestimated, especially in the case of noisy images with low contrast.

The AOA thresholding method is by no means optimal, but simple global thresholding turned out not to be an option. The literature was also surprisingly sparse, especially for discontinuous fiber reinforced polymers. Apart from a few sources already mentioned in the State of the Art, Yu et al. [55] investigated fiber volume contents, for example, using a method that combines scanning electron microscopy and micro-computed tomography; however, for yarns in textile composites, which in turn has a completely different structure. Wintiba et al.'s work [330] also deals with woven composites. For these reasons, various approaches were tested, including a normalization of the threshold values in relation to the mean grey value/brightness of a scan, but none of them worked as convincingly as the AOA method. It is a quick and simple method with sample-wise even better results compared to the high development and training effort of the CNN. However, it was particularly convincing that the FVC values calculated by the method increased or decreased with the experimental values, apart from the two outliers, so that the trend was mostly correct.

The possibility of easily investigating the course of the FVC over the thickness of the sample, which was already mentioned in the motivation, should of course be emphasized. As can be seen in Figure 4.12, the material examined in this work shows a strong decrease in fiber concentration towards the edges, with a

maximum in the middle of the sample. Gandhi et al. [5] showed similar results for LFT processed by injection molding (see also [331]). They also showed that the core layer width and maximum concentration increases with higher nominal fiber concentrations. This fiber distribution across the thickness is of course particularly important in bending tests, where most of the load falls on the outer areas.

5.2.3 CNN [2]

At first glance, the CNN was convincing with the prediction of a mean FVC of 24.3 % (compared to the experimental value of 24.2 %) and a mean absolute deviation of only 1.5 % (see Table 4.3). However, as already mentioned in the previous section, this is probably due to the average of the training data or is a coincidence. The good agreement of the mean values should not be given great importance if the individual values are usually not correct. It is noticeable that, as with the thresholding approach, some samples are predicted very well, while others deviate by well over one percent. Apart from the two outliers of the AOA thresholdings, however, the CNN predictions are worse than those of the AOA method. So while the maximum error is lower with CNN, the logical relationship (increasing experimental FVC leads to increasing predicted FVC) is found less frequently than with the thresholding approach (as seen in Figure 4.14). Since the CNN uses the experimental data as a training basis, while the AOA does not, and the CNN nevertheless makes relatively weak predictions, this clearly speaks in favor of AOA thresholding.

There are various possible reasons for the loss behavior mentioned in the Results, according to which the validation loss is smaller than the training loss. The first possibility is the influence of so-called regularization methods (such as dropout). This deliberate, random omission of neuronal connections is used in the training process in order to better generalize the training, as explained

[2] This section is based on the author's publication [188].

in the State of the Art. This often sacrifices some of the training accuracy. However, dropout is not used in the validation process, which is also the default setting in Keras. As a result, the validation loss may be better than the training loss. It would be possible to determine this regularization loss by manually applying dropout in the validation phase, for example, in which case the curves would probably look different. Another reason could be the timing of the loss calculation. The training loss is normally reported continuously as an average of the losses of a batch within an epoch. The validation loss, on the other hand, is determined once at the end of an epoch, after the model has already been updated throughout the epoch, allowing it to benefit from the full extent of the epoch's learning. If the validation loss were shifted to the left by half an epoch, which would correspond to the mean report time difference, the graphs would also already look different. The last and probably worst reason would be that the validation set is too simple. This would be conceivable because the training data (original and augmented scans) of a sample did not strictly belong to the training or validation set in this work. Adjusting this would be a necessary improvement.

There are several further ways to improve the CNN. Perhaps the most obvious would be more diverse input training data. Although 448 scans were used for training, these came from only 14 original scans that were augmented. Not using more original scans was due to the fact that experimental FVCs were only available from the 14 samples, as these examinations had to be carried out externally at FIBRE. In addition, the fiber volume contents of these 14 samples were all between 22.08 % and 28.6 %. As a result, the network was highly trained to predict in this range. The only attempt to augment this dataset was to use scans of pure PA6, as it had the guaranteed FVC of 0 %, but this was not successful. It would certainly increase the accuracy of the CNN in the long term if a wider variety of FVCs were included. However, it should be mentioned here that the CNN naturally made better predictions in terms of mean error than simply taking the mean of the FVC values for each sample. The reason for using these samples of very similar fiber volume content was again due to the available data: the plates were manufactured at Fraunhofer ICT and

CF-PA6 plates were only produced with the same mean FVC. The fluctuation of the FVCs is therefore purely due to the fluctuations of the FVC across the plates. These were not small, which in turn demonstrates the necessity of such a local investigation. Another way to increase performance would be to use a higher resolution for the scans. The reduction to $100\,\mathrm{px} \times 100\,\mathrm{px} \times 100\,\mathrm{px}$ was necessary due to the limitation of computational resources. Any higher resolution led to memory issues in the bwHPC cluster, which was used for the calculation. A higher resolution would allow a better inclusion of single fibers or smaller fiber bundles and improve the distinction between fibers and irrelevant gray value deviations of the matrix. However, it should also be mentioned that this resolution is quite competitive when compared to other CNNs [332–334]. Of course, an originally higher resolution of the CT scans could also improve this further, but this brings us back to one of the basic dilemmas of this thesis, namely that this would result in very small specimens whose informative value with regard to mechanical characterization parameters such as the FVC is low (not to mention the fact that a corresponding device quality is also required for this). The aforementioned increase in noise at low resolutions further complicates the evaluation, so that even small, high resolution samples in combination with corresponding larger but lower resolution scans is not an optimal solution.

The direct use of 3D scans should be emphasized as a unique selling point compared to other CNNs in the field of materials science that work with CT scans as training data [334, 335]. No prior slicing is required. The complete evaluation of the 3D scan, of course, slows down the calculation enormously. With regard to the above-mentioned point of computational resource scarcity, the use of 2D slices instead is of course very helpful. It also simplifies the use of pre-implemented methods from tensorflow (but also pytorch), as most of them are designed for 2D images. In general, the development effort and also the training effort compared to the AOA thresholding must of course be emphasized negatively. However, once trained, the effort required to determine the FVC is considerably less than the experimental effort.

5.3 Fiber length distributions

The results of the fiber length evaluation in Figure 4.15 and Figure 4.16 show an average fiber length of just over one millimeter. While this does not sound like much for a long fiber reinforced material, it corresponds to an aspect ratio of a good 150 for carbon fibers of 7 µm, which can certainly be counted as long fibers. However, the fact that the median is less than half as large at around half a millimeter raised the question of the correct means of a distribution function. In fact, depending on the choice of the mean value in a model, significant differences can be achieved. This would be a separate mathematical topic that shall not be addressed here.

It was also noticeable that the average fiber lengths in the charge area appeared to be smaller than in the flow area. This is consistent with the results of Radtke [321]. He also found that the fiber length distribution at the end of the flow path showed a shift towards longer fibers. Such an effect is also observed directly behind the insertion area, which corresponds more to the position of the specimens FLD_{10} - FLD_{12} in this study. According to Radtke, the proportion of fine fibers at the end of the flow path is lower than in the charge area, which means that greater average fiber lengths are observed towards the end of the flow path. Longer fibers offer more interaction with the melt and are therefore entrained for longer. This is consistent with the fiber length distributions in Figure 4.15.

It is also noticeable that the average lengths increase with the size of the specimens. This is not surprising, as longer fibers are cut off at the edge of smaller specimens and it is therefore statistically more likely to have fewer long fibers in the specimen. These long fibers have a strong influence on the average fiber length and therefore raise the average. As this was no longer quite so clearly visible between the medium-sized and large specimens, it was assumed that this effect is saturated at around 25 mm side length (due to the fact that hardly any fibers are longer than this side length). For this reason (and also because, for example, the diameter of the round samples for commercial FVC determination

by ashing was also 25 mm), this size was used for the FVC investigation and retained for the microstructure GAN.

In principle, the average fiber lengths are rather short compared to the results of Radtke [321] and Tröster [320]. However, both used glass fibers and it is assumed that the thin and fragile carbon fibers are cut/destroyed much more easily by the screw extruder and the shearing. Tröster also explains that a maximum stiffness of 95 % is already achieved with a fiber length of approximately 1 mm [320], see also [5, 336]. In systematic studies of the influence of fiber length, Thomason and Vlug even found no dependence of stiffness on fiber length above 0.5 mm as long as FVC and FOT are the same [337]. However, the influence on the strength is still existent, otherwise there would be no need for long fibers.

5.4 Fiber orientation distributions

The measured fiber orientation distributions and fiber orientation tensors, of which selected ones were shown in the Results, showed partly expected and partly surprising characteristics, which are discussed in the following.

5.4.1 A_{11} dominant and planar orientation

Looking first at the measured orientations of Plate 3 and Plate 4 in Figure 4.33, for example, it becomes clear that the tensors are - as expected - very much aligned with the direction of flow of the material in the press. The tensors of Plate 3 are printed below for clarification. In addition, the main components of the nine tensors are shown again in Figure 5.4, where the dominance of the A_{11} component becomes even more visible.

$$\boldsymbol{C_1} = \begin{bmatrix} 0.798318 & 0.153878 & 0.004610 \\ 0.153878 & 0.195227 & 0.006829 \\ 0.004610 & 0.006829 & 0.006455 \end{bmatrix}$$

$$\boldsymbol{C_2} = \begin{bmatrix} 0.902426 & 0.055382 & 0.009222 \\ 0.055382 & 0.090359 & 0.006853 \\ 0.009222 & 0.006853 & 0.007215 \end{bmatrix}$$

$$\boldsymbol{C_3} = \begin{bmatrix} 0.810983 & 0.098676 & 0.003675 \\ 0.098676 & 0.185093 & -0.002959 \\ 0.003675 & -0.002959 & 0.003924 \end{bmatrix}$$

$$\boldsymbol{CF_1} = \begin{bmatrix} 0.821371 & -0.026740 & -0.011685 \\ -0.026740 & 0.176429 & 0.001062 \\ -0.011685 & 0.001062 & 0.002200 \end{bmatrix}$$

$$\boldsymbol{CF_2} = \begin{bmatrix} 0.923717 & 0.032840 & -0.005360 \\ 0.032840 & 0.074781 & -0.000746 \\ -0.005360 & -0.000746 & 0.001502 \end{bmatrix}$$

$$\boldsymbol{CF_3} = \begin{bmatrix} 0.876894 & 0.080256 & -0.008694 \\ 0.080256 & 0.121239 & -0.000411 \\ -0.008694 & -0.000411 & 0.001867 \end{bmatrix}$$

$$\boldsymbol{F_1} = \begin{bmatrix} 0.871685 & 0.204490 & -0.002035 \\ 0.204490 & 0.124755 & 0.003296 \\ -0.002035 & 0.003296 & 0.003560 \end{bmatrix}$$

$$\boldsymbol{F_2} = \begin{bmatrix} 0.851010 & 0.086686 & 0.005897 \\ 0.086686 & 0.140799 & 0.009363 \\ 0.005897 & 0.009363 & 0.008191 \end{bmatrix}$$

$$\boldsymbol{F_3} = \begin{bmatrix} 0.749631 & -0.234513 & -0.000550 \\ -0.234513 & 0.247066 & -0.001583 \\ -0.000550 & -0.001583 & 0.003303 \end{bmatrix}$$

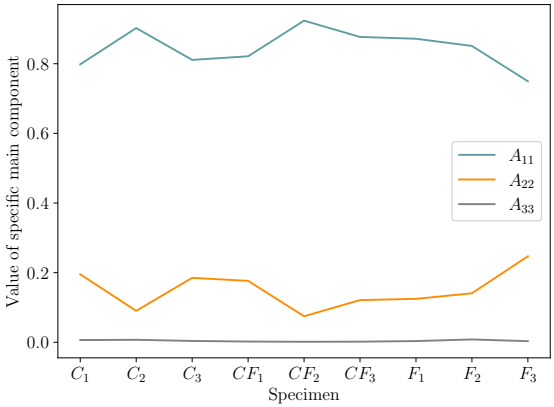


Figure 5.4: Main components of the nine measured tensors of Plate 3 (CF-PA6).

However, if you look at the tensors of Plate 1 (cf., e.g., blue tensors in Figure 4.24), you can see that the A_{22} component is also dominant in some cases and not all tensors are strongly aligned in the direction of flow. This difference is caused by the different specimen sizes. The 10 mm $\times$ 10 mm large specimens of Plate 1 cover very small, local areas where a fiber bundle part can sometimes be oriented against the flow direction and this is not compensated by enough other fibers and fiber bundles that are oriented "normally". An A_{11}-dominant orientation can therefore certainly be assumed, as the results of Plate 3 (and 4 and the hybrid one) are to be regarded as more generally valid. However, as a starting scenario for the tensor interpolation, this rather complex state of Plate 1 was not undesirable in order to test the methods for difficult cases.

It is also noticeable that all the orientation tensors/distributions measured over an entire specimen are almost planar states. The A_{33} component is always below 0.1 (in the case of Plate 3 even below 0.01, cf. Figure 5.4). This was also to be expected due to the thicker plastificate, the thinness of the final plate and the resulting flow path.

5.4.2 Orientation behavior across thickness

Figure 4.19 shows curves of the main components of the fiber orientation tensors over the thickness of different specimens. For the carbon fiber reinforced specimens, similar plots of the main components were found for the same specimen positions as for the glass fiber reinforced specimens. Slight differences at the edge of the thickness may also be due to slightly different cuts of the scans of the samples. Cutting away the unevenness at the edges correctly without taking away too much of the specimen is challenging. In contrast, what clearly differs are the curves of different specimen positions; especially the curves of charge specimens and flow specimens. The curves of flow specimens were relatively predictable with constant values across the thickness and a strong A_{11} component. In contrast, the A_{11} component of the specimens in the charge region drops sharply at both edges and the A_{22} component increases there.

While this effect is also a kind of shell-core effect, it is not the shell-core effect known in injection molding, which occurs due to the fountain flow [5]. Because the mass has to flow from the center to the edge, the fibers in the core layer are mainly directed against the flow and in the shell layers rather in the direction of the flow [5]. This is not the case here. Instead, the observed courses are due to a different phenomenon: The plastificate is inserted in the charge area and the fibers in its outer layer there are mainly oriented in the extrusion direction, i.e. e_{22} direction. On contact with the "cold" tool surface, the orientation there freezes directly as it appears in the plastificate. In the flow area, however, the orientation of the material flow in the e_{11} direction naturally dominates. This is consistent with the results of Radtke [321]. He also claims that the longitudinal orientation prevailing in the outer layers of the plastificate remains unchanged in the test plates in the insertion area, as it freezes immediately on contact with the mold [321].

5.4.3 Skewed flow front

As already seen, the expected main flow direction also developed in the plates. Due to the fact that the length of the plastificate does not extend completely over one side of the plate, a slight widening of the flow in the direction of the upper and lower edge of the plate at the end is to be expected, as shown in Figure 5.5.

In fact, this phenomenon can also be seen, e.g. in the tensors of the plates on the left-hand side in Figure 4.33. In addition, however, the averaged tensors of these plates (see middle column in Figure 4.33) are never completely oriented in the e_{11} direction (or also 0° direction). Instead, they all point slightly upwards. This was already evident in the tensors of second-order of Plate 1 with the smaller specimens, which was used for the interpolation method evaluation (see e.g. Figure 4.24). This deviation from the extrusion direction has already been partially observed in literature [320, 321]. The reasons for this could be manifold. An uneven tool in the press would immediately come to mind, but this was checked and did not appear to be the case. This is also contradicted by

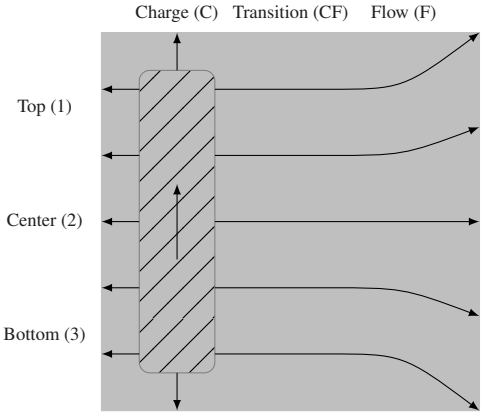


Figure 5.5: Schematic representation of the expected, mostly one-dimensional flow pattern based on the initial situation.

the fact that an attempt was made to insert the plastificate rotated by 180° and a downward drift was detected. It therefore appears to have been caused by the plastificate itself, which seems to be inhomogeneous in some way. Tröster comes to the same conclusion: In addition to the insertion position of the plastificate, the direction of the plastificate is also decisive for the resulting fiber orientation and, in particular, deviations from the flow direction. However, he does not investigate further why this is the case. Radtke [321] investigates the cause by using thermographic images. He records a temperature difference of approx. 10 K - 30 K between the front and back ends of the plastificate strand. As a result of the plastificate being discharged from the plastificate nozzle, the plastificate cools slightly more at the end that is discharged first. The tunnel heater at the outlet of the discharge nozzle partially, but not completely, compensates for this heat loss. He claims that this temperature gradient in the plastificate strand results in a difference in viscosity in the strand. This could lead to a slightly rotated orientation of the flow front relative to the mold edges. [321] However, a clear temperature difference could not be detected in measurements in this case. In turn, it is assumed that the geometry and/or the porosity of the plastificate could play a decisive role. Corresponding investigations have already been shown in Section 5.1. Radtke also mentions the so-called lofting (expansion of

the strand due to the recovery of the relatively rigid fibers, which are not fixed in the molten matrix) at the front end of the plastificate [321]. However, since the fibers are quite short, the lofting could also be caused by the matrix. The short dwell time in the extrusion die subsequently leads to strand expansion. This lofting leads to increased deviation of the cross-section across the extrusion direction and the geometry itself (elongation of the front end in the extrusion direction) and to increased porosity of the front end. These effects could enable such a skewed flow front and thus lead to a deviation of the main orientation and hence to material anisotropy.

5.4.4 Interpolation of fiber orientation tensors

In the following, the fiber orientation tensor interpolation methods are discussed and evaluated.

5.4.4.1 Component averaging method [3]

The weighted, arithmetic averaging of the components based on the Euclidean distance is a linear approach and by far the simplest. It impresses with its calculation speed and is superior in terms of simplicity of implementation and general complexity. It is therefore still frequently used. The problem of tensor swelling already discussed in the State of the Art [122–125] was also evident in this work. Tensor characteristics were therefore not monotonically interpolated, which is considered a disadvantage. In the course of this, the scientific question of distinguishing between interpolation and averaging also arose (cf. [338]). This type of component averaging can certainly be used with a clear conscience when it comes to determining the mean value of a set of tensors. Interpolation, however, searches for values between given ones, which this method cannot provide in the author's opinion.

[3] This section is based on the author's publication [17].

5.4.4.2 Decomposition method [4]

The decomposition method impresses with its monotonic behavior. This is also visually appealing in the agile rotational behavior of the tensors and meaningful changes in shape, which was also observed by other authors [123, 125]. Although mathematically more complex, the calculation is still extremely fast, at least for the smaller scopes tested. It was impressive to see how equally good the interpolation was when the same nine given tensors were provided in different coordinate systems (rotated by angles). An alternative to the method, which already works very well as it is, would be to use projectors instead of quaternions for rotation interpolation, as implemented by Krauß and Kärger [125]. These projectors have the decisive advantage that they are unique for a given tensor.

5.4.4.3 ANN [5]

The ANN obviously requires the most time to output the interpolated tensors. With training times of less than an hour on an office laptop (Intel(R) Core(TM) i5-10210U CPU @ 1.60 GHz, 2.11 GHz and 16.0 GB RAM) with the subsequent output of 160 tensors, this is nevertheless still fast compared to around two hours for a single scan and tensor. However, the increased implementation effort for such a network must be taken into account. In this case, it was actually low, as a network already developed for a similar application was adapted. The interpolation of the ANN is not very convincing with non-monotonic behavior similar to component averaging. With further development, nonetheless, a better result could be expected. A more general question is of course whether a neural network, which relies on a lot of training data, is at all useful for an application with a data set this small. Presumably, the network would have to be given additional input for better performance. Working with scarce (or sparse) data

[4] This section is based on the author's publication [17].
[5] This section is based on the author's publication [17].

and the challenges involved have accompanied artificial neural networks since the beginning of their use phase [339] and are a field of research in their own right. Solutions range from transfer learning [340] and novel approaches for data augmentation [341] to the development of novel, "greedy" algorithms [342]. The additional use of the data employed by Sabiston et al. [133] alongside the nine tensors in this network of course did not lead to any improvements as different specimen positions were used and other initial basics differed. Alternatively, this network's application must be restricted to the prediction of FOT of an entire process with thousands of values and not a single plate, as by Sabiston et al. in [133], for which this network was also developed and where it was convincing. It has another very decisive disadvantage: The other two methods work instantaneously for each following set of tensors between which they are supposed to interpolate. The ANN, however, must theoretically be retrained for a new initial state.

5.4.4.4 Quantitative comparison of the three methods [6]

The interpolation methods were also analyzed quantitatively in the Results by omitting one measured tensor in each case and determining it using the respective method. The Frobenius norm of the measured and calculated tensors was subsequently calculated. The difference between these norms, representing the difference between two tensors in one value, was plotted for each method in the Results chapter (Section 4.6). However, the question arose as to whether this reduction of the tensor information to a single value of this norm can do justice to this question. Hence, the absolute differences of the individual tensor components per method can be seen in Figure 5.6 and the averaged differences for all nine tensors per component and method in Table 5.1.

It is immediately apparent that the component averaging method performs best according to this type of error determination, ahead of the neural network and

[6] This section is based on the author's publication [17].

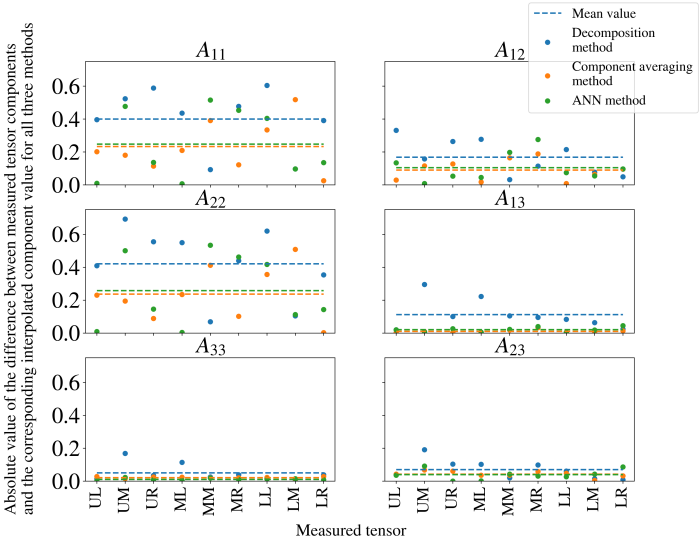


Figure 5.6: Comparison of all methods concerning the absolute difference between the same component of the measured and corresponding interpolated tensor. Extracted from Blarr et al. [17].

Table 5.1: Deviation of calculated value to measured value of a specific component for a specific method averaged over all nine measured tensors. Extracted from Blarr et al. [17].

Component	Average absolute error of different methods		
	CA	D	AI
A_{11}	0.233	0.401	0.248
A_{12}	0.09	0.169	0.105
A_{13}	0.009	0.112	0.02
A_{22}	0.236	0.421	0.258
A_{23}	0.043	0.071	0.04
A_{33}	0.021	0.05	0.011
MAE	0.105	0.204	0.114

with the decomposition method performing worst. This hardly coincides with the observed interpolation behavior. The shortcomings of the considered error measure will therefore be discussed here. First of all, this type of "substitute calculation" implies a change in the initial state, as the data set from which the calculation is made is reduced by 1/9. Furthermore, the determination of the corner tensors is particularly problematic if they are omitted: it resembles an extrapolation, as the position is outside the grid of given tensors. Thus, a fundamentally different property of the methods, that of extrapolation and not interpolation, is assessed. In fact, this statement is confirmed in so far as the error values are significantly reduced if these corner tensors are omitted from the calculation - especially considering the decomposition method (see Table 5.2).

Table 5.2: Deviation of calculated value to measured value of a specific component for a specific method averaged over the five tensors of $\mathcal{T}_m$ without the corner nodes ($\boldsymbol{UL}, \boldsymbol{UR}, \boldsymbol{LL}, \boldsymbol{LR}$). Extracted from Blarr et al. [17].

Component	Average absolute error of different methods		
	CA	D	AI
A_{11}	0.158	0.181	0.172
A_{12}	0.061	0.073	0.065
A_{13}	0.007	0.087	0.01
A_{22}	0.161	0.206	0.179
A_{23}	0.022	0.047	0.023
A_{33}	0.01	0.039	0.008
MAE	0.07	0.106	0.076
Deviation to error with all tensors	0.035	0.098	0.038

However, it remains unclear whether this direct difference between the tensor components is suitable as a measure of error either. It has been shown that the components alone do not seem to be sufficient for a description of their

nature, but only the alternative description via invariants and eigenvectors made it possible to grasp their structure at all.

In addition, there are of course further points for discussion. It is certainly important to mention the unusually complex initial state with very different rotations of the measured tensors, which does not represent the typical fiber course that occurs in this process. It became clear in the course of the dissertation that the chosen sample size of 10 mm $\times$ 10 mm was quite small and reflects very local effects. For the comparison with experimental values, which will follow in this discussion, significantly larger samples were therefore chosen to better represent the process behavior. As already mentioned, however, this allowed the methods to be tested in a kind of worst-case scenario, which in turn was not undesirable. Many mathematically motivated methods for tensor interpolation are often not even tested with more than two tensors, let alone more complex initial states, which makes realistic or non-academic use rather difficult. This was hence intended to be avoided in this work.

The general challenge is the use of methods to solve an algebraic problem for an application that is actually subject to physical boundary conditions and flow phenomena. This naturally raises the question of whether physical conditions (e.g., insertion of the plastificate or orientation information of the plastificate) should be added to such a method. Nevertheless, the decision was made not to pursue this, as it was perceived to be an unnatural means of achieving the desired outcome. For the same reason, a restriction of the possible angles was not implemented, which, e.g., Krauß and Kärger have done [125]. Brannon even argues that methods for mixing rotations must be selected according to the physical application [343]. This speaks against the existence of a universally valid procedure.

There are many possibilities for improvement. First of all, adjustments to the methods implemented here are conceivable. The aforementioned restriction of the possible angles could be achieved by comparing the scalar products of the possible quaternions involved. By choosing the combination with the maximum scalar product, the smallest angle is obtained. Normalization of the quaternions

would also be important in order to obtain unit quaternions for additions. The ANN could certainly be optimized by the number of layers, depth of the layers, learning rate, batch size, choice of loss, etc. In particular, the increased isotropy of the tensors in ANN (similar to the swelling effect in component averaging) should be addressed.

As a completely different approach, the direct interpolation of fourth-level tensors was tackled by some of the authors of the paper [17] with a complex method in another paper [338]. In addition, the author's simplified decomposition method for fourth-order tensors was also successful, which will be discussed below.

As far as further investigations are concerned, a sensitivity analysis with regard to specimen geometry and location and different tensor fields (different plates) should be worthwhile. In particular, more central specimens should lead to better results (especially for the decomposition method). The methods are also applicable for 3D geometries, so that they should be tested with parts of more complex geometries (especially concerning the z axis) than the plates.

5.4.4.5 Validation of fourth-order tensor interpolation through simplified decomposition method [7]

Benedikt Scheuring carried out tensile tests on tensile specimens taken from the plates at different angles [274]. This enabled stiffnesses to be determined for the different load directions. The mean values of these stiffnesses for multiple tensile tests on specimens from the flow area of the plates are shown in a polar plot for the CF plate in Figure 5.7 and for the GF plate in Figure 5.8.

The flow area was chosen for comparison because of its fundamentally more uniform structure. However, the differences to the charge area are not significant. The values determined were mirrored accordingly due to the expected symmetry.

[7] This section is based on the publication [274], co-authored by the author.

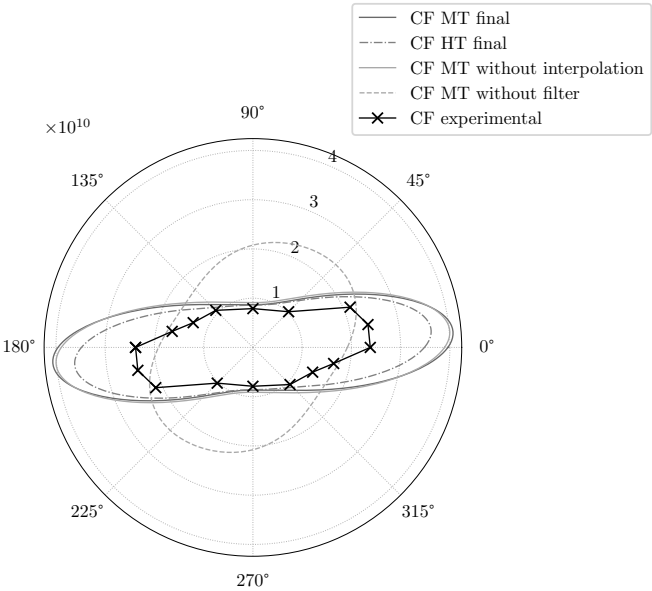


Figure 5.7: Polar plot with the experimental stiffnesses, the modeled stiffness through Halpin-Tsai and different Mori-Tanaka-modeled stiffnesses depending on the input fiber orientation tensor for Plate 3 (CF-PA6). Based on experimental and modeled data published in [274].

The higher reinforcement in 0°, i.e. flow direction, compared to 90° is just as recognizable as a drift of the highest stiffness values to about 5° instead of 0°, which was already seen in the CT evaluation. These observations apply more or less to both the CF and the GF material. The most noticeable difference between the two is that the ratio between the 0° and 90° directions differs. The CF material is about three times stiffer in the flow direction than at 90°, whereas the ratio for the GF material is only about two. This is due to the higher stiffness of the carbon fibers themselves.

The fiber orientation information determined by the author has then been used in turn for modeling approaches to compare the resulting stiffnesses with the experimental ones. The Halpin-Tsai modeling with a shear-lag modification

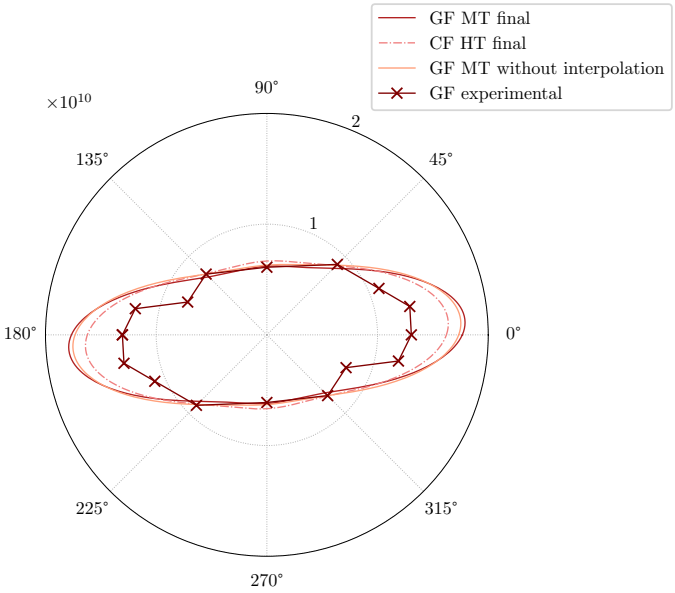
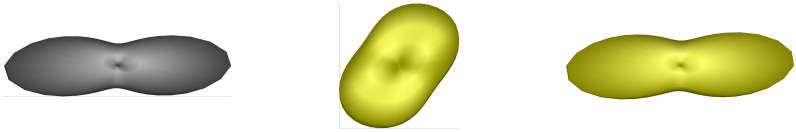


Figure 5.8: Polar plot with the experimental stiffnesses, the modeled stiffness through Halpin-Tsai and different Mori-Tanaka-modeled stiffnesses depending on the input fiber orientation tensor for Plate 4 (GF-PA6). Based on experimental and modeled data published in [274].

that has been explained in the State of the Art (cf. Section 2.1.2.2) required the determined orientation histograms (cf. Figure 4.33). Therefore, all occurrences of angles of the nine specimens of a plate were averaged. The Halpin-Tsai stiffnesses were then calculated with the Python package HomoPy [344]. However, for Mori-Tanaka (cf. Section 2.1.2.4) fiber orientation tensors of fourth-order were required. Several approaches were under consideration for the input tensor. First, the nine measured tensors of fourth-order were acquired with the same approach for both the CF plate and the GF plate and averaged. Additionally, the adapted tensor determination approach with the preceding median filter was applied to the scans of the CF plate and the resulting nine tensors were averaged. Finally, the presented simplified decomposition approach for tensors

of fourth-order (cf. Section 3.7.5) was applied on both the CF and the GF plate and the resulting 169 tensors were averaged, respectively. As an overview, the resulting averaged tensors are plotted as tensor glyphs in Figure 5.9 and Figure 5.10.



(a) (b) (c)

Figure 5.9: Tensor glyphs of different averaged tensors of the CF plate. The basis for these averaged tensors were (a) all measured and interpolated tensors with the preceding median filter, (b) all measured and interpolated tensors without median filter in the evaluation and (c) only the measured tensors with preceding median filter.

(a) (b)

Figure 5.10: Tensor glyphs of different averaged tensors of the GF plate. The basis for these averaged tensors were (a) all measured and interpolated tensors and (b) only the measured tensors.

These averaged tensors were then employed to calculate the stiffnesses with the Mori-Tanaka approach, again with the help of the package HomoPy [344]. The resulting stiffnesses are also plotted in Figure 5.7 and 5.8.

In the polar plots, it is initially noticeable that the stiffness curve, which is based on the orientation evaluation without a previous median filter, appears to be rotated by almost 45° and the shape also appears far too isotropic. The pre-filtering therefore seems to have an enormous effect, which leads to a much

more realistic result. It is also noticeable that all other stiffness curves, whether experimental or modeled, have approximately the same angle with the greatest stiffness, both for the CF and the GF plate. This indicates a good agreement between the measured fiber orientation tensors and the actual/experimentally determined fiber orientation distribution in the plate. What is also noticeable for both plates, however, is that the modeled stiffnesses significantly overestimate the actual values. This is somewhat more pronounced for the CF plate. Furthermore, this overestimation is slightly greater with the Mori-Tanaka modeling than with Halpin-Tsai. It can be assumed that this error is also due to the fact that neither the one nor the other model was designed for such a complex mixed single fiber and bundle structure. In particular, it was noticed that specimens in the tensile test often failed on non-fully impregnated bundles. Although the failure behavior is not directly related to the Young's modulus or the stiffness, it is possible that these non-impregnated or only partially impregnated fiber bundles already contribute only partially to the force transmission in the linear-elastic section. As a result, it is possible that the true mean fiber length and also the true fiber volume content may be significantly lower than what was measured experimentally and given to the models accordingly. Fiber waviness could also play a role in the reduced effective stiffnesses. In the future, it would be interesting to draw conclusions about possibly reduced material parameters by measuring the frequency of larger bundles and their length/width/volume. A further development of previous homogenization models would of course be conceivable too. It is also noticeable that the stiffness curves of the Mori-Tanaka model do not differ greatly, regardless of whether only the nine measured or also the interpolated ones were averaged. In fact, this is a strong argument in favor of the interpolation method in this case, because the corresponding initial situations of the plates were quite uniformly anisotropic (oriented in the 11 direction) and no strongly deviating tensors would have been expected, which seems to be confirmed. Overall, these results show how well the orientation measurement agrees with the experimental values.

In general, the simplified decomposition method for tensors of fourth-order by the author yields very good results (cf. tensor fields in Figure 4.33). In

fact, it performed better than the much more complex and mathematically more sound method by Bauer et al. [338], when used in this context of a rather big two-dimensional tensor field.

5.5 Microstructure generation through generative adversarial network [8]

First of all, it should be mentioned that the use of the GAN for the generation of artificial CT images of the material of this work should be considered as a kind of feasibility study. Although there is no immediate application of the generated images, it was important to be able to assess the enormous potential of these networks in relation to the generation of rather irregular structures such as the DicoFRP. With the help of some further developments, which will be discussed below, they could be of great help.

5.5.1 Assessment of generated images

After much optimization of the network, generated images of the final network architecture look surprisingly convincing. While some images show either an unusually large number or barely any structures recognizable as fibers and some images seem to combine attributes of different images in an unsuitable way, there are also images that are difficult to distinguish from the original even for the trained eye. This variance is also evident in the quantitative evaluation methods. First, the signal-to-noise ratio of the output images was considered, which, with a value of 3.70, was lower than that of the input images, but still clearly above 1, indicating the presence of meaningful signals in comparison to the background noise.

[8] This section is based on the author's publication [275].

Upon examination of the grey value histograms, it becomes evident that the output images exhibit a flatter profile and a darker maximum value compared to the input images (cf. Figure 5.11).

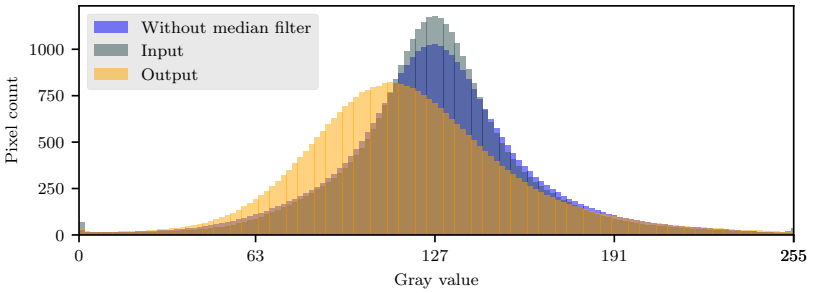


Figure 5.11: Global histogram of gray value intensities of all input images without median filter and with median filter as well as of the batch of 128 output images. Extracted from Blarr et al. [275].

The underlying cause of this deviation is challenging to ascertain, as are potential corrective measures. One potential solution to achieve a more accurate match between the histograms is to incorporate the mean value of the histograms or even the entire distribution as additional input (cf. the following explanation of conditional GANs (cGANs) and InfoGANs). In the context of neural networks, evaluation is typically based on loss plots, which provide insights into the convergence and accuracy of the network's predictions. However, the loss plot of this GAN exhibited significant oscillations and did not demonstrate a continuous decrease (cf. Figure 4.35). Nevertheless, the training was sufficiently stable. It is assumed that the oscillating behavior of the generator loss is due to the fact that the discriminator becomes very good very quickly and thus later outputs very similar probabilities. The generator therefore receives little meaningful feedback. Both vanishing gradients and general convergence problems could be responsible for this failure mode. This behavior of the loss is also not unusual, which has already been mentioned in the Results (cf. Section 4.7). The State of the Art (Section 2.4.2) also elucidated that it is nearly impossible to

attain an optimal equilibrium between the generator and discriminator, as the optimal point is a saddle point and not a minimum as for most other types of neural networks [242]. Nevertheless, since the generator loss does not increase exponentially, a kind of metastable state is assumed here. However, it is also evident that further metrics must be employed to assess the quality of the network. Subsequently, quantitative metrics were employed to directly evaluate performance on the generated images, which will be discussed hereinafter.

The FID plot (cf. Figure 4.36) illustrates the mean FID of 128 generated images per epoch. The improvement in the generated images can be observed as the distance decreases. However, this decrease plateaus after a certain point, which suggests that there is no further improvement in the images. To investigate this in more detail, the development of a single generated image was assessed over the epochs. In the initial example (cf. Figure 4.37), it is not possible to determine whether the image "improves" in the final 30 epochs. This suggests that an earlier termination of the training would also be possible. In contrast, in other examples (cf. Figure 4.38), it appears that the final image is the best. It is evident that the structures within the image continue to undergo significant alterations in the final epochs, which can be attributed to the fluctuations observed in the loss plot. However, the inability to make an absolute assessment of which image represents the optimal outcome precludes the formulation of a clear recommendation regarding the optimal training duration. Nevertheless, these examples illustrate that the images do not suddenly deteriorate with further training. Additionally, it is unlikely that any discernible improvement will be observed beyond the 75th epoch in comparison to the computational effort. Returning to the FID plot, it can be observed that the final value is relatively high possibly indicating bad final image results. However, it is important to note that the FID is highly dependent on the number of samples. In fact, the fewer the samples, the higher the score [312]. Therefore, it is essential to exercise caution when assessing absolute values and comparing them with those of other networks. This is particularly relevant when considering CT images, which present a completely different set of conditions than those encountered with most images generated by such networks. A comparison of the ImageNet

dataset, on which the FID is pre-trained, with the gray scale images reveals that the former contains everyday, coloured photos with clearly definable objects, people, etc., which cannot be compared with the stochastically distributed fibers in the latter. Furthermore, similar FID values can also be found in other papers with good results [345].

The capacity of the DCGAN to generate novel images was investigated by searching for nearest neighbors in the training dataset for each generated image. Two different metrics, ED and SSIM, were employed as a basis for this investigation (cf. Figure 4.39). Both metrics demonstrated an ability to select appropriate images as nearest neighbors and both were able to demonstrate that the network generates images that differ significantly from training images. The ED metric, despite being the more straightforward and faster method to calculate, was unable to find suitable nearest neighbors in some instances for highly novel images. This is because the ED is not capable of recognizing similar structures that are offset slightly in spatial location as similar [319]. As a result, it is often the case that neighbors determined using SSIM are perceived to be closer to those visually observed. The values of the ED to the nearest neighbor and the SSIM were also plotted for the two single image developments (cf. Figure 4.40 and Figure 4.41 for the first image in Figure 4.37 and Figure 4.42 and Figure 4.43 for the second one in Figure 4.38). All plots exhibited a decrease at the beginning, followed by an increase in the first example or a period of relatively constant performance apart from fluctuations. This corroborates the difficulty previously identified in the context of the FID in determining when an image has reached a point of no further improvement. Apparently, after just a few epochs, images are generated to which similarly close neighbors can be found as to the final generated image. It is noticeable that low distance values, i.e. high correspondence with a training image, often occur for images with few structures recognizable as fibers. Presumably, these are then structurally so homogeneous that an image with equally few fibers in the training data set simply shows a high degree of correspondence. However, since there are even more images with many fibers in the training data set, but their distance to similar generated images is typically greater, it is recommended

that these nearest neighbor determinations are not primarily used to assess the development of a single generated image. The FID is probably more suitable for this purpose. In principle, however, both the nearest neighbor method itself and the two metrics ED and SSIM proved to be quite helpful for assessing the network. ED is particularly useful for quick comparisons and SSIM for more precise ones.

5.5.2 Further development

Further developments of this network are conceivable and recommended. One option would be to provide the network with additional information about the training images. Various quantities are possible here. In addition to purely image-based information such as the Haralick entropy from the Haralick features [346], which provide information on the heterogeneity of the images, the mechanical parameters mentioned in this paper, FVC, FLD and FOD/FOT, would be of particular interest. This physical information would ensure that the network is able to recognize and reproduce superordinate structural relationships between individual voxels. The network would have to be modified accordingly. In literature, a so-called conditional GAN, or cGAN for short, was first introduced, which enables the assignment of a specific label to each training image [254]. This was tested for this problem by assigning a value of its Haralick entropy to each image. However, the significance of such a single value is low, and accordingly this did not improve the network. Instead, it would be more beneficial to evaluate several values of entropy across the image in a grid. Alternatively, the mechanical parameters previously mentioned should definitely be employed. Subsequent developments, such as the continuous conditional GAN (CcGAN) [255] or the so-called InfoGAN [256], are more suitable for this purpose. A further development (to a continuous conditional DCGAN, CcDCGAN) with mechanical parameters could be used to develop a true microstructure generator that can generate images with a specific FVC, FLD or FOD requested by the user. Furthermore, the development of a network

that can process and generate 3D images must be pursued in the long term in order to exploit the potential for the creation of RVEs.

6 Final remarks

6.1 Summary

In this work, investigations into the plastificates were first presented. It was shown that both the rotational speed during extrusion and the lofting of the plastificates have an influence on their porosity. As expected, the orientation results show double circular structures in the cross-section due to the twin-screw extruder and almost perpendicular orientation to the extrusion direction.

To determine fiber volume contents from the scans, a slightly adapted thresholding method and a convolutional neural network were developed and validated using experimentally obtained data. While the network performed well, the AOA thresholding was sample-wise better. Further development and inclusion of samples with more variable fiber volume contents would be useful. The second research question formulated in the introduction aimed at how quantities such as fiber volume content can be reliably extracted from CFRP scans, which are more difficult to process. This work shows that adaptation of known methods, but above all AI-based methods, can certainly provide a remedy for low-contrast or noisy images. However, eliminating the cause and generating better images in the first place would of course be an even better measure, but this was not possible within the scope of this dissertation.

With regard to the interpolation of second-order fiber orientation tensors, three different methods were tested, whereby the decomposition method based on spectral decomposition was the most convincing. It was then applied in simplified form to fourth-order fiber orientation tensors. The resulting averaged

tensors could in turn be used without closure approximation in Mori-Tanaka homogenization and the resulting modeled stiffnesses could be compared with experimentally determined stiffnesses. Concerning the orientation, the results were largely in agreement. This tensor interpolation provides initial answers to the first research question formulated at the beginning, namely how information from smaller specimens can be transferred to larger dimensions. First of all, it should be noted that the consideration of information from smaller specimens instead of purely macroscopic observations makes sense considering the fluctuations of mechanical variables over, e.g., such a plate as in this work. In order to obtain a reference to a macroscopic behavior, interpolation methods such as the ones described above are suitable. Even for the more complex case of tensors, there are methods that can generate realistic interpolations. However, the resolution problem of CT scans for materials with such small constituents as carbon fibers greatly complicates the processing of medium-sized samples. Samples that are too small also exacerbate these methods though, as the information obtained can be so local and deviate so strongly from the global material behavior that a corresponding interpolation is again difficult. The problem has therefore not been conclusively clarified and would have to be checked in particular by interpolating fiber volume contents or fiber length distributions too.

Finally, the development and successful use of a generative adversarial network for the generation of artificial two-dimensional CT images was demonstrated. From the author's point of view, the third research question of this thesis can therefore be clearly answered in the affirmative. Realistic images could be generated by the GAN and there is also potential for future use to generate three-dimensional RVEs. However, the second part must be answered positively with reservations and caution, as this must be preceded by a great deal of further development work.

To address the final research question of the thesis: The enormous potential of AI-based methods, especially neural networks, for the application to such material science characterization questions could be clearly demonstrated.

The networks showed particularly good results especially for image evaluation/generation questions, where convolutional layers proved to be powerful. In the case of tensor interpolation, more complex algebraic approaches proved to be the better solution. However, the risks of AI-based methods should also be pointed out here. While the GAN generated very realistic images, their objective assessment was difficult. Still, the use of such generated images to co-train networks (as an extension of a "real" data set) should not be problematic and can certainly improve the performance of other networks. The overtraining that presumably occurred in the case of the CNN, which only output fiber volume contents in the area in which it was trained, should also be viewed critically. However, the conclusion that a bias in the training dataset leads to a bias in the results is straight-forward and therefore a lot of prior knowledge and work should be put into the creation of the training dataset. The argument concerning the unscientific nature of the "black box" of a neural network remains, as the depth of layers, branches and weightings are usually not comprehensible.

6.2 Conclusion

This dissertation shows how image analysis methods and especially neural networks can be used to determine micro- and mesomechanical parameters from CT scans of long carbon fiber reinforced polymers. This enables comprehensive characterization and scale-bridging of mechanical quantities in addition to classical macroscopic characterization.

The following conclusions can be drawn from this work:

Neural networks with convolutional layers are a powerful tool for image processing or image generation even of slightly noisy CT images. However, prior knowledge of materials science must be taken into account when selecting training data and network architecture. Interpolation methods can help to close the gap between microstructural variables that can be evaluated from small samples and macroscopic material behavior. However, it does not solve the

dilemma between resolution and sample size and the poor CT image quality due to the small size of the fibers and their low contrast to the matrix. A fundamental improvement of the images would of course still be desirable for all evaluation and processing methods. Furthermore, it can be concluded that the process - as expected - has an enormous influence on the final workpiece quality and that an increased understanding of the process is essential in order to improve it. There is still a great deal of research to be done on the LFT-D process in particular. In principle, image processing, whether AI-based or not, opens up enormous possibilities for fast, statistical microstructure analysis, even beyond the field of fiber reinforced polymers, and will sooner or later find its way into all aspects of materials science.

7 Author's publications

[127] J. Blarr, N. Kresin, C. Krauß, K. A. Weidenmann, W. V. Liebig, and P. Elsner, "Application of a tensor interpolation method on the determination of fiber orientation tensors from computed tomography images," in *Composites Meet Sustainability - Proceedings of the 20th European Conference on Composite Materials*. Anastasios, Vassilopoulos, 2022

[276] L. Schreyer, J. Blarr, K. Höger, N. Meyer, and L. Kärger, "Generation of initial fiber orientation states for long fiber reinforced thermoplastic compression molding simulation," in *Composites Meet Sustainability - Proceedings of the 20th European Conference on Composite Materials*. Anastasios, Vassilopoulos, 2022

[347] J. Blarr, "3D µCT images of specimens of carbon fiber reinforced polyamide 6 plaque, fiber orientation tensor data of these images, and three python code files for two different algebraic and one machine learning based tensor interpolation algorithms," *KIT Open*, 2022, dataset,
DOI: 10.5445/IR/1000153725

[17] J. Blarr, T. Sabiston, C. Krauß, J. K. Bauer, W. V. Liebig, K. Inal, and K. A. Weidenmann, "Implementation and comparison of algebraic and machine learning based tensor interpolation methods applied to fiber orientation tensor fields obtained from ct images," *Computational Materials Science*, vol. 228, p. 112286, 2023

[348] J. Blarr, "Computed tomography scans of 15 different carbon fiber reinforced polyamide 6 specimens (both original files (16 bit) and files of further cut scans (8 bit))," *KIT Open*, 2023, dataset, DOI: 10.35097/1707

[349] J. Blarr and S. Klinder, "2D images of CT scans of carbon fiber reinforced polyamide 6," *KIT Open*, 2023, dataset, DOI: 10.35097/1822

[188] J. Blarr, P. Kunze, N. Kresin, W. V. Liebig, K. Inal, and K. A. Weidenmann, "Novel thresholding method and convolutional neural network for fiber volume content determination from 3D µCT images," *NDT & E International*, vol. 144, p. 103067, 2024

[350] L. Schreyer, B. M. Scheuring, N. Christ, J. Blarr, C. Krauß, W. V. Liebig, K. A. Weidenmann, T. Böhlke, A. Hrymak, and L. Kärger, "Continuous simulation of a continuous-discontinuous fiber reinforced thermoplastic (CoDiCoFRTP) compression molding process," in *Proceedings of the twenty-third international conference on composite materials (ICCM23)*. Queen's University Belfast, 2023

[275] J. Blarr, S. Klinder, W. V. Liebig, K. Inal, L. Kärger, and K. A. Weidenmann, "Deep convolutional generative adversarial network for generation of computed tomography images of discontinuously carbon fiber reinforced polymer microstructures," *Nature Scientific Reports*, vol. 14, no. 1, 2024

[274] B. M. Scheuring, N. Christ, J. Blarr, W. V. Liebig, J. Hohe, J. Montesano, and K. A. Weidenmann, "Experimental and homogenized orientation-dependent properties of hybrid long fiber-reinforced thermoplastics," *International Journal of Mechanical Sciences*, p. 109470, 2024

[338] J. K. Bauer, C. Krauß, J. Blarr, P. L. Kinon, L. Kärger, and T. Böhlke, "Evaluation of a decomposition-based interpolation method for fourth-order fiber-orientation tensors: An eigensystem approach," *Mathematics and Mechanics of Solids*, p. 10812865241241002, 2024

[351] N. Afrasiabian, A. Elmoghazy, J. Blarr, B. Scheuring, A. Prahs, D. Schneider, W. V. Liebig, K. A. Weidenmann, C. Denniston, and B. Nestler, "Crystallization and crystal morphology of polymers: A multiphase-field study," *Journal of Thermoplastic Composite Materials*, p. 08927057241296472, 2024

[352] L. Kehrer, B. Scheuring, J. Blarr, and T. Böhlke, "Hydrothermal behavior of pure PA6 and homogenization of discontinuous long carbon fiber-reinforced PA6," *Proceedings in Applied Mathematics and Mechanics*, p. e202400135, 2024

8 Supervised student theses

Landa, Francisco Xabier (2022): Design of a biaxial tensile testing device for use in a computed tomography system. Master thesis. Karlsruhe Institute of Technology, Karlsruhe. Institute for Applied Materials (IAM-WK).

Klinder, Steffen (2023): Microstructure generation of carbon fiber reinforced polyamide 6 by a Generative Adversarial Network based on µCT data. Bachelor Thesis. Karlsruhe Institute of Technology, Karlsruhe. Institute for Applied Materials (IAM-WK).

9 Code and data availability

In the following section, links to retrieve the code and the corresponding data, such as the scan files, are provided for all major topics in this paper that have already been published by the author as first author. The respective link contains more detailed information (e.g. on the exact contents of the data set).

- FVC:
  - Code AOA: `https://github.com/jewelsbla/AOA_thresholding`
  - Code CNN: `https://github.com/jewelsbla/FVC_CNN`
  - Data: `https://doi.org/10.35097/1707` [348]
- FOT interpolation:
  - Code: `https://github.com/jewelsbla/oriopy`
  - Python package: Oriopy `https://pypi.org/project/oriopy/`
  - Data: `https://doi.org/10.5445/IR/1000153725` [347]
- GAN:
  - Code: `https://github.com/sklinder/microDCGAN`
  - Data: `https://doi.org/10.35097/1822` [349]

Bibliography

[1] E. Commission and D.-G. for Communication, "European green deal – delivering on our targets," https://ec.europa.eu/commission/presscorner/api/files/attachment/869807/EGD_brochure_EN.pdf.pdf, 2021, online, accessed 2024-06-24, Luxembourg City, Luxembourg.

[2] W. Frenz, "Bundes-Klimaschutzgesetz (KSG)," in *Klimaschutzrecht*, W. Frenz and S. Altenschmidt, Eds. Berlin: Erich Schmidt Verlag, 2022, pp. 525–841.

[3] T. Gänsicke, M. Goede, and J. Sandino, "Die Technische Motivation," in *Leichtbau in der Fahrzeugtechnik*, H. E. Friedrich, Ed. Springer Vieweg Wiesbaden, 2017, pp. 33–44.

[4] E. R. Fuchs, F. R. Field, R. Roth, and R. E. Kirchain, "Strategic materials selection in the automobile body: Economic opportunities for polymer composite design," *Composites Science and Technology*, vol. 68, no. 9, pp. 1989–2002, 2008.

[5] U. Gandhi, T. A. Osswald, S. Goris, and Y.-Y. Song, *Discontinuous fiber-reinforced composites: Fundamentals and applications*, 1st ed. Cincinnati: Hanser Publications, 2020.

[6] G. Mittal, K. Y. Rhee, V. Mišković-Stanković, and D. Hui, "Reinforcements in multi-scale polymer composites: Processing, properties, and applications," *Composites Part B: Engineering*, vol. 138, pp. 122–139, 2018.

[7] T. Tagawa and T. Miyata, “Size effect on tensile strength of carbon fibers,” *Materials Science and Engineering: A*, vol. 238, no. 2, pp. 336–342, 1997.

[8] A. A. Griffith and G. I. Taylor, “VI. The phenomena of rupture and flow in solids,” *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 221, no. 582-593, pp. 163–198, 1921.

[9] F. W. Zok, “On weakest link theory and Weibull statistics,” *Journal of the American Ceramic Society*, vol. 100, no. 4, pp. 1265–1268, 2017.

[10] R. Stewart, “Thermoplastic composites — recyclable and fast to process,” *Reinforced Plastics*, vol. 55, no. 3, pp. 22–28, 2011.

[11] E. Parodi, G. W. M. Peters, and L. E. Govaert, “Prediction of plasticity-controlled failure in polyamide 6: Influence of temperature and relative humidity,” *Journal of Applied Polymer Science*, vol. 135, no. 11, p. 45942, 2018.

[12] T. Böhlke, F. Henning, A. N. Hrymak, L. Kärger, K. A. Weidenmann, and J. T. Wood, *Continuous-discontinuous fiber-reinforced polymers: An integrated engineering approach*, 1st ed. Cincinnati: Hanser Publications, 2019.

[13] P. Pinter, B. Bertram, and K. A. Weidenmann, “A novel method for determination of fiber length distributions from µCT-data,” *6th Conference on Industrial Computed Tomography (iCT) 2016, 9-12 February 2016, Wels, Austria*, 2016.

[14] L. Schöttl, K. A. Weidenmann, T. Sabiston, K. Inal, and P. Elsner, “Fiber bundle tracking method to analyze sheet molding compound microstructure based on computed tomography images,” *NDT & E International*, vol. 117, p. 102370, 2021.

[15] T. A. Osswald and G. Menges, *Material science of polymers for engineers*, 3rd ed. Munich: Hanser, 2012.

[16] O. Geiger, F. Henning, P. Eyerer, R. Brüssel, and H. Ernst, "LFT-D: materials tailored for new applications," *Reinforced Plastics*, vol. 50, no. 1, pp. 30–35, 2006.

[17] J. Blarr, T. Sabiston, C. Krauß, J. K. Bauer, W. V. Liebig, K. Inal, and K. A. Weidenmann, "Implementation and comparison of algebraic and machine learning based tensor interpolation methods applied to fiber orientation tensor fields obtained from ct images," *Computational Materials Science*, vol. 228, p. 112286, 2023.

[18] W. Krause, F. Henning, S. Tröster, O. Geiger, and P. Eyerer, "LFT-D – a process technology for large scale production of fiber reinforced thermoplastic components," *Journal of Thermoplastic Composite Materials*, vol. 16, no. 4, pp. 289–302, 2003.

[19] R. Hill, "The essential structure of constitutive laws for metal composites and polycrystals," *Journal of the Mechanics and Physics of Solids*, vol. 15, pp. 79–95, 1967.

[20] R. Hill, "Elastic properties of reinforced solids: Some theoretical principles," *Journal of the Mechanics and Physics of Solids*, vol. 11, pp. 357–372, 1963.

[21] W. Voigt, "Ueber die Beziehung zwischen den beiden Elasticitätsconstanten isotroper Körper," *Annalen der Physik*, vol. 274, no. 12, pp. 573–587, 1889.

[22] W. Yu, "An introduction to micromechanics," *Applied Mechanics and Materials*, vol. 828, pp. 3–24, 2016.

[23] A. Reuss, "Berechnung der Fließgrenze von Mischkristallen auf Grund der Plastizitätsbedingung für Einkristalle," *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 9, no. 1, pp. 49–58, 1929.

[24] R. Hill, "The elastic behaviour of a crystalline aggregate," *Proc. Phys. Soc. A*, vol. 65, pp. 349–354, 1952.

[25] L. Schöttl, “Development of methods for the characterization of damage in sheet molding compounds by using in-situ computed tomography,” Doctoral thesis, Karlsruhe Institute of Technology, Karlsruhe, 2021.

[26] B. Z. Hashin and S. Shtrikman, “A variational approach to the theory of the elastic behaviour of polycrystals,” *Journal of the Mechanics and Physics of Solids*, vol. 10, pp. 343–352, 1962.

[27] R. F. Ker, “The design of soft collagenous load-bearing tissues,” *The Journal of experimental biology*, vol. 202, no. 23, pp. 3315–3324, 1999.

[28] S. W. Tsai and N. J. Pagano, *Invariant properties of composite materials*. Technomic Publishing Company, 1968.

[29] J. C. H. Affdl and J. L. Kardos, “The Halpin–Tsai equations: A review,” *Polymer Engineering and Science*, vol. 16, no. 5, pp. 344–352, 1976.

[30] S.-Y. Fu, G. Xu, and Y.-W. Mai, “On the elastic modulus of hybrid particle/short-fiber/polymer composites,” *Composites Part B: Engineering*, vol. 33, no. 4, pp. 291–299, 2002.

[31] H. L. Cox, “The elasticity and strength of paper and other fibrous materials,” *British Journal of Applied Physics*, vol. 3, pp. 72–79, 1951.

[32] S.-Y. Fu, B. Lauke, and Y.-W. Mai, Eds., *Science and engineering of short fiber reinforced polymer composites*, 2nd ed., ser. Woodhead Publishing Series in Composites Science and Engineering. Cambridge: Woodhead, 2008.

[33] J. Görthofer, “Microstructure generation and micromechanical modeling of sheet molding compound composites,” Doctoral thesis, Karlsruhe Institute of Technology, Karlsruhe, 2022.

[34] N. Meyer, L. Schöttl, L. Bretz, A. N. Hrymak, and L. Kärger, “Direct bundle simulation approach for the compression molding process of sheet molding compound,” *Composites Part A: Applied Science and Manufacturing*, vol. 132, p. 105809, 2020.

[35] J. Feder, “Random sequential adsorption,” *Journal of Theoretical Biology*, vol. 87, no. 2, pp. 237–254, 1980.

[36] B. D. Lubachevsky and F. H. Stillinger, “Geometric properties of random disk packings,” *Journal of Statistical Physics*, vol. 60, no. 5/6, pp. 561–583, 1990.

[37] S. Torquato and Y. Jiao, “Robust algorithm to generate a diverse class of dense disordered and ordered sphere packings via linear programming,” *Phys. Rev. E*, vol. 82, p. 061302, Dec 2010.

[38] S. R. Williams and A. P. Philipse, “Random packings of spheres and spherocylinders simulated by mechanical contraction,” *Phys. Rev. E*, vol. 67, p. 051301, May 2003.

[39] O. Pierard, C. Friebel, and I. Doghri, “Mean-field homogenization of multi-phase thermo-elastic composites: a general framework and its validation,” *Composites Science and Technology*, vol. 64, no. 10-11, pp. 1587–1603, 2004.

[40] B. Klusemann and B. Svendsen, “Homogenization methods for multi-phase elastic composites: Comparison and benchmarks,” *Technische Mechanik*, vol. 30, no. 4, 2010.

[41] J. D. Eshelby, “The determination of the elastic field of an ellipsoidal inclusion, and related problems,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 241, no. 1226, pp. 376–396, 1957.

[42] T. Mori and K. Tanaka, “Average stress in matrix and average elastic energy of materials with misfitting inclusions,” *Acta Metallurgica*, vol. 21, no. 5, pp. 571–574, 1973.

[43] S. G. Advani and C. L. Tucker, “The use of tensors to describe and predict fiber orientation in short fiber composites,” *Journal of Rheology*, vol. 31, no. 8, pp. 751–784, 1987.

[44] Y. Benveniste, “A new approach to the application of Mori-Tanaka’s theory in composite materials,” *Mechanics of Materials*, vol. 6, pp. 145–157, 1987.

[45] J. K. Bauer and T. Böhlke, “On the dependence of orientation averaging mean field homogenization on planar fourth-order fiber orientation tensors,” *Mechanics of Materials*, vol. 170, p. 104307, 2022.

[46] M. Rohde, A. Ebel, F. Wolff-Fabris, and and V. Altstädt, “Influence of processing parameters on the fiber length and impact properties of injection molded long glass fiber reinforced polypropylene,” *International Polymer Processing*, vol. 16, no. 3, pp. 292–303, 2011.

[47] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[48] W.-H. Tsai, “Moment-preserving thresolding: A new approach,” *Computer vision, graphics, and image processing*, vol. 29, no. 3, pp. 377–393, 1985.

[49] C. A. Glasbey, “An analysis of histogram-based thresholding algorithms,” *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 6, pp. 532–537, 1993.

[50] T. W. Ridler and S. Calvard, “Picture thresholding using an iterative selection method,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 8, pp. 630–632, 1978.

[51] H. J. Trussell, “Comments on "picture thresholding using an iterative selection method"," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 5, p. 311, 1979.

[52] P. Pinter, “Microstructure characterization of continuous-discontinuous fibre reinforced polymers based on volumetric images,” Doctoral thesis, Karlsruhe Institute of Technology, Karlsruhe, 2018.

[53] J. S. Rathore, T. Konopczyński, J. Hesser, G. Lucchetta, and S. Carmignato, "Investigation on tomographic-based nondestructive characterization of short glass fiber-reinforced composites as obtained from micro injection molding," *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems*, vol. 3, no. 2, 2020.

[54] J. A. Butenegro, M. Bahrami, Y. Swolfs, J. Ivens, M. Á. Martínez, and J. Abenojar, "Novel thermoplastic composites strengthened with carbon fiber-reinforced epoxy composite waste rods: Development and characterization," *Polymers*, vol. 14, no. 19, 2022.

[55] X.-W. Yu, H. Wang, and Z.-W. Wang, "Analysis of yarn fiber volume fraction in textile composites using scanning electron microscopy and x-ray micro-computed tomography," *Journal of Reinforced Plastics and Composites*, vol. 38, no. 5, pp. 199–210, 2019.

[56] I. Tretiak and R. A. Smith, "A parametric study of segmentation thresholds for x-ray ct porosity characterisation in composite materials," *Composites Part A: Applied Science and Manufacturing*, vol. 123, pp. 10–24, 2019.

[57] P. Galvez-Hernandez, K. Gaska, and J. Kratz, "Phase segmentation of uncured prepreg x-ray ct micrographs," *Composites Part A: Applied Science and Manufacturing*, vol. 149, p. 106527, 2021.

[58] F. Sosa-Rey, Y. Abderrafai, A. Diouf Lewis, D. Therriault, N. Piccirelli, and M. Lévesque, "Openfiberseg: Open-source segmentation of individual fibers and porosity in tomographic scans of additively manufactured short fiber reinforced composites," *Composites Science and Technology*, vol. 226, p. 109497, 2022.

[59] J. R. Ferreira, D. A. C. Cardenas, R. A. Moreno, M. D. F. de Sá Rebelo, J. E. Krieger, and M. A. Gutierrez, "Multi-view ensemble convolutional neural network to improve classification of pneumonia in low contrast chest x-ray images," in *2020 42nd annual international conference of the*

IEEE engineering in Medicine & Biology Society (EMBC). IEEE, 2020, pp. 1238–1241.

[60] Y. S. Moon, B. G. Han, H. S. Yang, and H. G. Lee, "Low contrast image enhancement using convolutional neural network with simple reflection model," *Advances in Science, Technology and Engineering Systems Journal*, vol. 4, no. 1, pp. 159–164, 2019.

[61] R. Kaur, H. GholamHosseini, and R. Sinha, "Hairlines removal and low contrast enhancement of melanoma skin images using convolutional neural network with aggregation of contextual information," *Biomedical Signal Processing and Control*, vol. 76, p. 103653, 2022.

[62] Y. Doi, A. Teramoto, A. Yamada, M. Kobayashi, K. Saito, and H. Fujita, "Estimating subjective evaluation of low-contrast resolution using convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 44, no. 4, pp. 1285–1296, 2021.

[63] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3708–3712.

[64] E. Palovcak, D. Asarnow, M. G. Campbell, Z. Yu, and Y. Cheng, "Enhancing the signal-to-noise ratio and generating contrast for cryo-em images with convolutional neural networks," *IUCrJ*, vol. 7, no. Pt 6, pp. 1142–1150, 2020.

[65] 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*, vol. 56, pp. 1179–1190, 1996.

[66] J. Andersons, R. Joffe, M. Hojo, and S. Ochiai, "Glass fibre strength distribution determined by common experimental methods," *Composites Science and Technology*, vol. 62, pp. 131–145, 2002.

[67] N. Le Moigne, M. den van Oever, and T. Budtova, "A statistical analysis of fibre size and shape distribution after compounding in composites

reinforced by natural fibres," *Composites Part A: Applied Science and Manufacturing*, vol. 42, no. 10, pp. 1542–1550, 2011.

[68] A. Bernasconi, P. DAVOLI, A. BASILE, and A. FILIPPI, "Effect of fibre orientation on the fatigue behaviour of a short glass fibre reinforced polyamide-6," *International Journal of Fatigue*, vol. 29, no. 2, pp. 199–208, 2007.

[69] R. Maertens, A. Hees, L. Schöttl, W. Liebig, P. Elsner, and K. A. Weidenmann, "Fiber shortening during injection molding of glass fiber-reinforced phenolic molding compounds: fiber length measurement method development and validation," *Polymer-Plastics Technology and Materials*, vol. 60, no. 8, pp. 872–885, 2021.

[70] M. R. Hartwich, N. Höhn, H. Mayr, K. Sandau, and R. Stengler, "FASEP ultra-automated analysis of fibre length distribution in glass-fibre-reinforced products," in *Optical Measurement Systems for Industrial Inspection VI*, ser. SPIE Proceedings, P. H. Lehmann, Ed. SPIE, 2009, p. 738921.

[71] D. Salaberger, K. A. Kannappan, J. Kastner, J. Reussner, and T. Auinger, "Evaluation of computed tomography data from fibre reinforced polymers to determine fibre length distribution," *International Polymer Processing*, vol. 26, no. 3, pp. 283–291, 2011.

[72] T. Lee, R. Kashyap, and C. Chu, "Building skeleton models via 3-d medial surface axis thinning algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 462–478, 1994.

[73] F. Pfeifer, J. Kastner, and R. Freytag, "Method for three-dimensional evaluation and visualization of the distribution of fibres in glass-fibre reinforced injection molded parts by μ-x-ray computed tomography," in *17th World Conference on Non-Destructive Testing (WCNDT)*, 2008, pp. 1–7.

[74] M. Teßmann, S. Mohr, S. Gayetskyy, U. Haßler, R. Hanke, and G. Greiner, "Automatic determination of fiber-length distribution in composite material using 3D CT data," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, 2010.

[75] F. Ibrahim, N. A. A. Osman, and J. Usman, "Quantification of collagen orientation in 3d engineered tissue," in *IFMBE proceedings*, vol. 15. Springer-Verlag, 2007.

[76] D. Salaberger, M. Jerabek, T. Koch, and J. Kastner, "Consideration of accuracy of quantitative x-ray ct analyses for short-glass-fibre-reinforced polymers," *Materials Science Forum*, vol. 825-826, pp. 907–913, 2015.

[77] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[78] A. F. Frangi, W. J. Niessen, R. M. Hoogeveen, T. van Walsum, and M. A. Viergever, "Model-based quantitation of 3-d magnetic resonance angiographic images," *IEEE transactions on medical imaging*, vol. 18, no. 10, pp. 946–956, 1999.

[79] B. Bertram and P. Pinter, "Composight: http://sourceforge.net/projects/composight/," 2015, version 0.2.0.

[80] M. Krause, J. M. Hausherr, B. Burgeth, C. Herrmann, and W. Krenkel, "Determination of the fibre orientation in composites using the structure tensor and local X-ray transform," *Journal of Materials Science*, vol. 45, no. 4, pp. 888–896, 2010.

[81] P. Pinter, S. Dietrich, B. Bertram, L. Kehrer, P. Elsner, and K. A. Weidenmann, "Comparison and error estimation of 3D fibre orientation analysis of computed tomography image data for fibre reinforced composites," *NDT & E International*, vol. 95, pp. 26–35, 2018.

[82] B. Jähne, *Digitale Bildverarbeitung: und Bildgewinnung*, 7th ed. Springer-Verlag Berlin, Heidelberg, 2012.

[83] I. Arganda-Carreras, R. Fernández-González, A. Muñoz-Barrutia, and C. Ortiz-De-Solorzano, "3D reconstruction of histological sections: Application to mammary gland tissue," *Microscopy research and technique*, vol. 73, no. 11, pp. 1019–1029, 2010.

[84] M. Doube, M. M. Kłosowski, I. Arganda-Carreras, F. P. Cordelières, R. P. Dougherty, J. S. Jackson, B. Schmid, J. R. Hutchinson, and S. J. Shefelbine, "BoneJ: Free and extensible bone image analysis in ImageJ," *Bone*, vol. 47, no. 6, pp. 1076–1079, 2010.

[85] J. K. Bauer and T. Böhlke, "Variety of fiber orientation tensors," *Mathematics and Mechanics of Solids*, p. 108128652110576, 2021.

[86] N. Meyer, "Mesoscale simulation of the mold filling process of sheet molding compound," Doctoral thesis, Karlsruhe Institute of Technology, Karlsruhe, 2021.

[87] I. Neutelings, "3D coordinate systems," 2021, tikZ.net - Graphics with TikZ in LaTeX, https://tikz.net/axis3d/.

[88] C. L. Tucker III, *Fundamentals of fiber orientation: Description, measurement and prediction*. Carl Hanser Verlag GmbH Co KG, 2022.

[89] K. Kanatani, "Distribution of directional data and fabric tensors," *International Journal of Engineering Science*, vol. 22, no. 2, pp. 149–164, 1984.

[90] B. Brylka, "Charakterisierung und Modellierung der Steifigkeit von langfaserverstärktem Polypropylen," Doctoral thesis, Karlsruher Institut für Technologie, Karlsruhe, 2017.

[91] S. Gajek, M. Schneider, and T. Böhlke, "An FE–DMN method for the multiscale analysis of short fiber reinforced plastic components," *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113952, 2021.

[92] C. Krauß, “Direction-dependent result data transfer in virtual process chains for fiber-reinforced polymers and the impact on structural simulation,” Doctoral thesis, Karlsruhe Institute of Technology, Karlsruhe, 2023.

[93] J. K. Bauer and T. Böhlke, “Fiber orientation distributions based on planar fiber orientation tensors of fourth order,” *Mathematics and Mechanics of Solids*, vol. 28, no. 3, pp. 773–794, 2023.

[94] D. A. Jack and D. E. Smith, “Elastic properties of short-fiber polymer composites, derivation and demonstration of analytical forms for expectation and variance from orientation tensors,” *Journal of Composite Materials*, vol. 42, no. 3, pp. 277–308, 2008.

[95] G. B. Jeffery, “The motion of ellipsoidal particles immersed in a viscous fluid,” *Proceedings of the Royal Society of London. Series A, Containing Papers of aMathematical and Physical Character*, vol. 102, no. 715, pp. 161–179, 1922.

[96] F. Folgar and C. L. Tucker III, “Orientation behavior of fibers in concentrated suspensions,” *Journal of reinforced plastics and composites*, vol. 3, no. 2, pp. 98–119, 1984.

[97] M. Doi, “Molecular dynamics and rheological properties of concentrated solutions of rodlike polymers in isotropic and liquid crystalline phases,” *Journal of Polymer Science: Polymer Physics Edition*, vol. 19, no. 2, pp. 229–243, 1981.

[98] G. L. Hand, “A theory of anisotropic fluids,” *Journal of Fluid Mechanics*, vol. 13, no. 1, p. 33–46, 1962.

[99] S. G. Advani and C. L. Tucker III, “Closure approximations for three-dimensional structure tensors,” *Journal of Rheology*, vol. 34, no. 3, pp. 367–386, 1990.

[100] K.-H. Han and Y.-T. Im, "Modified hybrid closure approximation for prediction of flow-induced fiber orientation," *Journal of Rheology*, vol. 43, no. 3, pp. 569–589, 1999.

[101] J. S. Cintra Jr and C. L. Tucker III, "Orthotropic closure approximations for flow-induced fiber orientation," *Journal of Rheology*, vol. 39, no. 6, pp. 1095–1122, 1995.

[102] D. H. Chung and T. H. Kwon, "Improved model of orthotropic closure approximation for flow induced fiber orientation," *Polymer Composites*, vol. 22, no. 5, pp. 636–649, 2001.

[103] D. H. Chung and T. H. Kwon, "Invariant-based optimal fitting closure approximation for the numerical prediction of flow-induced fiber orientation," *Journal of Rheology*, vol. 46, no. 1, pp. 169–194, 2002.

[104] V. Verleye, "Prediction of fiber orientation in complex injection molded parts," *ASME Appl. Mech. Division*, vol. 175, pp. 139–163, 1993.

[105] K. Breuer, M. Stommel, and W. Korte, "Analysis and evaluation of fiber orientation reconstruction methods," *Journal of Composites Science*, vol. 3, no. 3, p. 67, 2019.

[106] V. Müller and T. Böhlke, "Prediction of effective elastic properties of fiber reinforced composites using fiber orientation tensors," *Composites Science and Technology*, vol. 130, pp. 36–45, 2016.

[107] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[108] C. Lauff, M. Schneider, J. Montesano, and T. Böhlke, "Generating microstructures of long fiber-reinforced composites by the fused sequential addition and migration method," *Submitted*, 2024.

[109] N. Goldberg, F. Ospald, and M. Schneider, "A fiber orientation-adapted integration scheme for computing the hyperelastic tucker average for

short fiber reinforced composites," *Computational Mechanics*, vol. 60, no. 4, pp. 595–611, 2017.

[110] J. Köbler, M. Schneider, F. Ospald, H. Andrä, and R. Müller, "Fiber orientation interpolation for the multiscale analysis of short fiber reinforced composite parts," *Computational Mechanics*, vol. 61, no. 6, pp. 729–750, 2018.

[111] J. Mandel, "Généralisation de la théorie de plasticité de wt koiter," *International Journal of Solids and structures*, vol. 1, no. 3, pp. 273–295, 1965.

[112] C. Pierpaoli and P. J. Basser, "Toward a quantitative assessment of diffusion anisotropy," *Magnetic Resonance in Medicine*, vol. 36, no. 6, pp. 893–906, 1996.

[113] P. J. Basser and C. Pierpaoli, "Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI," *Journal of Magnetic Resonance*, vol. 213, no. 2, pp. 560–570, 2011.

[114] D. H. Laidlaw, E. T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead, "Visualizing diffusion tensor images of the mouse spinal cord," in *Proceedings Visualization '98 (Cat. No.98CB36276)*. IEEE, 1998, pp. 127–134.

[115] F. J. Post, T. van Walsum, F. H. Post, and D. Silver, "Iconic techniques for feature visualization," in *Proceedings Visualization '95*. IEEE Comput. Soc. Press, 1995, pp. 288–295.

[116] G. Kindlmann, "Superquadric tensor glyphs," in *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, 2004, pp. 147–154.

[117] T. Schultz and G. L. Kindlmann, "Superquadric glyphs for symmetric second-order tensors," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 6, pp. 1595–1604, 2010.

[118] A. Barmpoutis, B. C. Vemuri, T. M. Shepherd, and J. R. Forder, "Tensor splines for interpolation and approximation of DT-MRI with applications to segmentation of isolated rat hippocampi," *IEEE transactions on medical imaging*, vol. 26, no. 11, pp. 1537–1546, 2007.

[119] P. Pinter, S. Dietrich, and K. A. Weidenmann, "Algorithms for the determination of orientation-tensors from three dimensional μCT images with various microstructures," in *Conference Proceedings ICCM20, Copenhagen*, 2015.

[120] O. Wirjadi, K. Schladitz, A. Rack, and T. Breuel, "Applications of anisotropic image filters for computing 2d and 3d-fiber orientations," in *Stereology and Image Analysis–10th European Congress of ISS*. Milan, 2009, pp. 107–112.

[121] M. van Ginkel, "Image analysis using orientation space based on steerable filters," Doctoral thesis, Technical University Delft, 2002.

[122] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic resonance in medicine*, vol. 56, no. 2, pp. 411–421, 2006.

[123] J. K. Gahm, N. Wisniewski, G. Kindlmann, G. L. Kung, W. S. Klug, A. Garfinkel, and D. B. Ennis, "Linear invariant tensor interpolation applied to cardiac diffusion tensor MRI," *Medical image computing and computer-assisted intervention: MICCAI (International Conference on Medical Image Computing and Computer-Assisted Intervention)*, vol. 15, no. Pt 2, pp. 494–501, 2012.

[124] D. B. Ennis and G. Kindlmann, "Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images," *Magnetic resonance in medicine*, vol. 55, no. 1, pp. 136–146, 2006.

[125] C. Krauß and L. Kärger, "Tensor interpolation in virtual manufacturing chains for fiber reinforced composites," *International Journal of Mechanical Sciences*, vol. 226, p. 107378, 2022.

[126] F. Hiai and D. Petz, “Riemannian metrics on positive definite matrices related to means,” *Linear Algebra and its Applications*, vol. 430, no. 11-12, pp. 3105–3130, 2009.

[127] J. Blarr, N. Kresin, C. Krauß, K. A. Weidenmann, W. V. Liebig, and P. Elsner, “Application of a tensor interpolation method on the determination of fiber orientation tensors from computed tomography images,” in *Composites Meet Sustainability - Proceedings of the 20th European Conference on Composite Materials*. Anastasios, Vassilopoulos, 2022.

[128] S. Jung, A. Schwartzman, and D. Groisser, “Scaling-Rotation Distance and Interpolation of Symmetric Positive-Definite Matrices,” *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 3, pp. 1180–1201, 2015.

[129] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, “Averaging Quaternions,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1193–1197, 2007.

[130] R. Hartley, J. Trumpf, Y. Dai, and H. Li, “Rotation Averaging,” *International Journal of Computer Vision*, vol. 103, no. 3, pp. 267–305, 2013.

[131] M. Moakher, “Means and Averaging in the Group of Rotations,” *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, pp. 1–16, 2002.

[132] S. Han and O. A. Bauchau, “On the global interpolation of motion,” *Computer Methods in Applied Mechanics and Engineering*, vol. 337, pp. 352–386, 2018.

[133] T. Sabiston, K. Inal, and P. Lee-Sullivan, “Application of Artificial Neural Networks to predict fibre orientation in long fibre compression moulded composite materials,” *Composites Science and Technology*, vol. 190, p. 108034, 2020.

[134] C. Sun and R. Vaidya, "Prediction of composite properties from a representative volume element," *Composites Science and Technology*, vol. 56, no. 2, pp. 171–179, 1996.

[135] S. Bargmann, B. Klusemann, J. Markmann, J. E. Schnabel, K. Schneider, C. Soyarslan, and J. Wilmers, "Generation of 3D representative volume elements for heterogeneous materials: A review," *Progress in Materials Science*, vol. 96, pp. 322–384, 2018.

[136] J. Görthofer, M. Schneider, F. Ospald, A. Hrymak, and T. Böhlke, "Computational homogenization of sheet molding compound composites based on high fidelity representative volume elements," *Computational Materials Science*, vol. 174, p. 109456, 2020.

[137] K. A. Acton and S. C. Baxter, "Characterization of Random Composite Properties Based on Statistical Volume Element Partitioning," *Journal of Engineering Mechanics*, vol. 144, no. 2, 2018.

[138] X. Yin, W. Chen, A. To, C. McVeigh, and W. K. Liu, "Statistical volume element method for predicting microstructure–constitutive property relations," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 43-44, pp. 3516–3529, 2008.

[139] T. Sabiston, K. Inal, and P. Lee-Sullivan, "Method to determine the required microstructure size to be represented by a second order fibre orientation tensor using X-ray micro computed tomography to evaluate compression moulded composites," *Composites Science and Technology*, vol. 182, p. 107777, 2019.

[140] M. Schneider, "The sequential addition and migration method to generate representative volume elements for the homogenization of short fiber reinforced plastics," *Computational Mechanics*, vol. 59, no. 2, pp. 247–263, 2017.

[141] M. Schneider, "An algorithm for generating microstructures of fiber-reinforced composites with long fibers," *International Journal for*

Numerical Methods in Engineering, vol. 123, no. 24, pp. 6197–6219, 2022.

[142] S. Fliegener, M. Luke, and P. Gumbsch, "3d microstructure modeling of long fiber reinforced thermoplastics," *Composites Science and Technology*, vol. 104, pp. 136–145, 2014.

[143] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[144] T. M. Buzug, *Computed tomography: From photon statistics to modern cone-beam CT*. Berlin: Springer, 2008.

[145] G. N. Hounsfield, "Computerized transverse axial scanning (tomography): Part 1. description of system," *The British journal of radiology*, vol. 46, no. 552, pp. 1016–1022, 1973.

[146] Beer, "Bestimmung der Absorption des rothen Lichts in farbigen Flüssigkeiten," *Annalen der Physik*, vol. 162, no. 5, pp. 78–88, 1852.

[147] J. H. Lambert, *Lambert's Photometrie: (Photometria, sive De mensura et gradibus luminis, colorum et umbrae) (1760)*. Wilhelm Engelmann, Leipzig, 1892.

[148] D. F. Swinehart, "The Beer-Lambert law," *Journal of Chemical Education*, vol. 39, no. 7, p. 333, 1962.

[149] R. Paris, "Chapter 17 - x-ray tomography applied to tsunami deposits," in *Geological Records of Tsunamis and Other Extreme Waves*, M. Engel, J. Pilarczyk, S. M. May, D. Brill, and E. Garrett, Eds. Elsevier, 2020, pp. 365–380.

[150] J. H. Hubbell and S. M. Seltzer, "Tables of x-ray mass attenuation coefficients and mass energy-absorption coefficients 1 kev to 20 mev for elements z= 1 to 92 and 48 additional substances of dosimetric interest,"

National Inst. of Standards and Technology-PL, Gaithersburg, MD (United States), Tech. Rep., 1995.

[151] I. Han, L. Demir, and M. Şahin, "Determination of mass attenuation coefficients, effective atomic and electron numbers for some natural minerals," *Radiation Physics and Chemistry*, vol. 78, no. 9, pp. 760–764, 2009.

[152] I. M. Low and N. Z. N. Azman, *Polymer Composites and Nanocomposites for x-rays shielding*. Springer, Composites Science and Technology (CST) book series, 2020.

[153] G. T. Herman, *Fundamentals of computerized tomography: image reconstruction from projections*. Springer Science & Business Media, 2009.

[154] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," *Journal of the Optical Society of America A*, vol. 1, no. 6, pp. 612–619, 1984.

[155] J. J. Lifton, "Evaluating the standard uncertainty due to the voxel size in dimensional computed tomography," *Precision Engineering*, vol. 79, pp. 245–250, 2023.

[156] C. V. More, R. R. Bhosale, and P. P. Pawar, "Detection of new polymer materials as gamma-ray-shielding materials," *Radiation Effects and Defects in Solids*, vol. 172, no. 5-6, pp. 469–484, 2017.

[157] P. A. Hessman, T. Riedel, F. Welschinger, K. Hornberger, and T. Böhlke, "Microstructural analysis of short glass fiber reinforced thermoplastics based on x-ray micro-computed tomography," *Composites Science and Technology*, vol. 183, p. 107752, 2019.

[158] J. Kastner, C. Heinzl, B. Plank, D. Salaberger, C. Gusenbauer, and S. Senck, "New x-ray computed tomography methods for research and industry," in *7th Conference on Industrial Computed Tomography (iCT2017). ICT, Leuven, Belgium. http://www. ndt. net*, 2017.

[159] P. Sprawls, “Aapm tutorial. ct image detail and noise,” *Radiographics : a review publication of the Radiological Society of North America, Inc*, vol. 12, no. 5, pp. 1041–1046, 1992.

[160] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 4th ed. New York, NY: Pearson, 2018.

[161] M. Hashemi, “Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation,” *Journal of Big Data*, vol. 6, no. 1, 2019.

[162] G. Liu, A. Dundar, K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, X. Yang, A. Tao, and B. Catanzaro, “Partial convolution for padding, inpainting, and image synthesis,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 5, pp. 6096–6110, 2023.

[163] M. Xu, C. Li, S. Zhang, and P. Le Callet, “State-of-the-art in 360° video/image processing: Perception, assessment and compression,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 5–26, 2020.

[164] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 839–846.

[165] D. Liu and J. Yu, “Otsu method and k-means,” in *2009 Ninth International Conference on Hybrid Intelligent Systems*, J.-S. Pan, Ed. Piscataway, NJ: IEEE, 2009, pp. 344–349.

[166] W. Doyle, “Operations useful for similarity-invariant pattern recognition,” *Journal of the ACM (JACM)*, vol. 9, no. 2, pp. 259–267, 1962.

[167] A. Turing and B. J. Copeland, Eds., *The essential Turing: Seminal writings in computing, logic, philosophy, artificial intelligence, and artificial life, plus the secrets of Enigma*. Oxford and New York: Clarendon Press, 2010.

[168] Y. Hong, B. Hou, H. Jiang, and J. Zhang, "Machine learning and artificial neural network accelerated computational discoveries in materials science," *WIREs Computational Molecular Science*, vol. 10, no. 3, 2020.

[169] J. Manyika and J. Bughin, "The coming of ai spring," *McKinsey Global Institute*, 2019.

[170] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, London, 2016.

[171] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain [j]," *Psychological review*, vol. 65, no. 6, pp. 386–408, 1958.

[172] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[173] D. Hammerstrom, "Working with neural networks," *IEEE Spectrum*, vol. 30, no. 7, pp. 46–53, 1993.

[174] J. Singh and R. Banerjee, "A study on single and multi-layer perceptron neural network," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 32019, pp. 35–40.

[175] B. Cheng and D. M. Titterington, "Neural Networks: A Review from a Statistical Perspective," *Statistical Science*, vol. 9, no. 1, pp. 2 – 30, 1994.

[176] C. Shao, "A quantum model for multilayer perceptron," *arXiv preprint arXiv:1808.10561*, 2018.

[177] B. J. Wythoff, "Backpropagation neural networks: A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 2, pp. 115–155, 1993.

[178] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[179] J. Cheng, Y. Yang, X. Tang, N. Xiong, Y. Zhang, and F. Lei, "Generative adversarial networks: A literature review," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 12, pp. 4625–4647, 2020.

[180] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[181] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press Inc., New York, 1995.

[182] M. T. Hagan, H. B. Demuth, and O. de Jesús, "An introduction to the use of neural networks in control systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 11, pp. 959–985, 2002.

[183] K. Atanassov and S. Sotirov, "Index matrix interpretation of the multilayer perceptron," in *2013 IEEE INISTA*, 2013, pp. 1–3.

[184] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[185] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.

[186] J. Lever, M. Krzywinski, and N. Altman, "Model selection and overfitting," *Nature Methods*, vol. 13, no. 9, pp. 703–704, 2016.

[187] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.

[188] J. Blarr, P. Kunze, N. Kresin, W. V. Liebig, K. Inal, and K. A. Weidenmann, "Novel thresholding method and convolutional neural network for fiber volume content determination from 3D µCT images," *NDT & E International*, vol. 144, p. 103067, 2024.

[189] O. H. Rodriguez and J. M. Lopez Fernandez, "A semiotic reflection on the didactics of the chain rule," *The Mathematics Enthusiast*, vol. 7, no. 2-3, pp. 321–332, 2010.

[190] S. Linnainmaa, "Taylor expansion of the accumulated rounding error," *BIT*, vol. 16, no. 2, pp. 146–160, 1976.

[191] P. J. Werbos, "Applications of advances in nonlinear sensitivity analysis," in *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31–September 4, 1981*. Springer, 2005, pp. 762–770.

[192] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Neurocomputing, Volume 1: Foundations of Research*. The MIT Press, Cambridge, MA, 1988.

[193] R. Rojas, *Neural networks: A systematic introduction*. Berlin and Heidelberg: Springer, 1996.

[194] J. Leonard and M. Kramer, "Improvement of the backpropagation algorithm for training neural networks," *Computers & Chemical Engineering*, vol. 14, no. 3, pp. 337–341, 1990.

[195] W. Jin, Z. J. Li, L. S. Wei, and H. Zhen, "The improvements of bp neural network learning algorithm," in *WCC 2000 - ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000*, vol. 3, 2000, pp. 1647–1649 vol.3.

[196] A. Cauchy, "Méthode générale pour la résolution des systemes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, no. 1847, pp. 536–538, 1847.

[197] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM review*, vol. 60, no. 2, pp. 223–311, 2018.

[198] S. Ruder, “An overview of gradient descent optimization algorithms,” https://arxiv.org/abs/1609.04747, 2016.

[199] H. E. Robbins, “A stochastic approximation method,” *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[200] J. Kiefer and J. Wolfowitz, “Stochastic estimation of the maximum of a regression function,” *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

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

[202] S. Bock and M. Weiß, “A proof of local convergence for the adam optimizer,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[203] C. Gulcehre, M. Moczulski, M. Denil, and Y. Bengio, “Noisy activation functions,” in *International conference on machine learning*. PMLR, 2016, pp. 3059–3068.

[204] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, “Rectifier nonlinearities improve neural network acoustic models,” in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, vol. 30, no. 1. Atlanta, GA, 2013, p. 3.

[205] J. Lederer, “Activation functions in artificial neural networks: A systematic overview,” *arXiv preprint arXiv:2101.09957*, 2021.

[206] B. Ding, H. Qian, and J. Zhou, “Activation functions and their characteristics in deep neural networks,” in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 1836–1841.

[207] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, “A review of activation function for artificial neural network,” in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2020, pp. 281–286.

[208] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[209] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[210] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[211] A. Livnat, C. Papadimitriou, N. Pippenger, and M. W. Feldman, "Sex, mixability, and modularity," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 4, pp. 1452–1457, 2010.

[212] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[213] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456.

[214] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimedia Tools and Applications*, vol. 79, no. 19-20, pp. 12 777–12 815, 2020.

[215] S. B. Damelin and W. Miller, *The mathematics of signal processing*. Cambridge University Press, Cambridge, UK, 2012, no. 48.

[216] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation

network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. Morgan-Kaufmann, 1989.

[217] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[218] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.

[219] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[220] J. Bouvrie, "Notes on convolutional neural networks," *Massachusetts Institute of Technology*, 2006.

[221] J. Koushik, "Understanding convolutional neural networks," *arXiv preprint arXiv:1605.09081*, 2016.

[222] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3313–3332, 2023.

[223] G. Scholar, "Google scholar citation statistics of Ian J. Goodfellow," https: //scholar.google.de/citations?user=iYN86KEAAAAJ&hl=de&oi=ao, 2024, accessed: 2024-06-24.

[224] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100004, 2021.

[225] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, and J. Hu, "Generative adversarial networks (gan) based efficient sampling of chemical composition space

for inverse design of inorganic materials," *npj Computational Materials*, vol. 6, no. 1, 2020.

[226] Y. Mao, Q. He, and X. Zhao, "Designing complex architectured materials with generative adversarial networks," *Science Advances*, vol. 6, no. 17, 2020.

[227] B. Ma, X. Wei, C. Liu, X. Ban, H. Huang, H. Wang, W. Xue, S. Wu, M. Gao, Q. Shen, M. Mukeshimana, A. O. Abuassba, H. Shen, and Y. Su, "Data augmentation in microscopic images for material data mining," *npj Computational Materials*, vol. 6, no. 1, 2020.

[228] J.-W. Lee, N. H. Goo, W. B. Park, M. Pyo, and K.-S. Sohn, "Virtual microstructure design for steels using generative adversarial networks," *Engineering Reports*, vol. 3, no. 1, 2021.

[229] A. Marani, A. Jamali, and M. L. Nehdi, "Predicting ultra-high-performance concrete compressive strength using tabular generative adversarial networks," *Materials*, vol. 13, no. 21, 2020.

[230] D. Fokina, E. Muravleva, G. Ovchinnikov, and I. Oseledets, "Microstructure synthesis using style-based generative adversarial networks," *Phys. Rev. E*, vol. 101, p. 043308, 2020.

[231] R. Karamov, S. V. Lomov, I. Sergeichev, Y. Swolfs, and I. Akhatov, "Inpainting micro-ct images of fibrous materials using deep learning," *Computational Materials Science*, vol. 197, p. 110551, 2021.

[232] R. Karamov, C. Breite, S. V. Lomov, I. Sergeichev, and Y. Swolfs, "Super-resolution processing of synchrotron ct images for automated fibre break analysis of unidirectional composites," *Polymers*, vol. 15, no. 9, 2023.

[233] S. Kench and S. J. Cooper, "Generating three-dimensional structures from a two-dimensional slice with generative adversarial network-based dimensionality expansion," *Nature Machine Intelligence*, vol. 3, no. 4, pp. 299–305, 2021.

[234] R. Guo, M. Alves, M. Mehdikhani, C. Breite, and Y. Swolfs, "Generation of realistic 2D transverse microstructures of unidirectional fiber-reinforced composites using a generative adversarial network," in *Proceedings of the 20th European Conference on Composite Materials-Composites Meet Sustainability (Vol 1-6)*, vol. 4. EPFL Lausanne, Composite Construction Laboratory; Lausanne, Switzerland, 2022, pp. 827–834.

[235] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[236] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[237] L. Gonog and Y. Zhou, "A review: Generative adversarial networks," in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2019, pp. 505–510.

[238] D. Saxena and J. Cao, "Generative adversarial networks (GANs): Challenges, solutions, and future directions," *ACM Computing Surveys*, vol. 54, no. 3, 2021.

[239] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[240] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997.

[241] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, and P. L'Ecuyer, "Chapter 3 - the cross-entropy method for optimization," in *Handbook of Statistics*, ser. Handbook of Statistics, C. Rao and V. Govindaraju, Eds. Elsevier, 2013, vol. 31, pp. 35–59.

[242] A. Yadav, S. Shah, Z. Xu, D. Jacobs, and T. Goldstein, "Stabilizing adversarial nets with prediction methods," *arXiv preprint arXiv:1705.07364*, 2017.

[243] A. Sajeeda and B. M. M. Hossain, "Exploring generative adversarial networks and adversarial training," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 78–89, 2022.

[244] Z. Ding, S. Jiang, and J. Zhao, "Take a close look at mode collapse and vanishing gradient in GAN," in *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*, 2022, pp. 597–602.

[245] K. S. Lee, N.-T. Tran, and N.-M. Cheung, "InfoMax-GAN: Improved adversarial image generation via information maximization and contrastive learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 3942–3952.

[246] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," *arXiv preprint arXiv:1705.07215*, 2017.

[247] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in gans," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–10.

[248] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[249] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[250] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[251] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[252] L. Weng, "From GAN to WGAN," *arXiv preprint arXiv:1904.08994*, 2019, https://arxiv.org/abs/1904.08994.

[253] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[254] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[255] X. Ding, Y. Wang, Z. Xu, W. J. Welch, and Z. J. Wang, "CcGAN: Continuous conditional generative adversarial networks for image generation," in *International Conference on Learning Representations*, 2020, https://openreview.net/forum?id=PrzjugOsDeE.

[256] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.*, 2016.

[257] C. Herriott and A. D. Spear, "Predicting microstructure-dependent mechanical properties in additively manufactured metals with machine- and deep-learning methods," *Computational Materials Science*, vol. 175, p. 109599, 2020.

[258] L. Yin, J. Wang, H. Hu, S. Han, and Y. Zhang, "Prediction of weld formation in 5083 aluminum alloy by twin-wire CMT welding based on deep learning," *Welding in the World*, vol. 63, no. 4, pp. 947–955, 2019.

[259] Y.-C. Hsu, C.-H. Yu, and M. J. Buehler, "Using Deep Learning to Predict Fracture Patterns in Crystalline Solids," *Matter*, vol. 3, no. 1, pp. 197–211, 2020.

[260] K. Ryan, J. Lengyel, and M. Shatruk, "Crystal Structure Prediction via Deep Learning," *Journal of the American Chemical Society*, vol. 140, no. 32, pp. 10 158–10 168, 2018.

[261] A. R. Kitahara and E. A. Holm, "Microstructure Cluster Analysis with Transfer Learning and Unsupervised Learning," *Integrating Materials and Manufacturing Innovation*, vol. 7, no. 3, pp. 148–156, 2018.

[262] M. X. Bastidas-Rodriguez, L. Polania, A. Gruson, and F. Prieto-Ortiz, "Deep Learning for fractographic classification in metallic materials," *Engineering Failure Analysis*, vol. 113, p. 104532, 2020.

[263] E. A. Holm, R. Cohn, N. Gao, A. R. Kitahara, T. P. Matson, B. Lei, and S. R. Yarasi, "Overview: Computer Vision and Machine Learning for Microstructural Characterization and Analysis," *Metallurgical and Materials Transactions A*, vol. 51, no. 12, pp. 5985–5999, 2020.

[264] S. M. Azimi, D. Britz, M. Engstler, M. Fritz, and F. Mücklich, "Advanced Steel Microstructural Classification by Deep Learning Methods," *Scientific reports*, vol. 8, no. 1, p. 2128, 2018.

[265] B. L. DeCost, B. Lei, T. Francis, and E. A. Holm, "High Throughput Quantitative Metallography for Complex Microstructures Using Deep Learning: A Case Study in Ultrahigh Carbon Steel," *Microscopy and microanalysis : the official journal of Microscopy Society of America, Microbeam Analysis Society, Microscopical Society of Canada*, vol. 25, no. 1, pp. 21–29, 2019.

[266] K. Pazdernik, N. L. LaHaye, C. M. Artman, and Y. Zhu, "Microstructural classification of unirradiated LiAlO2 pellets by deep learning methods," *Computational Materials Science*, vol. 181, p. 109728, 2020.

[267] C. Shashank Kaira, X. Yang, V. de Andrade, F. de Carlo, W. Scullin, D. Gursoy, and N. Chawla, "Automated correlative segmentation of large Transmission X-ray Microscopy (TXM) tomograms using deep learning," *Materials Characterization*, vol. 142, pp. 203–210, 2018.

[268] M. Ziatdinov, O. Dyck, A. Maksov, B. M. Hudak, A. R. Lupini, J. Song, P. C. Snijders, R. K. Vasudevan, S. Jesse, and S. V. Kalinin, "Deep analytics of atomically-resolved images: manifest and latent features," *arXiv preprint arXiv:1801.05133*, 2018.

[269] Q. Qiao, J. Wang, L. Ye, and R. X. Gao, "Digital Twin for Machining Tool Condition Prediction," *Procedia CIRP*, vol. 81, pp. 1388–1393, 2019.

[270] K. de Haan, Z. S. Ballard, Y. Rivenson, Y. Wu, and A. Ozcan, "Resolution enhancement in scanning electron microscopy using deep learning," *Scientific reports*, vol. 9, no. 1, p. 12050, 2019.

[271] W. Kaiser, *Kunststoffchemie für Ingenieure: Von der Synthese bis zur Anwendung*, 3rd ed. München: Hanser, 2011.

[272] P. Eyerer, T. Hirth, and P. Elsner, *Polymer Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[273] Zoltek, "How is carbon fiber made?" *https://zoltek.com/carbon-fiber/how-is-carbon-fiber-made/*, 2024.

[274] B. M. Scheuring, N. Christ, J. Blarr, W. V. Liebig, J. Hohe, J. Montesano, and K. A. Weidenmann, "Experimental and homogenized orientation-dependent properties of hybrid long fiber-reinforced thermoplastics," *International Journal of Mechanical Sciences*, p. 109470, 2024.

[275] J. Blarr, S. Klinder, W. V. Liebig, K. Inal, L. Kärger, and K. A. Weidenmann, "Deep convolutional generative adversarial network for generation of computed tomography images of discontinuously carbon

fiber reinforced polymer microstructures," *Nature Scientific Reports*, vol. 14, no. 1, 2024.

[276] L. Schreyer, J. Blarr, K. Höger, N. Meyer, and L. Kärger, "Generation of initial fiber orientation states for long fiber reinforced thermoplastic compression molding simulation," in *Composites Meet Sustainability - Proceedings of the 20th European Conference on Composite Materials*. Anastasios, Vassilopoulos, 2022.

[277] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[278] G. Bradski, "The opencv library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[279] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[280] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[281] J. Serra, *Image analysis and mathematical morphology*. Academic Press, Inc., 1983.

[282] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[283] M. J. Emerson, K. M. Jespersen, A. B. Dahl, K. Conradsen, and L. P. Mikkelsen, "Individual fibre segmentation from 3d x-ray computed tomography for characterising the fibre orientation in unidirectional

composite materials," *Composites Part A: Applied Science and Manufacturing*, vol. 97, pp. 83–92, 2017.

[284] R. Beare, B. Lowekamp, and Z. Yaniv, "Image segmentation, registration and characterization in r with simpleitk," *Journal of statistical software*, vol. 86, 2018.

[285] Z. Yaniv, B. C. Lowekamp, H. J. Johnson, and R. Beare, "Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research," *Journal of digital imaging*, vol. 31, no. 3, pp. 290–303, 2018.

[286] B. C. Lowekamp, D. T. Chen, L. Ibáñez, and D. Blezek, "The design of simpleitk," *Frontiers in neuroinformatics*, vol. 7, p. 45, 2013.

[287] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," *Software available from tensorflow.org*, 2015.

[288] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[289] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proceedings of the IEEE international conference on computer vision workshops*, 2017, pp. 887–895.

[290] H. Zunair, A. Rahman, N. Mohammed, and J. P. Cohen, "Uniformizing techniques to process ct scans with 3d cnns for tuberculosis prediction," in *Predictive Intelligence in Medicine: Third International Workshop,*

PRIME 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 3. Springer, 2020, pp. 156–168.

[291] Haris Iqbal, "Harisiqbal88/plotneuralnet v1.0.0," *Zenodo*, 2018.

[292] J. Brownlee, *Probability for machine learning: Discover how to harness uncertainty with Python*. Machine Learning Mastery, 2019.

[293] M. M. Noel, A. Banerjee *et al.*, "Alternate loss functions can improve the performance of artificial neural networks," *arXiv preprint arXiv:2303.09935*, 2023.

[294] F. Chollet *et al.*, "Building autoencoders in keras," *The Keras Blog*, vol. 14, 2016.

[295] A. Creswell, K. Arulkumaran, and A. A. Bharath, "On denoising autoencoders trained to minimise binary cross-entropy," *arXiv preprint arXiv:1708.08487*, 2017.

[296] D.-J. Kroon, "Read medical data 3D," https://www.mathworks.com/matlabcentral/fileexchange/29344-read-medical-data-3d, 2024, accessed: 2024-06-24 on MATLAB Central File Exchange.

[297] D. Shepard, "A two-dimensional interpolation for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, 1968, pp. 517–524.

[298] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256.

[299] "BwUniCluster2.0/Hardware and Architecture - bwHPC Wiki," https://wiki.bwhpc.de/e/BwUniCluster2.0/Hardware_and_Architecture, 2024, accessed: 2024-06-24.

[300] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012.

[301] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[302] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1485–1488.

[303] N. S. Detlefsen, J. Borovec, J. Schock, A. H. Jha, T. Koker, L. D. Liello, D. Stancl, C. Quan, M. Grechkin, and W. Falcon, "Torchmetrics - measuring reproducibility in pytorch," *Journal of Open Source Software*, vol. 7, no. 70, p. 4101, 2022.

[304] T. Yep, "torchinfo," *https://github.com/TylerYep/torchinfo*, 2020.

[305] F. Perez and B. E. Granger, "Ipython: A system for interactive scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 21–29, 2007.

[306] C. O. da Costa-Luis, "'tqdm': A fast, extensible progress meter for python and cli," *Journal of Open Source Software*, vol. 4, no. 37, p. 1277, 2019.

[307] L. P. Coelho, "Mahotas: Open source software for scriptable computer vision," *Journal of Open Research Software*, Jul 2013.

[308] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *Advances*

in neural information processing systems, vol. 33, pp. 12 104–12 114, 2020.

[309] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, "Image augmentations for GAN training," *arXiv preprint arXiv:2006.02595*, 2020.

[310] N. Inkawhich, "DCGAN tutorial," *https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html*, 2023.

[311] M. Hasani, "Lung DCGAN 128 $\times$ 128," *Kaggle*, 2021.

[312] A. Borji, "Pros and cons of GAN evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[313] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[314] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *Advances in neural information processing systems*, vol. 29, 2016.

[315] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

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

[317] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my GAN?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 213–229.

[318] G. Parmar, R. Zhang, and J. Zhu, "On aliased resizing and surprising subtleties in GAN evaluation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2022, pp. 11 400–11 410.

[319] L. Wang, Y. Zhang, and J. Feng, "On the Euclidean distance of images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1334–1339, 2005.

[320] S. Tröster, "Materialentwicklung und-charakterisierung für thermoplastische faserverbundwerkstoffe im direktverfahren," Doctoral thesis, Fraunhofer-Institut für Chemische Technologie ICT, 2004.

[321] A. Radtke, "Steifigkeitsberechnung von diskontinuierlich faserverstärkten thermoplasten auf der basis von faserorientierungs-und faserlängenverteilungen," Doctoral thesis, Universität Stuttgart, 2009.

[322] M. McLeod, É. Baril, J. Hétu, T. Deaville, and M. Bureau, "Morphological and mechanical comparision of injection and compression moulding in-line compounding of direct long fibre thermoplastics," *Proceedings of the ACCE*, pp. 109–118, 2010.

[323] G. Cao, Y. Yang, J. Lei, C. Jin, Y. Liu, and M. Song, "Tripletgan: Training generative model with triplet loss," *arXiv preprint arXiv:1711.05084*, 2017.

[324] E. Barsoum, J. Kender, and Z. Liu, "HP-GAN: Probabilistic 3D human motion prediction via gan," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1418–1427.

[325] J. McGinn, C. Messenger, I. S. Heng, and M. J. Williams, "Generalised gravitational burst generation with generative adversarial networks," *Classical and Quantum Gravity*, vol. 38, no. 15, p. 155005, 2021.

[326] S. Kora Venu and S. Ravula, "Evaluation of deep convolutional generative adversarial networks for data augmentation of chest x-ray images," *Future Internet*, vol. 13, no. 1, p. 8, 2021.

[327] S. Löwe, "Charakterisierung des Plastifikates an der Schnittstelle zwischen Compounding und Fließpressen langfaserverstärkter thermoplastischer Formmassen," Bachelor's thesis, Karlsruhe Institute of Technology, Karlsruhe, 2023.

[328] D. Bücheler, A. Kaiser, and F. Henning, "Using thermogravimetric analysis to determine carbon fiber weight percentage of fiber-reinforced plastics," *Composites Part B: Engineering*, vol. 106, pp. 218–223, 2016.

[329] M. Mohsin, L. Iannucci, and E. Greenhalgh, "Fibre-volume-fraction measurement of carbon fibre reinforced thermoplastic composites using thermogravimetric analysis," *Heliyon*, vol. 5, no. 1, 2019.

[330] B. Wintiba, D. Vasiukov, S. Panier, S. V. Lomov, K. Ehab Moustafa Kamel, and T. J. Massart, "Automated reconstruction and conformal discretization of 3d woven composite ct scans with local fiber volume fraction control," *Composite Structures*, vol. 248, p. 112438, 2020.

[331] S. Goris and T. Osswald, "Process-induced fiber matrix separation in long fiber-reinforced thermoplastics," *Composites Part A: Applied Science and Manufacturing*, vol. 105, pp. 321–333, 2018.

[332] P. Weinberger, B. Plank, B. Fröhler, J. Kastner, and C. Heinzl, "Segmentation of multiple features in carbon fiber reinforced polymers using a convolutional neural network," in *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XVI*, H. F. Wu, A. L. Gyekenyesi, P. J. Shull, and T. Yu, Eds., vol. 12047, International Society for Optics and Photonics. SPIE, 2022, p. 120470Z.

[333] P. Bleiziffer, J. Hofmann, R. Zboray, T. Wiege, and R. Herger, "Predicting the fiber orientation in glass fiber reinforced polymers using

the moment of inertia and convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104351, 2021.

[334] R. Helwing, D. Hülsbusch, and F. Walther, "Deep learning method for analysis and segmentation of fatigue damage in x-ray computed tomography data for fiber-reinforced polymers," *Composites Science and Technology*, vol. 230, p. 109781, 2022.

[335] J. H. Jeon, E. Chemali, S. S. Yang, and Y. J. Kang, "Convolutional neural networks for estimating transport parameters of fibrous materials based on micro-computerized tomography images," *The Journal of the Acoustical Society of America*, vol. 149, no. 4, pp. 2813–2828, 04 2021.

[336] M. Schemme, "Gmt/lft-materialien und anwendungen im fahrzeugbau," *SKZFachtagung Karosseriekonzepte mit Kunststoffen*, 2002.

[337] J. Thomason and M. Vlug, "Influence of fibre length and concentration on the properties of glass fibre-reinforced polypropylene: 1. tensile and flexural modulus," *Composites Part A: Applied Science and Manufacturing*, vol. 27, no. 6, pp. 477–484, 1996.

[338] J. K. Bauer, C. Krauß, J. Blarr, P. L. Kinon, L. Kärger, and T. Böhlke, "Evaluation of a decomposition-based interpolation method for fourth-order fiber-orientation tensors: An eigensystem approach," *Mathematics and Mechanics of Solids*, p. 10812865241241002, 2024.

[339] P. Siriphala, "Controlling artificial neural networks overtraining when data is scarce," PhD thesis, Wichita State University, 2000.

[340] Y.-Y. Tsai, P.-Y. Chen, and T.-Y. Ho, "Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 9614–9624.

[341] H. L. Cheung, P. Uvdal, and M. Mirkhalaf, "Augmentation of scarce data—a new approach for deep-learning modeling of composites," *Composites Science and Technology*, vol. 249, p. 110491, 2024.

[342] A. de Wynter, "An algorithm for learning smaller representations of models with scarce data," *arXiv*, 2020.

[343] R. M. Brannon, *Rotation, Reflection, and Frame Changes*, ser. 2053-2563. IOP Publishing, 2018.

[344] N. Christ, B. M. Scheuring, J. Montesano, and J. Hohe, "A python package for homogenization procedures in fiber reinforced polymers," *Journal of Open Source Software*, vol. 8, no. 87, p. 5295, 2023.

[345] G. Lambard, K. Yamazaki, and M. Demura, "Generation of highly realistic microstructural images of alloys from limited data with a style-based generative adversarial network," *Scientific reports*, vol. 13, no. 1, p. 566, 2023.

[346] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[347] J. Blarr, "3D μCT images of specimens of carbon fiber reinforced polyamide 6 plaque, fiber orientation tensor data of these images, and three python code files for two different algebraic and one machine learning based tensor interpolation algorithms," *KIT Open*, 2022, dataset, DOI: 10.5445/IR/1000153725 .

[348] J. Blarr, "Computed tomography scans of 15 different carbon fiber reinforced polyamide 6 specimens (both original files (16 bit) and files of further cut scans (8 bit))," *KIT Open*, 2023, dataset, DOI: 10.35097/1707.

[349] J. Blarr and S. Klinder, "2D images of CT scans of carbon fiber reinforced polyamide 6," *KIT Open*, 2023, dataset, DOI: 10.35097/1822 .

[350] L. Schreyer, B. M. Scheuring, N. Christ, J. Blarr, C. Krauß, W. V. Liebig, K. A. Weidenmann, T. Böhlke, A. Hrymak, and L. Kärger, "Continuous simulation of a continuous-discontinuous fiber reinforced thermoplastic (CoDiCoFRTP) compression molding process," in *Proceedings of the twenty-third international conference on composite materials (ICCM23)*. Queen's University Belfast, 2023.

[351] N. Afrasiabian, A. Elmoghazy, J. Blarr, B. Scheuring, A. Prahs, D. Schneider, W. V. Liebig, K. A. Weidenmann, C. Denniston, and B. Nestler, "Crystallization and crystal morphology of polymers: A multiphase-field study," *Journal of Thermoplastic Composite Materials*, p. 08927057241296472, 2024.

[352] L. Kehrer, B. Scheuring, J. Blarr, and T. Böhlke, "Hydrothermal behavior of pure PA6 and homogenization of discontinuous long carbon fiber-reinforced PA6," *Proceedings in Applied Mathematics and Mechanics*, p. e202400135, 2024.

Appendices

A FVC evaluations

A.1 Results of FVC determined by different conventional thresholding methods

Table A.1: Fiber volume contents of the specimens of the first plaque determined by chemical dissolution and by exemplary slice-wise conventional thresholding procedures in Python that are also available in ImageJ. The results of the white row of FLD_{10} were not incorporated into the final maximum deviations, sum of relative deviations and relative average deviations in order to be comparable to the results in Appendix B. Extracted from Blarr et al. [188].

Specimen	**Exp. values**	**opencv Otsu**		**medianBlur(...,15) + O.**		**skimage mean**		**medianBlur(...,15) + m.**	
	FVC	FVC	Deviation	FVC	Deviation	FVC	Deviation	FVC	Deviation
FLD_1	22.30 %	48.90 %	119.28 %	33.18 %	48.79 %	49.58 %	122.33 %	44.10 %	97.76 %
FLD_2	25.50 %	48.89 %	91.73 %	28.20 %	10.59 %	49.66 %	94.75 %	43.50 %	70.59 %
FLD_3	28.60 %	48.93 %	71.08 %	30.89 %	8.01 %	49.69 %	73.74 %	43.55 %	52.27 %
FLD_{10}	17.90 %	49.31 %	175.47 %	44.36 %	147.82 %	49.77 %	178.04 %	47.88 %	167.49 %
FLD_{11}	24 %	48.69 %	102.88 %	32.49 %	35.38 %	49.49 %	106.21 %	43.74 %	82.25 %
FLD_{12}	26.60 %	48.88 %	83.76 %	31.45 %	18.23 %	49.59 %	86.43 %	42.51 %	59.81 %
Max. deviation			119.28 %		48.79 %		122.33 %		97.76 %
Sum of rel. dev.			468.73 %		120.99 %		483.46 %		362.68 %
Rel. aver. dev.			**93.75 %**		**24.20 %**		**96.69 %**		**72.54 %**

Table A.2: Fiber volume contents of the specimens of the seecond plaque determined by chemical dissolution and by exemplary slice-wise conventional thresholding procedures in Python that are also available in ImageJ. Extracted from Blarr et al. [188].

Specimen	**Exp. values**	**opencv Otsu**		**medianBlur(...,23) + O.**		**skimage mean**		**medianBlur(...,23) + m.**	
	FVC	FVC	Deviation	FVC	Deviation	FVC	Deviation	FVC	Deviation
C_1	23.07 %	47.97 %	107.96 %	30.88 %	33.87 %	49.30 %	113.73 %	44.69 %	93.74 %
C_2	22.08 %	48.78 %	120.91 %	46.33 %	109.82 %	49.47 %	124.04 %	49.00 %	121.91 %
C_3	23.06 %	48.10 %	108.60 %	33.84 %	46.75 %	49.32 %	113.89 %	45.71 %	98.23 %
CF_1	25.57 %	48.32 %	88.94 %	33.25 %	30.01 %	49.37 %	93.05 %	44.16 %	72.68 %
CF_2	22.31 %	47.44 %	112.60 %	36.85 %	65.14 %	49.06 %	119.86 %	45.64 %	104.54 %
CF_3	22.81 %	47.77 %	109.42 %	32.94 %	44.40 %	49.16 %	115.51 %	46.13 %	102.23 %
F_1	26.36 %	48.27 %	83.14 %	39.78 %	50.93 %	49.47 %	87.69 %	46.88 %	77.87 %
F_2	23.10 %	48.98 %	112.01 %	46.58 %	101.62 %	49.71 %	115.17 %	49.00 %	112.10 %
F_3	23.81 %	48.22 %	102.55 %	36.07 %	51.51 %	49.34 %	107.25 %	46.43 %	95.03 %
Max. deviation			120.91 %		109.82 %		124.04 %		121.91 %
Sum of rel. dev.			946.13 %		534.07 %		990.19 %		878.31 %
Rel. aver. dev.			**105.13 %**		**59.34 %**		**110.02 %**		**97.59 %**

A.2 Results of FVC of Plaque 2 determined by AOA thresholding for different pre-filters

Table A.3: Fiber volume contents determined by chemical dissolution and by the novel thresholding technique when applying different configurations of the "blur" filter onto the CT images. Extracted from Blarr et al. [188].

Specimen	**Exp. values**	**blur(...,(11,11))**		**blur(...,(13,13))**		**blur(...,(15,15))**		**blur(...,(19,19))**	
	FVC	FVC	Deviation	FVC	Deviation	FVC	Deviation	FVC	Deviation
FLD_1	22.30 %	25.43 %	14.04 %	23.66 %	6.10 %	22.41 %	0.49 %	21.05 %	-5.61 %
FLD_2	25.50 %	26.77 %	4.98 %	25 %	-1.96 %	23.76 %	-6.82 %	21.26 %	-16.63 %
FLD_3	28.60 %	29.83 %	4.30 %	28.95 %	1.22 %	28.36 %	-0.84 %	27.64 %	-3.36 %
FLD_{10}	17.90 %	36.30 %	102.79 %	35.09 %	96.03 %	34.16 %	90.84 %	32.97 %	84.19 %
FLD_{11}	24 %	26.07 %	8.63 %	24.65 %	2.71 %	23.26 %	-3.08 %	21.91 %	-8.71 %
FLD_{12}	26.60 %	26.24 %	-1.35 %	23.58 %	-11.35 %	22.56 %	-15.19 %	20.40 %	-23.31 %
Max. deviation			14.04 %		11.35 %		15.19 %		23.31 %
Sum of rel. dev.			33.30 %		23.34 %		26.43 %		57.61 %
Rel. aver. dev.			**6.66 %**		**4.67 %**		**5.29 %**		**11.52 %**

Table A.4: Fiber volume contents determined by chemical dissolution and by the novel thresholding technique when applying different configurations of the "medianBlur" filter onto the CT images. Extracted from Blarr et al. [188].

Specimen	**Exp. values**	**medianBlur(...,11)**		**medianBlur(...,13)**		**medianBlur(...,15)**		**medianBlur(...,19)**	
	FVC	FVC	Deviation	FVC	Deviation	FVC	Deviation	FVC	Deviation
FLD_1	22.30 %	26.12 %	17.13 %	24.37 %	9.28 %	22.97 %	3.00 %	21.26 %	-4.66 %
FLD_2	25.50 %	27.57 %	8.12 %	25.64 %	0.55 %	24.18 %	-5.18 %	22.49 %	-11.80 %
FLD_3	28.60 %	30.14 %	5.38 %	29.16 %	1.96 %	28.50 %	-0.35 %	27.69 %	-3.18 %
FLD_{10}	17.90 %	36.30 %	102.79 %	39.47 %	120.50 %	38.68 %	116.09 %	36.81 %	105.64 %
FLD_{11}	24 %	26.68 %	11.17 %	25.06 %	4.42 %	23.91 %	-0.37 %	22.31 %	-7.04 %
FLD_{12}	26.60 %	29.20 %	9.77 %	27.56 %	3.61 %	26.64 %	0.15 %	25.79 %	-3.05 %
Max. deviation			17.13 %		9.28 %		5.18 %		11.80 %
Sum of rel. dev.			51.57 %		19.82 %		9.06 %		29.74 %
Rel. aver. dev.			**10.31 %**		**3.96 %**		**1.81 %**		**5.95 %**

Table A.5: Fiber volume contents determined by chemical dissolution and by the novel thresholding technique when applying a "gaussianBlur" and "bilateralFilter" onto the CT images. Extracted from Blarr et al. [188].

Specimen	Exp. values	GaussianBlur(...,(15,15),0)		bilateralFilter(...,15,350,350)	
	FVC	FVC	Deviation	FVC	Deviation
FLD_1	22.30 %	28.30 %	26.91 %	24.39 %	9.37 %
FLD_2	25.50 %	30.96 %	21.41 %	25.56 %	0.24 %
FLD_3	28.60 %	32.39 %	13.25 %	29.23 %	2.20 %
FLD_{10}	17.90 %	39.79 %	122.29 %	35.52 %	98.44 %
FLD_{11}	24 %	28.39 %	18.29 %	25.11 %	4.63 %
FLD_{12}	26.60 %	29.87 %	12.29 %	25.01 %	-5.98 %
Max. deviation			26.91 %		9.37 %
Sum of rel. dev.			92.15 %		22.41 %
Rel. aver. dev.			**18.43 %**		**4.48 %**

A.3 Results of FVC of Plaque 3 determined by AOA thresholding for different pre-filters

Table A.6: Fiber volume contents determined by chemical dissolution and by the novel thresholding technique when applying different configurations of the "medianBlur" filter onto the CT images. Extracted from Blarr et al. [188].

Specimen	**Exp. values**	**medianBlur(...,15)**		**medianBlur(...,21)**		**medianBlur(...,23)**		**medianBlur(...,25)**	
	FVC	FVC	Deviation	FVC	Deviation	FVC	Deviation	FVC	Deviation
C_1	23.07 %	25.88 %	12.21 %	24.41 %	5.81 %	24.07 %	4.36 %	25.39 %	10.06 %
C_2	22.08 %	41.43 %	87.63 %	42.06 %	90.50 %	42.20 %	91.10 %	42.56 %	92.73 %
C_3	23.06 %	25.48 %	10.52 %	23.43 %	1.62 %	23.04 %	-0.08 %	22.59 %	-2.03 %
CF_1	25.57 %	22.65 %	-11.42 %	25.58 %	0.02 %	26.60 %	4.01 %	28.31 %	10.70 %
CF_2	22.31 %	24.52 %	9.90 %	23.65 %	5.97 %	23.53 %	5.44 %	23.48 %	5.24 %
CF_3	22.81 %	23.33 %	2.28 %	21.97 %	-3.69 %	22.53 %	-1.24 %	22.41 %	-1.76 %
F_1	26.36 %	26.90 %	2.07 %	25.70 %	-2.48 %	25.48 %	-3.33 %	25.28 %	-4.09 %
F_2	23.10 %	32.43 %	40.37 %	32.52 %	40.74 %	32.76 %	41.81 %	35.16 %	52.19 %
F_3	23.81 %	26.46 %	11.15 %	25.29 %	6.23 %	25.11 %	5.49 %	25.00 %	5.01 %
Max. deviation			12.21 %		6.23 %		5.49 %		10.70 %
Sum of rel. dev.			59.55 %		25.82 %		23.95 %		38.89 %
Rel. aver. dev.			**8.51 %**		**3.69 %**		**3.42 %**		**5.56 %**
Rel. aver. dev. with outliers			**20.84 %**		**17.45 %**		**17.43 %**		**20.42 %**

Table A.7: Fiber volume contents determined by chemical dissolution and by the novel thresholding technique when applying a "gaussianBlur" and "Blur" onto the CT images. Extracted from Blarr et al. [188].

Specimen	**Exp. values**	**GaussianBlur(...,(21,21),0)**		**blur(...,(15,15))**	
	FVC	FVC	Deviation	FVC	Deviation
C_1	23.07 %	27.03 %	17.20 %	25.86 %	12.13 %
C_2	22.08 %	39.09 %	77.03 %	41.19 %	86.55 %
C_3	23.06 %	26.53 %	15.05 %	24.99 %	8.36 %
CF_1	25.57 %	22.47 %	-12.14 %	21.03 %	-17.78 %
CF_2	22.31 %	25.47 %	14.16 %	24.85 %	11.39 %
CF_3	22.81 %	23.83 %	4.45 %	22.53 %	-1.24 %
F_1	26.36 %	26.00 %	-1.35 %	24.97 %	-5.28 %
F_2	23.10 %	30.93 %	33.88 %	30.41 %	31.64 %
F_3	23.81 %	27.81 %	16.80 %	26.51 %	12.89 %
Max. deviation			17.20 %		17.78 %
Sum of rel. dev.			81.15 %		69.07 %
Rel. aver. dev.			**11.59 %**		**9.87 %**
Rel. aver. dev. with outliers			**21.34 %**		**20.81 %**

Schriftenreihe
des Instituts für Angewandte Materialien

ISSN 2192-9963

Eine vollständige Übersicht der Bände finden Sie im Verlagsshop

Band 50 Michael Selzer
Mechanische und Strömungsmechanische Topologie-optimierung mit der Phasenfeldmethode.
ISBN 978-3-7315-0431-3

Band 51 Michael Mahler
Entwicklung einer Auswertemethode für bruchmechanische Versuche an kleinen Proben auf der Basis eines Kohäsivzonenmodells.
ISBN 978-3-7315-0441-2

Band 52 Christoph Bohnert
Numerische Untersuchung des Verformungs- und Bruchverhaltens von einkristallinem Wolfram auf mikroskopischer Ebene.
ISBN 978-3-7315-0444-3

Band 53 Stefan Guth
Schädigung und Lebensdauer von Nickelbasislegierungen unter thermisch-mechanischer Ermüdungsbeanspruchung bei verschiedenen Phasenlagen.
ISBN 978-3-7315-0445-0

Band 54 Markus Klinsmann
The Effects of Internal Stress and Lithium Transport on Fracture in Storage Materials in Lithium-Ion Batteries.
ISBN 978-3-7315-0455-9

Band 55 Thomas Straub
Experimental Investigation of Crack Initiation in Face-Centered Cubic Materials in the High and Very High Cycle Fatigue Regime.
ISBN 978-3-7315-0471-9

Band 56 Maren Lepple
Kupfer- und Eisenoxide als Konversions-Elektrodenmaterialien für Lithium-Ionen-Batterien: Thermodynamische und Elektrochemische Untersuchungen.
ISBN 978-3-7315-0482-5

Band 57 Stefan Andreas Slaby
Charakterisierung und Bewertung der Zug- und Ermüdungseigenschaften von Mikrobauteilen aus 17-4PH Edelstahl. Ein Vergleich von mikropulverspritzgegossenem und konventionell hergestelltem Material.
ISBN 978-3-7315-0484-9

Band 58 Kumar Ankit
Phase-field modeling of microstructural pattern formation in alloys and geological veins.
ISBN 978-3-7315-0491-7

Band 59 Kuo Zhang
Characterization and Modeling of the Ratcheting Behavior of the Ferritic-Martensitic Steel P91.
ISBN 978-3-7315-0503-7

Band 60 Nicht erschienen

Band 61 Fabian Lemke
Untersuchung des Sinterverhaltens von $SrTiO_3$ unter Berücksichtigung der Defektchemie.
ISBN 978-3-7315-0510-5

Band 62 Johannes Kümmel
Detaillierte Analyse der Aufbauschneidenbildung bei der Trockenzerspanung von Stahl C45E mit Berücksichtigung des Werkzeugverschleißes.
ISBN 978-3-7315-0518-1

Band 63 László Hagymási
Modellierung der Stoffübertragung beim Niederdruckcarbonitrieren mit Ammoniak und Acetylen.
ISBN 978-3-7315-0568-6

Band 64 Reza Eslami
A novel micro-mechanical model for prediction of multiaxial high cycle fatigue at small scales.
ISBN 978-3-7315-0583-9

Band 65 Sebastian Schulz
Phase-field simulations of multi-component solidification and coarsening based on thermodynamic datasets.
ISBN 978-3-7315-0618-8

Band 66 Markus Stricker
Die Übertragung von mikrostrukturellen Eigenschaften aus der diskreten Versetzungsdynamik in Kontinuumsbeschreibungen.
ISBN 978-3-7315-0658-4

Band 67 Luis Straßberger
Untersuchung und Modellierung des viskoplastischen Verformungsverhaltens oxidpartikelverstärkter Stähle.
ISBN 978-3-7315-0674-4

Band 68 Mark Wobrock
Microplasticity of idealized single crystalline Ag cantilevers characterized with methods of high resolution.
ISBN 978-3-7315-0682-9

Band 69 Amritesh Kumar
Micromechanical study on the deformation behaviour of directionally solidified NiAl-Cr eutectic composites.
ISBN 978-3-7315-0694-2

Band 70 Johannes Hötzer
Massiv-parallele und großskalige Phasenfeldsimulationen zur Untersuchung der Mikrostrukturentwicklung.
ISBN 978-3-7315-0693-5

Band 71 Thomas Hupfer
Herstellung von LATP für den Einsatz als Festkörperelektrolyt und dessen Eigenschaften.
ISBN 978-3-7315-0702-4

Band 72 Florentin Pottmeyer
Schädigungsverhalten von in CFK-Laminaten eingebetteten Inserts unter bauteilnahen Beanspruchungen.
ISBN 978-3-7315-0719-2

Band 73 Andres Höweling
Untersuchung der Hochvoltstabilität und Tiefentladung von dotierten $LiNi_{0,5}Mn_{1,5}O_4$-Hochvoltspinellen.
ISBN 978-3-7315-0728-4

Band 74 Tabea Gisela Schwark
Deformation and Fracture Properties of the Soft Magnetic Composite Somaloy 700 3P on Different Length Scales.
ISBN 978-3-7315-0759-8

Band 75 Klaudia Lichtenberg
Metallmatrixverbunde mit Verstärkungselementen aus metallischem Glas $Ni_{60}Nb_{20}Ta_{20}$ – Herstellung und Charakterisierung.
ISBN 978-3-7315-0782-6

Band 76 Claudio Findeisen
Charakterisierung und Modellierung von instabilen Metamaterialien.
ISBN 978-3-7315-0869-4

Band 77 Nilesha Mishra
Influence of strain on the functionality of ink-jet printed thin films and devices on flexible substrates.
ISBN 978-3-7315-0853-3

Band 78 Simon Werner Bonk
Plastische Verformungsmechanismen in hochgradig kaltgewalzten, ultrafeinkörnigen Wolframblechen.
ISBN 978-3-7315-0878-6

Band 79 Tim Gräning
Herstellung, Charakterisierung und Optimierung von austenitischen ODS Stählen.
ISBN 978-3-7315-0732-1

Band 80 Peter Rupp
Herstellung, Prüfung und Modellierung neuartiger hybrider Aluminiumschaum-CFK-Sandwichverbunde.
ISBN 978-3-7315-0880-9

Band 81 Benjamin Sebastian Ehreiser
Einfluss mechanischer Lasten auf die Herstellung von Stahl-Glaskeramik-Verbunden.
ISBN 978-3-7315-0954-7

Band 82 Hans Giel
Weiterentwicklung experimenteller Methoden zur Ermittlung thermodynamischer Werkstoffdaten von Lithium-Ionen-Batterien.
ISBN 978-3-7315-0981-3

Band 83 Anna Trauth
Characterisation and Modelling of Continuous-Discontinuous Sheet Moulding Compound Composites for Structural Applications.
ISBN 978-3-7315-0950-9

Band 84 Jonas Johannes Hüther
The Impact of Recycling on the Fibre and the Composite Properties of Carbon Fibre Reinforced Plastics.
ISBN 978-3-7315-0983-7

Band 85 Nicolas A. Mayer
Thermodynamik von Kobaltoxid Anodenmaterialien für Lithium-Ionen-Batterien und ihr elektrochemisches Verhalten.
ISBN 978-3-7315-0996-7

Band 86 Ulrich Führer
Untersuchung und Modellierung des Haltezeiteinflusses auf die zyklische Entfestigung ferritisch-martensitischer Stähle.
ISBN 978-3-7315-0837-3

Band 87 Ebru Cihan
Structure evolution in tribological interfaces studied by multilayer model alloys.
ISBN 978-3-7315-0999-8

Band 88 Markus Sudmanns
Entwicklung einer Kontinuumsbeschreibung für die Versetzungsmobilität in Versetzungsnetzwerken.
ISBN 978-3-7315-1001-7

Band 89 Tao Zhang
Phase-field Modeling of Phase Changes and Mechanical Stresses in Electrode Particles of Secondary Batteries.
ISBN 978-3-7315-1002-4

Band 90 Markus Ganser
On the Electro-Chemo-Mechanical Coupling in Solid State Batteries and its Impact on Morphological Interface Stability.
ISBN 978-3-7315-1047-5

Band 91 Michael Kellner
Modellierung mehrkomponentiger Materialsysteme für die Phasenfeldmethode und Analyse der simulierten Mikrostrukturen.
ISBN 978-3-7315-1044-4

Band 92 Felix Schröckert
Herstellung dünner Folien aus Lithium-Lanthan-Titanat zur Anwendung als Festkörperelektrolyt.
ISBN 978-3-7315-1008-6

Band 93 Ephraim Schoof
Chemomechanische Modellierung der Wärmebehandlung von Stählen mit der Phasenfeldmethode.
ISBN 978-3-7315-1050-5

Band 94 Alexander Valentin Brabänder
Registrierende Härtemessung an neutronenbestrahlten Materialien bei hohen Temperaturen.
ISBN 978-3-7315-1097-0

Band 95 Denny Schmidt
Einfluss der Kompaktierung auf die Elektrodenmikrostruktur und elektrochemische Performance bei Lithium-Ionen-Zellen.
ISBN 978-3-7315-1098-7

Band 96 Svenja Dittrich
Entwicklung von Siebdruckpasten zur Herstellung von Glaslotfügungen für die Festoxidbrennstoffzelle.
ISBN 978-3-7315-1085-7

Band 97 Michael Dippon
Bestimmung der Betriebsgrenzen für das Schnellladen von Lithium-Ionen Batterien.
ISBN 978-3-7315-1123-6

Band 98 Patricia Haremski
Diffusionseigenschaften von Nickel in einer Festoxid-Brennstoffzelle.
ISBN 978-3-7315-1124-3

Band 99 Florian Wankmüller
Mehrskalige Charakterisierung der Hochtemperatur-Brennstoffzelle (SOFC).
ISBN 978-3-7315-1142-7

Band 100 Niklas Russner
Modellgestützte Analyse des Stackbetriebs von Festoxidzellen.
ISBN 978-3-7315-1144-1

Band 101 Theo Fett, Karl G. Schell, Ethel C. Bucharsky, Gabriele Rizzi, Pascal Hettich, Susanne Wagner, Michael J. Hoffmann
Consequences of hydroxyl generation by the silica/water reaction – Part I: Diffusion and Swelling.
ISBN 978-3-7315-1148-9

Band 102 Theo Fett, Karl G. Schell, Ethel C. Bucharsky, Gabriele Rizzi, Susanne Wagner, Michael J. Hoffmann
Consequences of hydroxyl generation by the silica/water reaction – Part II: Global and local Swelling
Part III: Damage and Young's Modulus.
ISBN 978-3-7315-1159-5

Band 103 Johannes Dornheim
Modellfreies Lernen optimaler zeitdiskreter Regelungsstrategien für Fertigungsprozesse mit endlichem Zeithorizont.
ISBN 978-3-7315-1158-8

Band 104 Markus Muth
Grundlagenuntersuchungen an intrinsisch gefertigten lasttragenden FVK/Metall-Hybridträgern.
ISBN 978-3-7315-1161-8

Band 105 Oleg Birkholz
Modeling transport properties and electrochemical performance of hierarchically structured lithium-ion battery cathodes using resistor networks and mathematical half-cell models.
ISBN 978-3-7315-1172-4

Band 106 Verena Irene Becker
Modellierung der Mechanik und der effektiven Transporteigenschaften von partikulären Kathoden sowie deren Einfluss auf die elektrochemische Performance von Lithium-Ionen-Batterien.
ISBN 978-3-7315-1174-8

Band 107 Nikolai Zimber
Nanoskalige Analytik der Mikrostruktur von hochdosig bestrahltem Beryllium.
ISBN 978-3-7315-1178-6

Band 108 Francesco Mazzocchi
Development of NbN based Kinetic Inductance Detectors on sapphire and diamond substrates for fusion plasma polarimetric diagnostics.
ISBN 978-3-7315-1181-6

Band 109 Adrian Schmidt
Multiskalige Modellierung von Lithium-Ionen-Batterien.
ISBN 978-3-7315-1227-1

Band 110 Maximilian Röhe
Dynamic Model-based Analysis of Oxygen Reduction Reaction in Gas Diffusion Electrodes.
ISBN 978-3-7315-1234-9

Band 111 Michael Späth
Phasenfeldmodellierung von Bruchbildungs-, Kristallisations- und Auflösungsvorgängen in hydrothermalen Umgebungen.
ISBN 978-3-7315-1242-4

Band 112 Friedemann Streich
Numerische Modellierung und Simulation von bleifreien Relaxor-Ferroelektrikum-Kompositen.
ISBN 978-3-7315-1259-2

Band 113 Paul S. Zielonka
Entwicklung und Charakterisierung von Siliziumnitrid-/Siliziumkarbidkompositen für den Einsatz in tribologisch hochbeanspruchten Gleitsystemen.
ISBN 978-3-7315-1269-1

Band 114 Carsten Bonnekoh
Der Spröd-duktil-Übergang in ultrafeinkörnigem Wolfram.
ISBN 978-3-7315-1264-6

Band 115 Marcel Heinzmann
Analyse und Modellbildung von PEM-Brennstoffzellen mittels elektrochemischer Impedanzspektroskopie.
ISBN 978-3-7315-1319-3

Band 116 Felix Sutter
Materiell nichtlineare Kontinuumsmodellierung ferroelektrischer Funktionskeramiken mit piezoelektrischen und flexoelektrischen Eigenschaften.
ISBN 978-3-7315-1337-7

Band 117 Lukas Schöller
Multiscale Modeling of Curing and Crack Propagation in Fiber-Reinforced Thermosets.
ISBN 978-3-7315-1340-7

Band 118 Benjamin Hauck
Impedanzbasierte Spannungsprädiktion von Lithium-Ionen-Batterien.
ISBN 978-3-7315-1347-6

Band 119 Yannick Lingelbach
Application of Data Mining and Machine Learning Methods to Industrial Heat Treatment Processes for Hardness Prediction.
ISBN 978-3-7315-1352-0

Band 120 Philipp Lied
Steigerung der thermischen Stabilität von warm- und kaltgewalztem Wolfram durch Kalium-Dotierung für die Fusionsenergietechnik.
ISBN 978-3-7315-1356-8

Band 121 Paul Wilhelm Hoffrogge
Understanding Degradation Phenomena in Solid-Oxide Fuel-Cell Anodes by Phase-Field Modeling and Analytics.
ISBN 978-3-7315-1363-6

Band 122 Bastian Kraft
Herstellung und Charakterisierung von Nb- und Ta-Al2O3-Verbundwerkstoffen für Hochtemperaturanwendungen.
ISBN 978-3-7315-1378-0

Band 123 Peter Smyrek
Laserinduzierte Plasmaspektroskopie an strukturierten Lithium-Nickel-Mangan-Kobalt-Oxid Dickschichtelektroden.
ISBN 978-3-7315-1380-3

Band 124 Juliane Blarr
Development of Computational, Image Processing and Deep Learning Methods for the Microstructure Characterization of Carbon Fiber Reinforced Polyamide 6 Based on CT Images.
ISBN 978-3-7315-1396-4

KARLSRUHER INSTITUT FÜR TECHNOLOGIE (KIT)
SCHRIFTENREIHE DES INSTITUTS FÜR ANGEWANDTE MATERIALIEN

Discontinuously fiber reinforced polymers exhibit complex microstructures. Quantities to characterize the latter have been developed over time, such as the fiber volume content or fiber orientation distributions, which can be acquired through computed tomography images and subsequent image processing. This work deals with the development of both deterministic and AI based methods in this context, especially considering challenges of contrast and resolution with carbon fibers and scale-bridging issues.

ISSN 2192-9963
ISBN 978-3-7315-1396-4

Gedruckt auf FSC-zertifiziertem Papier