Graz University of Technology

Institute of Biomechanics
Stremayrgasse 16/II
8010 Graz, Austria

Master Thesis

Segmentation of collagen fibrils and proteoglycans from reconstructed tilt series acquired by TEM

to achieve the degree of Diplom-Ingenieur

Author: Sarah Anna Schneider
Registration number: 01330624

Supervisors: Dr.techn. Dipl.-Ing. Patrick Knöbelreiter
Univ.-Prof. Dipl.-Ing. Dr.techn. Thomas Pock
Inz. Mgr Inz. Anna Pukaluk
Ass.Prof. Dipl.-Ing. Dr.techn. Gerhard Sommer

Head of Institute: Univ.-Prof. Dipl.-Ing. Dr.techn. Professor Gerhard A. Holzapfel

October 15, 2020

Contents

Abstract

The interaction of proteoglycans with collagen fibrils in arterial tissue is still a matter of an ongoing debate, different theories have been developed to explain their interconnected mechanical behaviour. Measuring the response of the tissue to the application of mechanical stretch with subsequent generation and examination of three-dimensional images is an interesting and promising strategy.

However, evaluating the volumetric images recorded for this purpose takes the capacity of the human visual system to its limits, analysing them without any computational assistance is hardly possible. Therefore the aim of this thesis was to investigate, implement and compare different procedures for segmenting proteoglycans and collagen fibrils in images.

As a first simple approach, thresholding was applied. Proteoglycans were found to be rather tube-like, therefore, as a second approach, a tube detection filter was examined. For both thresholding and tube detection filtering, preprocessing the images with a Total Variation l1 denoising was necessary.

Next, two deep learning models based on encoder-decoder networks were investigated. As there were not many labelled images available, data augmentation was crucial.

The resulting segmentation masks were reliable and, as an additional task, orientations and diameters of segmented proteoglycans were evaluated with respect to axial, circumferential and radial direction of the tissue samples.

Kurzfassung

Die Interaktion von Proteoglykanen und Kollagenfibrillen im arteriellen Gewebe wird derzeit eingehend diskutiert, verschiedene Theorien wurden entwickelt um das mechanisch miteinander gekoppelte Verhalten beider Strukturen zu erklären. Das Messen des resultierenden Materialverhaltens auf eine mechanische Belastung mit anschließender Untersuchung von bildgebenden Darstellungen des Gewebes ist eine vielversprechende Strategie.

Die zu diesem Zweck aufgenommenen dreidimensionalen Aufnahmen bringen das menschliche visuelle System an seine Grenzen, das Analysieren der Bilder ohne eine computerbasierte Unterstützung ist schier unmöglich. Deshalb war es das Ziel dieser Masterarbeit, Methoden zur Segmentierung von Proteoglykanen und Kollagenfibrillen zu untersuchen, zu implementieren und zu vergleichen.

Der erste einfachste Ansatz war ein Thresholding der Bilder. Da die Proteoglykane sehr " tube-ähnlich" sind, wurde außerdem ein Tube-Detection Filter untersucht. Für das Thresholding und den Tube-Detection Filter war eine Vorverarbeitungsfilterung basierend auf einem Total Variaton l1 Modell notwendig.

Anschließend wurden zwei Deep Learning Modelle basierend auf Encoder-Decoder-Architekturen getestet. Daten Augmentierung war äußerst hilfreich, da nur sehr wenige annotierte Bilder zur Verfügung standen.

Die erzielten Segmentierungen waren so zuverlässig, dass zusätzlich eine Untersuchung der Orientierungen und Durchmesser der segmentierten Proteoglykane durchgeführt werden konnte. Diese wurde so ausgeführt, dass die Ergebnisse im Bezug auf die axiale, circumferentiale und radiale Richtung der Gewebeprobe ausgewertet werden konnten.

Acknowledgment

People seldom improve when they have no model but themselves to copy after. - Oliver Goldsmith

I would like to thank Gerhard Sommer for his steady helpfulness and kindness. Many thanks are due to Anna Pukaluk for her detailed and immediate advice. I was repeatedly impressed by her expertise on the behaviour and structure of human aortas.
I would like to express my sincere gratitude to Patrick Knöbelreiter and Thomas Pock. Their reassurance, humour, immense competence and faith in me made every challenge manageable. I always felt like there was a safety net of knowledge and guidance around me.
Many thanks go to my friends, their wise and encouraging words always lifted my spirits. Having them by my side made life in Graz as good as it could be.
Finally, I would like to thank my family for their unconditional support and patience not only during my time at university, but also at every other stage of life. They provided cheerful distraction whenever I needed it. In our "home office" in Tyrol, which felt more like a training camp from time to time, they showed me that sometimes the best ideas do not occur in front of a computer, but on top of a mountain.

1 Introduction

Arteries fulfil crucial tasks in the human body. They transport blood from the heart to tissues and organs, providing them with oxygen and nutrients. Arterial tissue has to cope with 35 - 40 million heartbeats, which means 35 - 40 million load cycles, during one year. If the tissue is affected by disease, the outcome can be life threatening and will at least decreasing the patient's quality of life. Pathogenetic development of vascular tissue accompanying diseases, such as hypertension, atherosclerosis or aneurysms, has been associated with a restructuring and reordering of the extracellular matrix and its functions. Interactions of the extracellular matrix with other constituents are altered. Due to its high stiffness, collagen is known to play a major role in the load-bearing characteristic of the arterial tissue (Holzapfel, 2008). As collagen fibrils are embedded in an extracellular matrix with significant amounts of proteoglycans, particular attention needs to be directed to their influence (Linkan et. al., 2016).

In order to find out more about the interaction of proteoglycans with collagen fibrils, the Insitute of Biomechanics at the University of Technology Graz has started investigating human aortas. Samples are tested and evaluated mechanically, loads are applied and mechanical responses measured. Mechanically treated tissue samples are then inspected using electron microscopy.

As a unit, our eyes and our brain are remarkably good when it comes to evaluating images and our ability to detect certain structures in images sets a high standard for any computer vision approach. However, interpreting these electron microscopy images constitutes a challenge for the human visual system. They are quite noisy, structures are not distinctively depicted and difficult to distinguish. In order to investigate them in a structured way, computational segmentation is necessary.

1.1 Collagen Fibrils

In humans, 28 different collagen types are known. For vascular tissue, fibrillar collagens of type I and type III are the most prominent ones. Each fibril forming collagen molecule is composed of three parallel left-handed helical polypeptide strands. These so called α - chains are characterized by a repeating amino acid sequence: Glycin-X-Y. X and Y can be any amino acid. These three α - chains then form a right-handed triple-helix. Collagen molecules can be either homotrimeric, which means that they consist of three identical α - chains, or heterotypic, which means that they consist of different α - chains. Classical fibrillar collagens have a long central triple-helical region in each polypeptide α - chain.

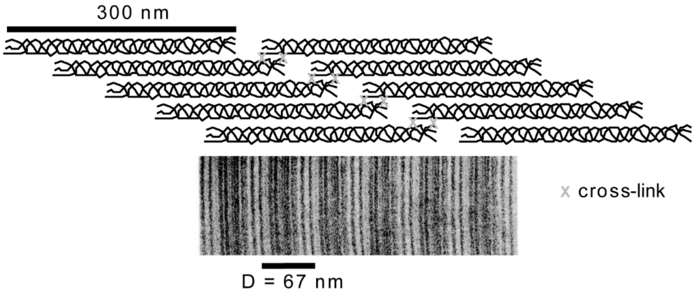


Figure 1.1 Illustrative example of the assembly of collagen molecules to collagen fibrils (Hulmes, 2008).

For collagen molecules of type I and III, this central helical region is flanked by short non-helical segments (Hulmes, 2008).

Collagen molecules self-assemble into D-periodic cross-striated fibrils with specific gap and overlap regions, see Figure 1.1. D is the characteristic axial periodicity of the collagen molecules. Collagen molecules are bound to collagen molecules of the next ”package” by covalent cross-links. The assembly process is driven by the loss of solvent molecules from the surface of the proteins, this minimizes the surface-area-to-volume-ratio of the final arrangement and is therefore favourable (Kadler et. al., 1996).

1.2 Proteoglycans

A basic proteoglycan (PG) consists of a core protein with one or more covalently bound glycosaminoglycans (GAGs). GAGs are long unbranched chains of repeating disaccharide units (Mattson et al., 2019). Figure 1.2 shows a schematic illustration of a proteoglycan. Proteoglycans were not considered to have a significant impact on the tissue and and thus often neglected. However, more and more studies have shown that they do have an influence on the behaviour of the arterial wall. For example, an increased presence of proteoglycans can cause a dissection of arterial wall layers. In contrast, insufficient levels of proteoglycans decrease the ability of the blood vessel to regulate electric charges and structure and was suggested to cause aortic rupture due to their influence on the functionality of other components (Mattson et al., 2019).

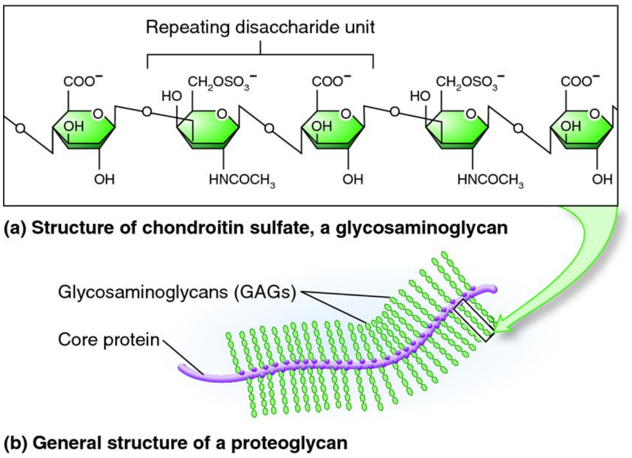


Figure 1.2 Illustrative figure of a proteoglycan structure (taken from https://biology-forums.com/index.php?action=gallery;sa=view;id=283, date of access: 31.08.2020.)

1.2.1 Mechanical Role of Proteoglycans

The mechanical role of PGs is still controversial. Different theories have been developed to explain the mechanical role of proteoglycans. The first theory states that proteoglycans act as elastic bridges between collagen fibrils, influencing their mechanical and stress relaxation properties. Proteoglycans are assumed to transmit loads between fibrils. A second assumption considers the proteoglycans and their glycosaminoglycans as an interjacent medium with zero stiffness under tension. Based on this assumption, a second theory is developed, stating that PGs do not contribute to load transmission between fibrils. A third ”slide-stuck” theory suggests that GAGs promote interfibrillar sliding. PGs are assumed to act as intermediate clutches between adjacent fibrils, protecting fibrils from rupturing (Linkan et. al., 2016).

1.3 Acquisition of three-dimensional Images of differently stretched human aortic Samples

In order to find out more about the mechanical influence of proteoglycans on collagen structures, the Institute of Biomechanics tested tissue samples of the human aorta biaxially. An equibiaxial stretch driven test protocol was used in order to achieve different stretch values: 1.00, 1.05, 1.10 and 1.15. When the desired stretch values were reached, the testing process was stopped and the sample was fixed with 2% glutaraldehyde. To visualize the nanostructure of the sample by a transmission electron microscope, it was embedded in Epoxy resin and it was cut in ultra-thin slices (approx. 250 nm). These slices were brought on a copper grid which was then attached in the transmission electron microscope.

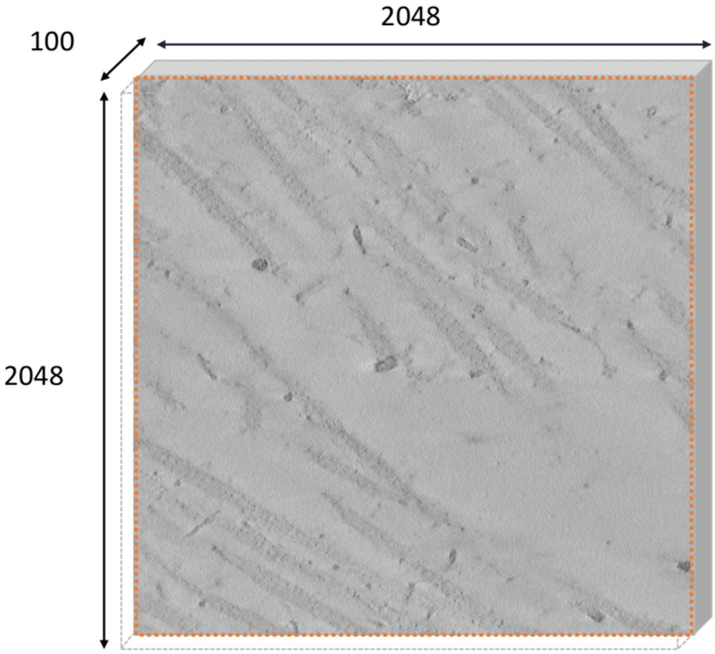
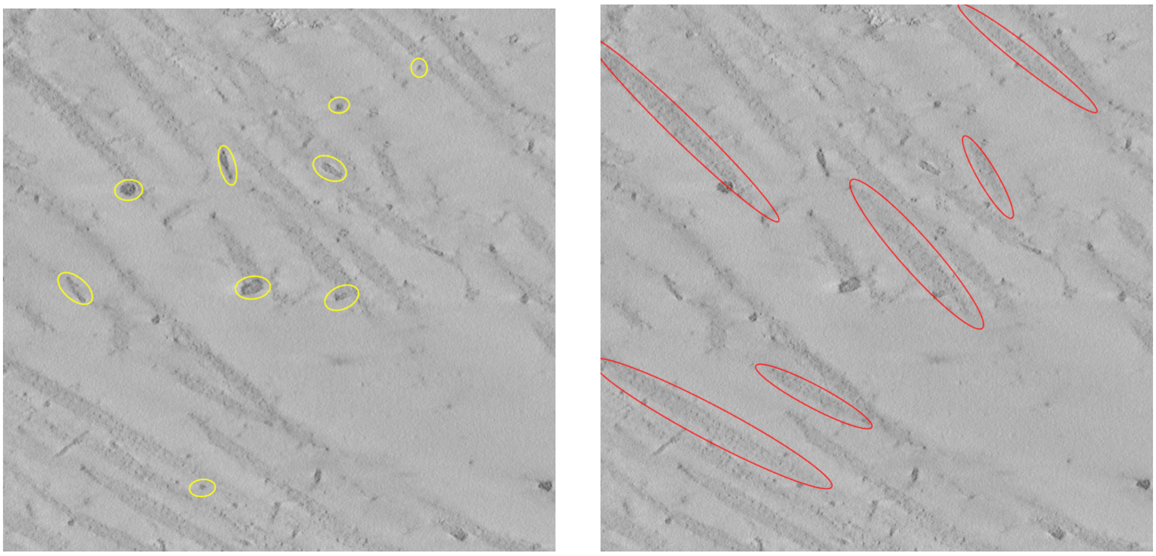


Figure 1.3 Slice of a reconstructed three-dimensional image generated by the transmission electron microscope. All values are given in voxels.



(a) Proteoglycans (b) Collagen fibrils

Figure 1.4 Slice of original image with some proteoglycans marked with a yellow frame and some collagen fibrils marked with a red frame.

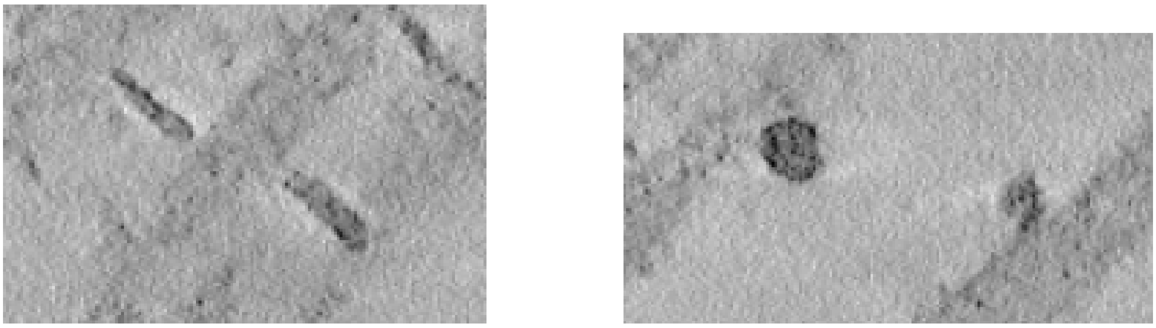
In order to get a three-dimensional image of the tissue, a tilt series with a tilt angle ranging from -65° to 65° and an angle increment of 1° was recorded. For alignment and 3D reconstruction out of tilt projections, the sample was stained with protein A coated colloidal 10 nm gold particles as reference points.

These three-dimensional reconstructed grayscale images with a resolution of $100 \times 2048 \times 2048$ voxels were the starting point for this thesis. Figure 1.3 illustrates what is meant if it is referred to image slices throughout the thesis, the three-dimensional images can be considered to be composed of 100 image slices with a resolution of 2048×2048 voxels. In Figure 1.4 some proteoglycans and collagen fibrils are highlighted.

2 Segmentation of Proteoglycans

2.1 First Insights

As a first step, a concept of the appearance and size of the proteoglycans in the images was developed. It could be observed that some proteoglycans look rather circular in an image slice, while others were more ellipsoidal, see Figure 2.1.

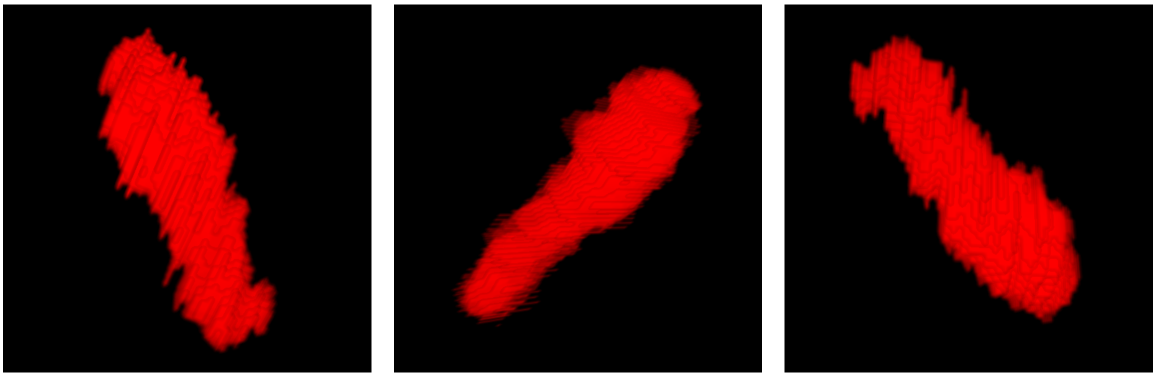


(a) Type 1: Ellipsoidal PG

(b) Type 2: More circular PG

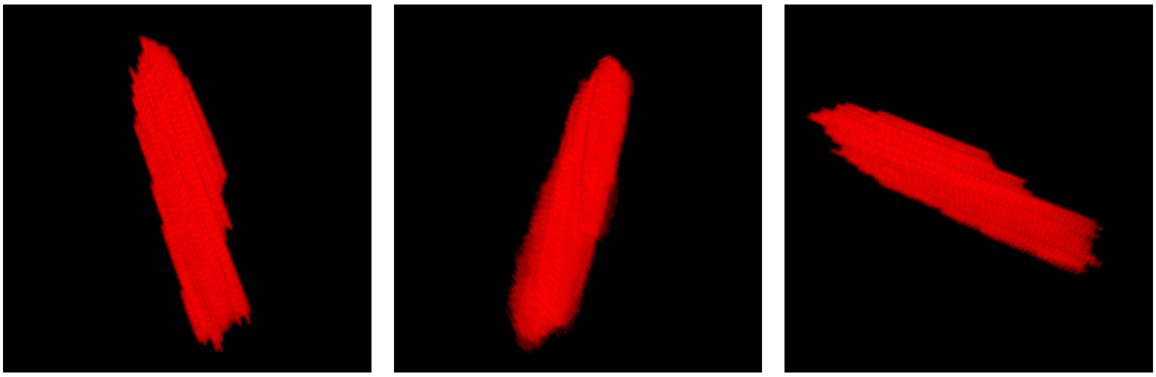
Figure 2.1 Exemplary proteoglycans of defined types, i.e., ellipsoidal in Figure 2.1a and circular in Figure 2.1b.

In order to find out more about their three-dimensional shape, a three-dimensional manual segmentation of individual proteoglycans was done using the segmentation editor of Fiji. It turned out that their depicted shape varies in a two-dimensional image slice because they are oriented differently in the three-dimensional image stack. Their three-dimensional shapes are all rather similar and can be described as “sausage”-like. Figure 2.2 shows a manual segmentation of a proteoglycan which appeared circular in an image slice. Figure 2.3 shows a segmentation of a proteoglycan which was depicted more ellipsoidal in the individual image slices.



(a) (b) (c)

Figure 2.2 Manually segmented proteoglycan appearing circular in two-dimensional image stack slices. Views from different perspectives (Figure 2.2a - 2.2c)



(a) (b) (c)

Figure 2.3 Manually segmented proteoglycan appearing ellipsoidal in two-dimensional image stack slices. Views from different perspectives (Figure 2.3a - 2.3c)

2.2 Preprocessing: TV-l1 Denoising

The original reconstructed results were quite noisy. Structures corresponding to proteoglycans might appear darker than the background, but zooming in one can observe that the dark pixels are “disconnected” by lighter coloured pixels, see Figure 2.4. In order to be able to use thresholding and tube detection filters, preprocessing was necessary.

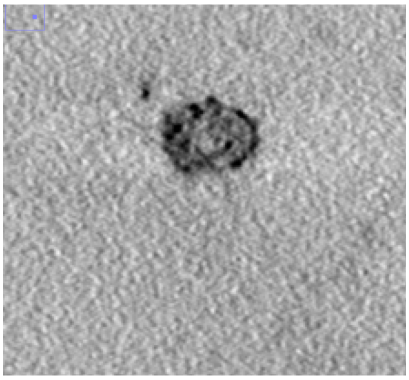
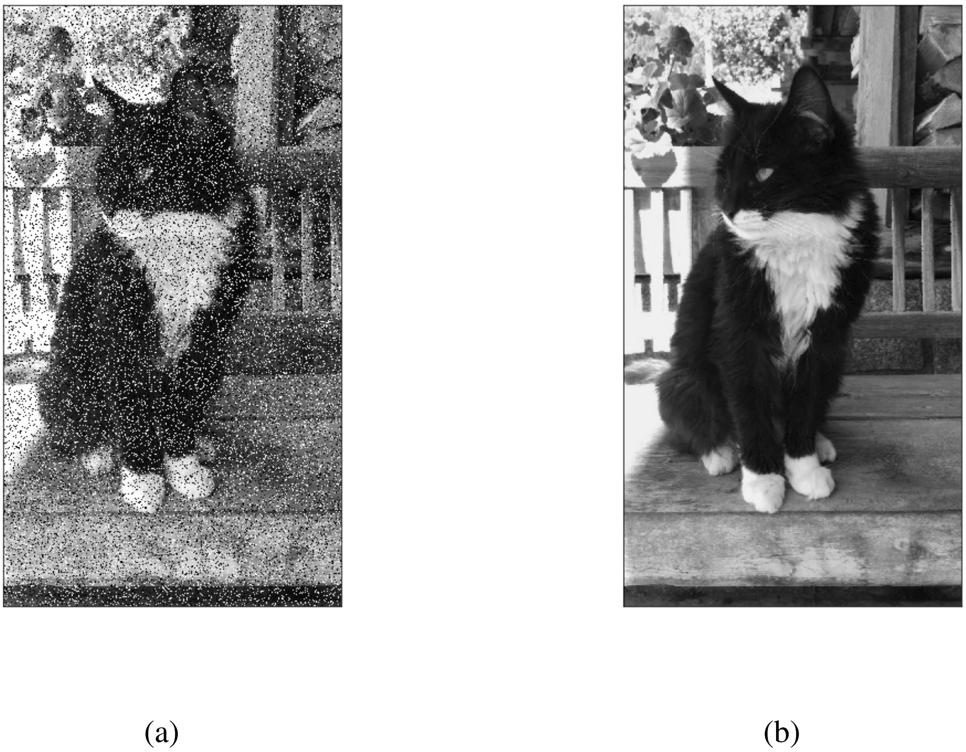


Figure 2.4 Magnification of a proteoglycan in the image stack.

To get rid of the irregularities in voxel intensities, a Total Variation l1 denoising was implemented. This denoising method is contrast independent and purely geometric, therefore it can be used to remove structures of a certain size (Chambolle et al., 2016).



(a) (b)

Figure 2.5 TV-l1 denoising can be used to remove salt and pepper noise. An example of an image before (Figure 2.5a) and after TV-l1 denoising (Figure 2.5b)

2.2.1 Methods

The TV-l1 model is the following:

$$\min_u \int_\Omega |Du| + \lambda \int_\Omega |u(x) - u^{\Diamond}(x)|dx \tag{2.1}$$

where Ω is the image domain and $u^{\Diamond}$ is the given noisy image. The left term, $Du = \nabla u dx$, is called a regularization term. It makes sure, that the total amount of edges in the image will be penalized while strong discontinuities corresponding to strong edges will remain. By selecting a suitable parameter λ, we can remove the ”disconnecting” light coloured voxel areas from the image in order to get ”connected” dark proteoglycan structures. The TV-l1 model (equation 2.1) is non-differentiable, it can not be minimized by standard gradient descent methods.

However, one can observe that the TV-l1 model matches the following problem:

$$\min_u f(Ku) + g(u) \tag{2.2}$$

where f and g are convex lower semicontinuous functions and K a linear operator. Instead of minimizing the original non-differentiable problem, a solution can be found for the equivalent saddle point problem with $K = D$ and a dual variable p:

$$\min_u \max_p \langle Du, p\rangle + g(u) - f^*(p) \tag{2.3}$$

One can see that $g(u) = \lambda||u - u^{\Diamond}||_1$ and the convex conjugate of f is $f^*(p) = \delta_{\{2,\infty\leq 1\}}(p)$. Combining this, one can state equation 2.4 as:

$$\min_u \max_p \langle Du, p\rangle + \lambda||u - u^{\Diamond}||_1 - \delta_{\{2,\infty\leq 1\}}(p) \tag{2.4}$$

To solve this saddle point problem, the following primal-dual algorithm was implemented:

```
for every $k \geq 0$ do
    $u^{k+1}, p^{k+1}$ by solving
        $u^{k+1} = \text{prox}_{\tau,g}(u^k - \tau D^* p^k)$
        $p^{k+1} = \text{prox}_{\sigma,f^*}(p^k + \sigma D(2u^{k+1} - u^k)$
end
```

Algorithm 1: TV-l1 primal-dual algorithm.

The proximal map of $\tilde{u}^{k+1} = u^k - \tau D^* p^k$ with respect to g is the following:

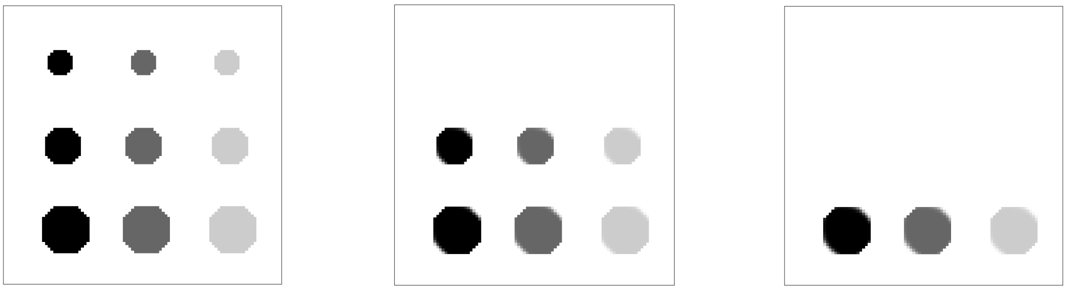
$$\text{prox}_{\tau,g}(\tilde{u}^{k+1}) = u^{\Diamond} + \max\{0, |\tilde{u}^{k+1} - u^{\Diamond}| - \tau\lambda\}\ \text{sign}\{\tilde{u}^{k+1} - u^{\Diamond}\} \tag{2.5}$$

The proximal map of the dual variable $\tilde{p}^{k+1} = p^k + \sigma D(2u^{k+1} - u^k)$ with respect to f^* is

$$\text{prox}_{\sigma,f^*}(\tilde{p}^{k+1}) = \frac{\tilde{p}^{k+1}}{\max\{1, ||\tilde{p}^{k+1}||_2\}}. \tag{2.6}$$

The term $2u^{k+1} - u^k = u^{k+1} - (u^k - u^{k+1})$ is called overrelaxation step and can be seen as an approximated ”extra” gradient (Pock, 2020; Chambolle et al., 2016).

λ can be chosen to remove structures up to a certain size. It is multiplied to the right term of the original TV-l1 model, which is responsible for the difference between solution and original image. This means that if λ is set to a high value, small structures are removed. A low value for λ makes sure that larger structure are erased.

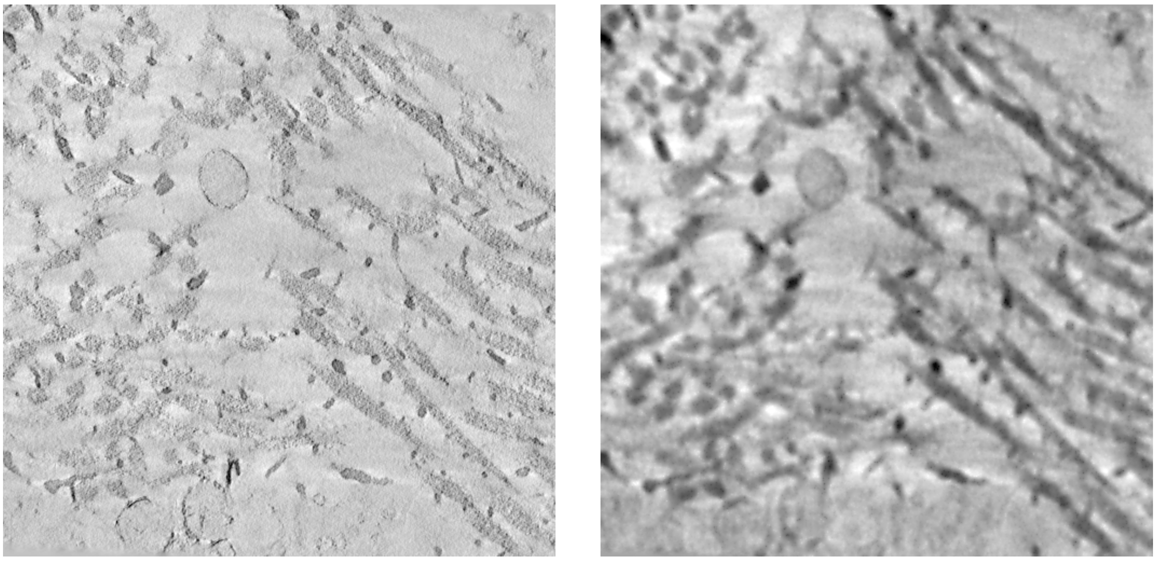


(a) Original image (b) TV-l1 filter result, $\lambda = 0.4$ (c) TV-l1 filter result, $\lambda = 0.3$

Figure 2.6 TV-l1 denoising can be used to remove structures of a certain size by setting λ to an appropriate value, independent of contrast. The original image in Figure 2.6a displays dots of differnt sizes and intensities, Figures 2.6b and 2.6c show that the lower the chosen value for λ is, the larger the sizes of the removed dots are.

2.2.2 Results and Discussion

The algorithm was applied in 3D. To decrease the computation time, the resolution was decreased to $25 \times 512 \times 512$ voxel.



(a) Original image slice (b) Denoised image slice

Figure 2.7 Image slice of the original images stack (Figure 2.7a), and the TV-l1 filter result (Figure 2.7b) of the same slice. Parameter λ was set to 0.32, 500 iterations were performed. It can be seen that proteoglycan structures are now ”continuously” coloured, their outer edges have been kept.

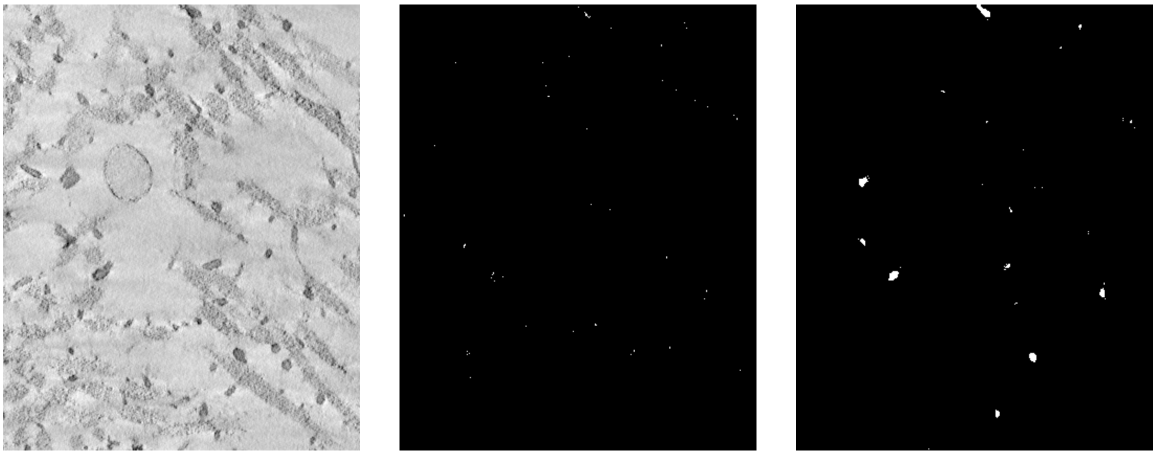
It can be seen in Figure 2.7 that proteoglycan structures are now ”continuously” coloured while their outer edges have been preserved. The selection of a suitable value for λ was not always straightforward and had to be adjusted for every dataset. If the chosen value was too high, the denoising effect was not sufficient. Setting it too low resulted in a removal of structures of interest.

2.3 Thresholding

2.3.1 Methods

It can be seen that the proteoglycans are in most cases depicted darker than other structures, therefore a thresholding was tested as a first simple approach after previous TV-l1 preprocessing. The images were normalized between 0 and 1, then a suitable threshold was set. Voxels with an intensity below a chosen threshold t were set to 1, others were set to 0.

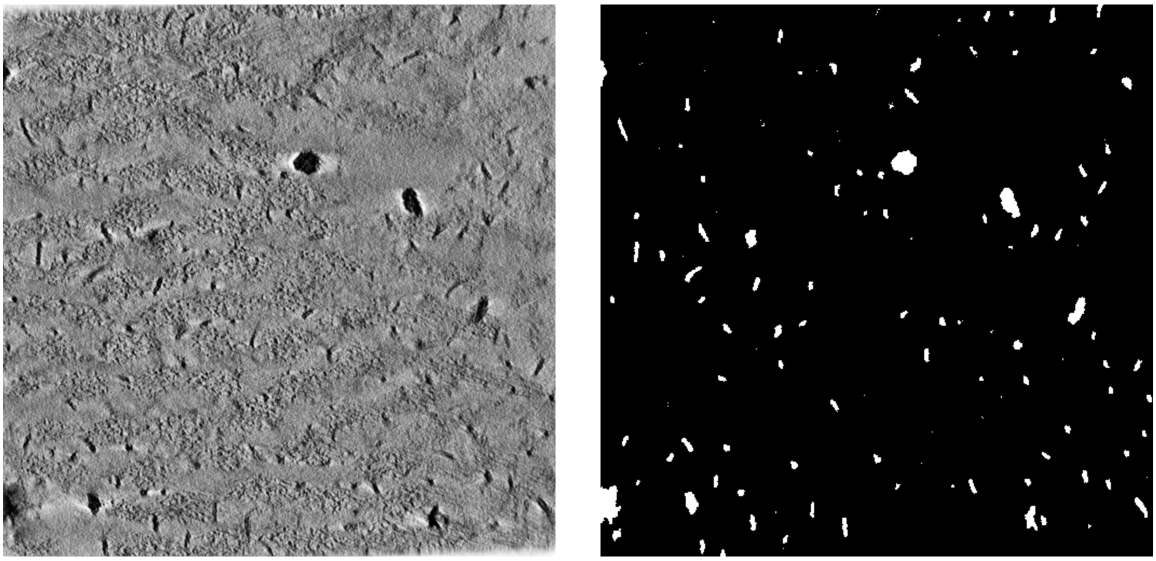
2.3.2 Results and Discussion



(a) Original image
(b) Thresholded image without TV-l1 filtering
(c) Thresholded image with applied TV-l1 filtering

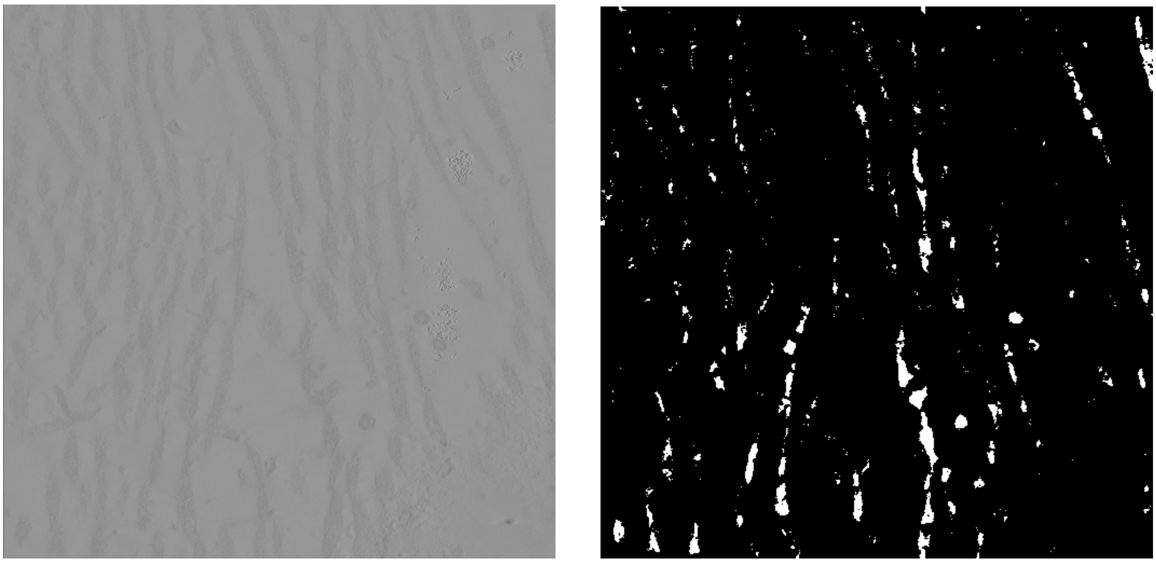
Figure 2.8 Comparison of thresholding ($t = 0.3$) with and without applied TV-l1 preprocessing. Cutout of the unprocessed original image (Figure 2.8a) is shown together with the same region of the thresholded result without any preprocessing (Figure 2.8b) and with previous TV-l1 filtering (Figure 2.8c).

Figure 2.8 shows nicely that the TV-l1 preprocessing was highly necessary in order for a simple thresholding to work at all. Without it, the proteoglycans are not ”continuously” darker than other structures and many voxels of the proteoglycan area are not detected. Figure 2.10 shows an example of an image dataset where many proteoglycans were segmented surprisingly well. For other datasets the contrast between proteoglycans and collagen fibrils is very low, a distinction by a single threshold value is nearly impossible. To conclude, it can be said that although thresholding is a very simple method, it led to nice outcomes for certain images. However, finding a suitable threshold can be cumbersome. If it is set too low, many proteoglycans are not detected. If the chosen value is too high, voxels which are not part of a proteoglycan are considered. The threshold has to be readjusted and fine-tuned for every dataset. Additionally, in order for the method to work at all, the images have to be denoised significantly.



(a) Original image slice (b) Thresholded image slice

Figure 2.9 Slice 17 of original image stack (Figure 2.9a) and thresholded image stack (Figure 2.9b). Thresholding led to reasonable results for this dataset, $t = 0.34$.



(a) Original image slice (b) Thresholded image slice

Figure 2.10 Slice 17 of original image stack (Figure 2.10a) and thresholded image stack (Figure 2.10b). Example of a dataset where thresholding led to poor results, $t = 0.39$.

2.4 Tube Detection Filters

As it has been observed that proteoglycans look rather tube-like, a tube detection filter was investigated. Although collagen fibrils might be described as tube-like too, they are never completely depicted in the images, only some segments of their total structure are imaged in the datasets. Therefore the tube detection filter was applied to detect proteoglycans only.

2.4.1 Methods

Tubularity Assumption

The fundamental base for the algorithms is a mathematical tubularity assumption. Tubular structures have a cross section which is shaped like a disc. The intensity values along the tube will remain rather constant while the intensities in the normal plane will change similarly to a Gaussian distribution. The neighbourhood of a pixel $\mathbf{x_0}$ can be examined via Taylor expansion around the pixel:

$$I(\mathbf{x_0} + \mathbf{\Delta x}) \approx I(\mathbf{x_0}) + \mathbf{\Delta x}\nabla I(\mathbf{x_0}) + \frac{1}{2}\mathbf{\Delta x}^T \nabla^\mathbf{2} I(\mathbf{x_0})\mathbf{\Delta x} \tag{2.7}$$

where ∇ represents the differential operator, see equation 2.8.

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \tag{2.8}$$

$\nabla^\mathbf{2}$ is defined as a dyadic multiplication of ∇ in the following way:

$$\nabla^\mathbf{2} = \nabla \cdot \nabla^T = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial^2}{\partial x \partial x} & \frac{\partial^2}{\partial x \partial y} & \frac{\partial^2}{\partial x \partial z} \\ \frac{\partial^2}{\partial y \partial x} & \frac{\partial^2}{\partial y \partial y} & \frac{\partial^2}{\partial y \partial z} \\ \frac{\partial^2}{\partial z \partial x} & \frac{\partial^2}{\partial z \partial y} & \frac{\partial^2}{\partial z \partial z} \end{pmatrix} \tag{2.9}$$

Using these operators, $\nabla I(\mathbf{x_0})$ and $\nabla^\mathbf{2} I(\mathbf{x_0})$ denote the gradient and the Hessian matrix at $\mathbf{x_0}$, respectively. In order to be able to take different scales into account, $\nabla^\mathbf{2} I^\sigma(\mathbf{x_0})$ describes the Hessian matrix of pixel $\mathbf{x_0}$ at scale σ. The anterior term σ^2 in equation 2.10

is needed to ensure invariance under image scaling.

$$\nabla^{\mathbf{2}} I^{\sigma}(\mathbf{x_0}) = \sigma^2 \begin{pmatrix} \frac{\partial^2 I}{\partial x \partial x} & \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial x \partial z} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y \partial y} & \frac{\partial^2 I}{\partial y \partial z} \\ \frac{\partial^2 I}{\partial z \partial x} & \frac{\partial^2 I}{\partial z \partial y} & \frac{\partial^2 I}{\partial z \partial z} \end{pmatrix} \tag{2.10}$$

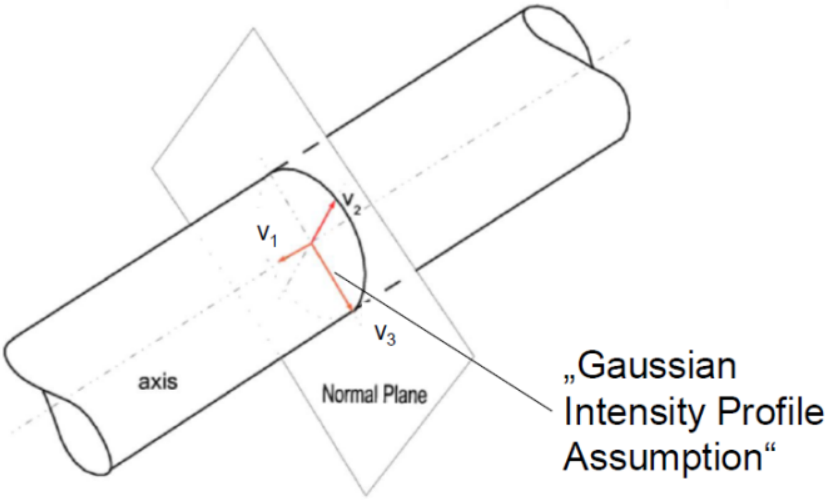
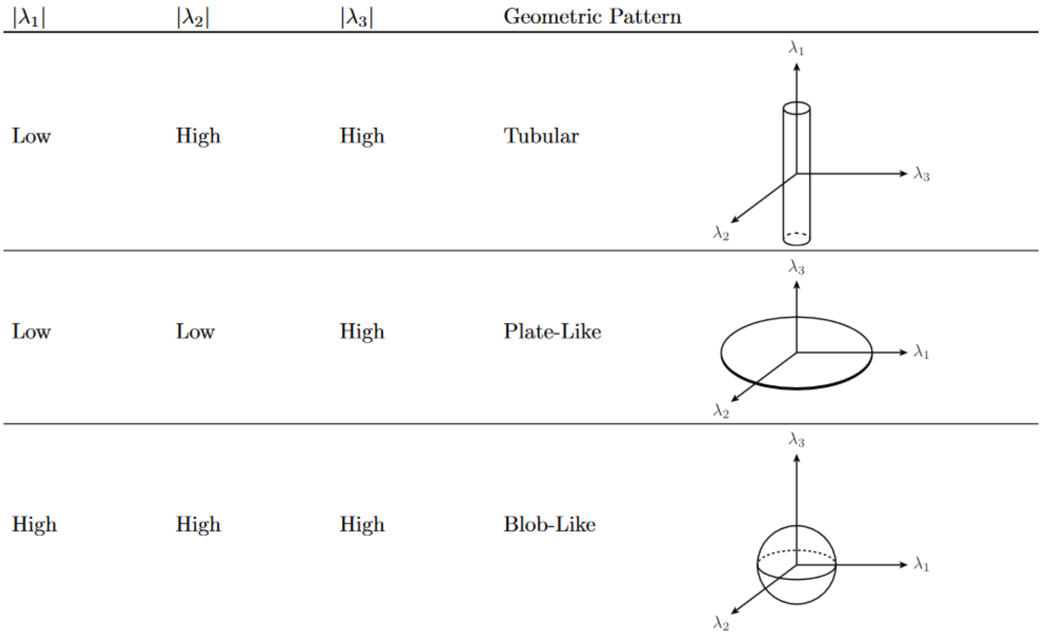


Figure 2.11 Visualization of eigenvectors of a ’perfect’ tube (Urschler, 2019).

Examining the Hessian matrix by looking at its eigenvalues and eigenvectors, information about the local curvature around a pixel $\mathbf{x_0}$ can be gathered.

An eigenvalue decomposition of the Hessian matrix results in three orthonormal directions, $\mathbf{v_1}, \mathbf{v_2}$ and $\mathbf{v_3}$, with corresponding eigenvalues λ_1, λ_2 and λ_3 where $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$ and $|\mathbf{v_i}| = 1$.

Figure 2.12 shows how the magnitude of the eigenvalues will look like for different three-dimensional shapes.



$\lvert\lambda_1\rvert$	$\lvert\lambda_2\rvert$	$\lvert\lambda_3\rvert$	Geometric Pattern
Low	High	High	Tubular
Low	Low	High	Plate-Like
High	High	High	Blob-Like

Figure 2.12 Different three-dimensional geometric shapes and their corresponding eigenvalues (Oruganti et al., 2013).

For a tube, λ_1 is the smallest eigenvalue, ideally zero. Therefore $\mathbf{v_1}$ can be assumed to show in the direction along the tube because the smallest change of intensity values is expected in this direction. λ_2 and λ_3 have a high magnitude and a positive sign if the tubular structures are dark in front of a light background and a negative sign if the structures are lighter coloured than the background. It can be assumed that $\mathbf{v_2}$ and $\mathbf{v_3}$ are lying in the orthogonal plane of the tube. Figure 2.11 visualizes the eigenvector directions.

Summarizing the statements above, for an ideal tube in a 3D image stack, the following should be true:

$$|\lambda_1| \approx 0 \tag{2.11}$$

$$|\lambda_1| \ll |\lambda_2| \tag{2.12}$$

$$|\lambda_2| \approx |\lambda_3| \tag{2.13}$$

This mathematical model is the central component of the Frangi algorithm (Frangi et al., 1998; Pock et al., 2005; Urschler, 2019).

Multiscale Vessel Enhancement Filtering

The Frangi algorithm was developed by Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken and Max A. Viergever and published in 1998 with a focus on blood vessel segmentation.

Figure 2.13 shows a classical Frangi filter application on a three-dimensional lung CT image.

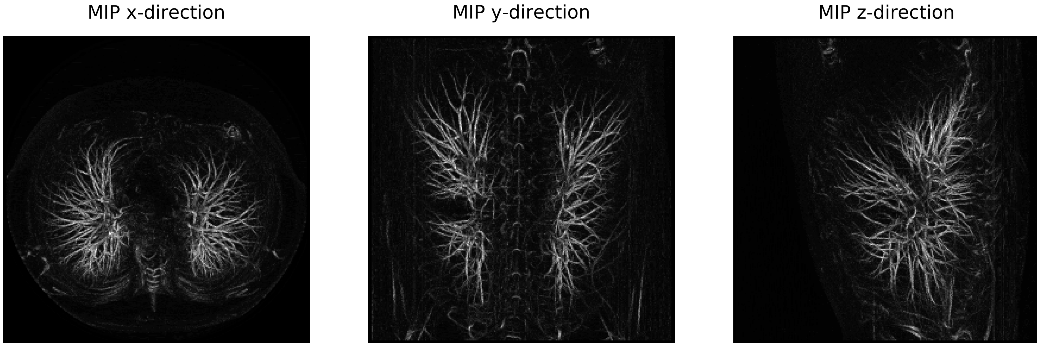


Figure 2.13 Maximum intensity projections of a Frangi filter response applied on a CT image of a human lung.

Using the eigenvalues of the Hessian matrix presented in the previous section, two dissimilarity measures are examined, R_A and R_B and a "second order structureness" S.

Ratio R_A will be close to 1 for tubular shapes because λ_2 will be approximately λ_3. If the structure differs from a tube, λ_2 and λ_3 will be significantly different and the result will be close to 0.

$$R_A = \frac{Largest\ Cross\ Section\ Area/\pi}{(Largest\ Axis\ Semilength)^2} = \frac{|\lambda_2|}{|\lambda_3|} \tag{2.14}$$

R_B evaluates the deviation from a blob-like structure, it will be very close to zero if there is hardly any change of intensities along the tube.

$$R_B = \frac{Volume\frac{4\pi}{3}}{(Largest\ Cross\ Section\ Area/\pi)^{\frac{3}{2}}} = \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}} \tag{2.15}$$

Both ratios are based only on the geometric informations of the image. In the background, the contrast will be low due to the scarcity of structures. The eigenvalues will be small and the "second order structureness" S, based on the Frobenius matrix norm of the eigenvalues, will be low. In regions where there is much contrast compared to the background due to the appearance of structures, the eigenvalues and therefore also S will be higher.

$$S = \sqrt{\sum_{j \leq D} \lambda_j^2} \tag{2.16}$$

In order to assign a high medialness response value R to voxels belonging to a tubular structure, R_A, R_B and S are combined as stated in equation 2.17 and 2.18. The following equation is valid for dark structures on a light coloured background:

$$R = \begin{cases} 0 & \text{if } \lambda_2 < 0 \text{ or } \lambda_3 < 0 \\ (1 - \exp(-\frac{R_A^2}{2\alpha^2}))(\exp(-\frac{R_B^2}{2\beta^2}))(1 - \exp(-\frac{S^2}{2c^2})) & \text{else} \end{cases} \tag{2.17}$$

If the tubular structures are light in contrast to a darker background, the condition changes slightly:

$$R = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0 \\ (1 - \exp(-\frac{R_A^2}{2\alpha^2}))(\exp(-\frac{R_B^2}{2\beta^2}))(1 - \exp(-\frac{S^2}{2c^2})) & \text{else} \end{cases} \tag{2.18}$$

In summary, one could say that R_A, R_B and S are features which are assembled into probability-like estimates of vesselness. The resulting estimation terms are combined in a multiplicative way which makes sure that the response R is large only if all criteria are fulfilled (Frangi et al., 1998; Urschler, 2019).

Frangi Algorithm Formulation and its Parameters

Since the tubular structures one wants to capture will most likely appear at different sizes, a scale space implementation is necessary. In order to achieve this, the image stack is filtered with a Gaussian filter with different standard deviations σ. At the end the maximum medialness response across the scale range is taken (Urschler, 2019) .

```
for all scales do
    for all voxels (x,y,z) do
        Compute Hessian matrix $\nabla^2 I(x, y, z)$
        Compute eigenvalues of Hessian matrix
        Investigate eigenvalues to determine medialness response $R(x, y, z)$
    end
end
Choose maximum medialness response across scale range
```

Algorithm 2: Frangi algorithm.

The parameters α, β and c are thresholds which specify the sensitivity of the filter. α and β are set to 0.5 for many applications and also for the results shown in this document. c it mostly set to the maximum of the norm of the Hessian matrix (Frangi et al., 1998). For the results generated for this thesis, $c = 75$.

Numerical implementation of derivatives

One might ask at this point what it means to talk about derivatives in the context of images. Finite difference methods approximate the differential operator by replacing derivatives with differential quotients. There are forward, backward and central differences. The central difference method approximates the derivative directly at the center voxel and is the most accurate variant, therefore it is used for both first and second order derivatives.

Assuming a simple one-dimensional smooth function u at a point $x \in \mathbb{R}$, one can define its derivative

$$\frac{\partial u}{\partial x} = \lim_{h \to 0} \frac{u(x+h) - u(x-h)}{2h} \tag{2.19}$$

This states the that the exact derivative can be calculated with the quotient on the right hand side if h approaches zero. A fairly good approximation of the derivative can be determined if h is sufficiently small (Frey et al., 2008).

First order derivatives for a three-dimensional signal are computed as follows:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2} \frac{I_{i+1,j,k} - I_{i-1,j,k}}{\Delta x} \tag{2.20}$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2} \frac{I_{i,j+1,k} - I_{i,j-1,k}}{\Delta y} \tag{2.21}$$

$$\frac{\partial I}{\partial z} \approx \frac{1}{2} \frac{I_{i,j,k+1} - I_{i,j,k-1}}{\Delta z} \tag{2.22}$$

Second derivatives are determined in the same way as stated in equations 2.23 and 2.24 for $\frac{\partial I^2}{\partial x \partial x}$ and$\frac{\partial I^2}{\partial x \partial y}$. Again, a central difference scheme is used.

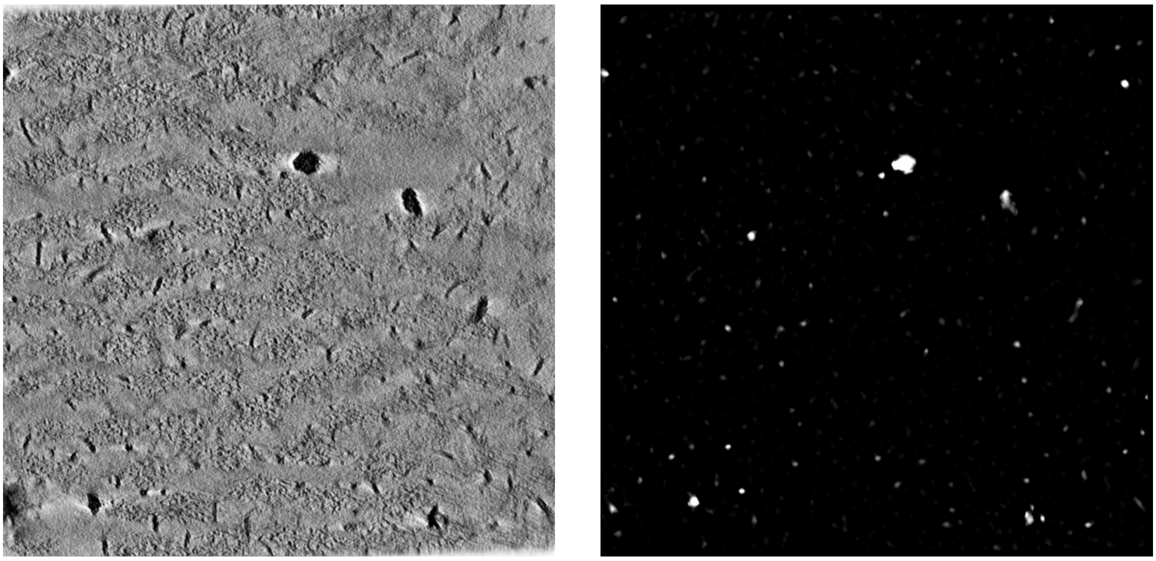
$$\frac{\partial I^2}{\partial x \partial x} \approx \frac{\frac{I_{i+1,j,k} - I_{i,j,k}}{\Delta x} - \frac{I_{i,j,k} - I_{i-1,j,k}}{\Delta x}}{\Delta x} = \frac{I_{i-1,j,k} - 2I_{i,j,k} + I_{i+1,j,k}}{\Delta x^2} \tag{2.23}$$

$$\frac{\partial I^2}{\partial x \partial y} \approx \frac{(\frac{\partial I}{\partial y})_{i+1,j,k} - (\frac{\partial I}{\partial y})_{i-1,j,k}}{\Delta x} = \frac{1}{4} \frac{I_{i+1,j+1,k} - I_{i+1,j,k} - I_{i,j+1,k} + I_{i,j,k}}{\Delta x \Delta y} \tag{2.24}$$

For $\Delta x, \Delta y$ and Δz one voxel is used for the implementation for this thesis.

2.4.2 Results and Discussion

The resolution of the image was decreased to $25 \times 512 \times 512$ voxel to speed up the process. The Frangi filter was applied after previous TV-l1 denoising ($\lambda = 0.9$, 200 iterations).



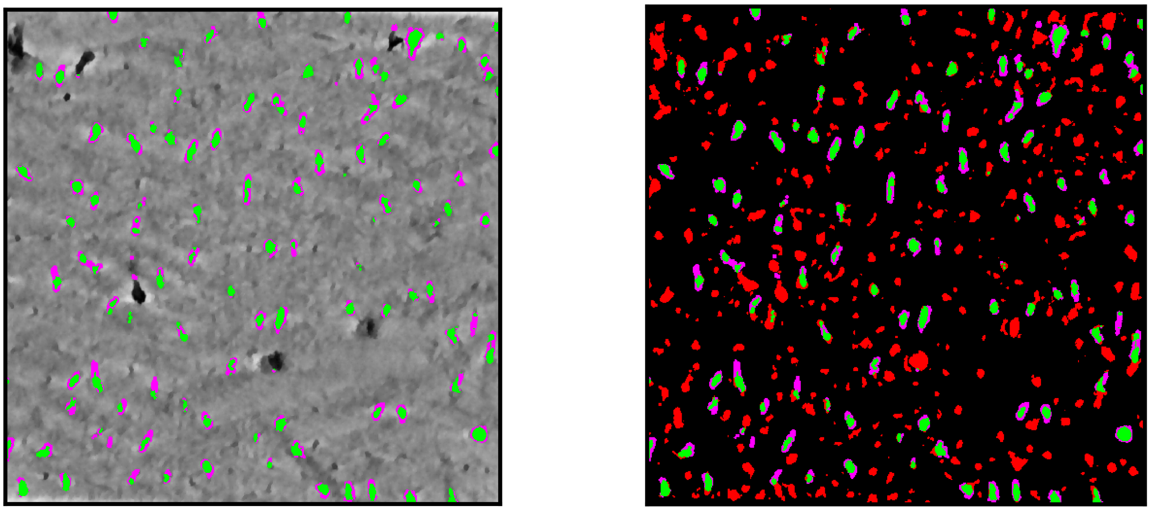
(a) Original image slice

(b) Frangi filter response

Figure 2.14 Application of Frangi filter with $\sigma \in \{1, ...5\}$. Slice of the original image stack (Figure 2.14a) and its corresponding Frangi filter response (Figure 2.14a).

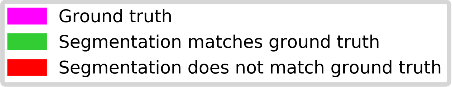
Figure 2.14 shows a typical filter response. The left image of Figure 2.15 suggests that many proteoglycans were detected quite well. However, looking at the right image of Figure 2.15, it is obvious that although many proteoglycans were correctly detected (voxels coloured in green), a significant amount of structures which should not have been considered (voxels coloured in red) were selected.

To reduce the number of falsely detected voxels, a Hysteresis thresholding was applied. For this type of thresholding a low and a high threshold are chosen, t_l and t_h, respectively. A response is considered as a strong response and kept if it is higher than t_h and denoted as too weak if it is below t_l. For responses in between t_l and t_h it is checked if they are located in the neighbourhood of a strong response. If they are, they remain as a response. Figure 2.16 shows that thresholding is highly necessary. Although the Hysteresis thresholding improves the result significantly, a lot of not correctly as proteoglycans detected voxels remain. It was also tried to set the thresholds stricter to get rid of more false positives, but when doing so a lot of correctly detected proteoglycans were lost as well. Another weakness of the algorithm when applied to the proteoglycan datasets was noticed. Figure 2.17 shows the filter response of a ”perfect” tube, which was created by stacking circles on top of each other. One can see that the filter works very well at the center region of the tube. However, one can see a change in response behaviour at the top and at the bottom of the tube. At those regions, the mathematical tubularity assumption does not apply anymore, since it states that intensities do not change along tube direction, whereas the intensities at the top and the bottom of the tube change abruptly. The assignment of eigenvectors to be oriented in-plane or along tube direction is no longer valid at those positions. However, response values at ”problematic” regions are low, the thresholding process eliminates them. Still this effect has to be kept in mind since proteoglycans are rather ”short” and so is their central region where the algorithm works best.



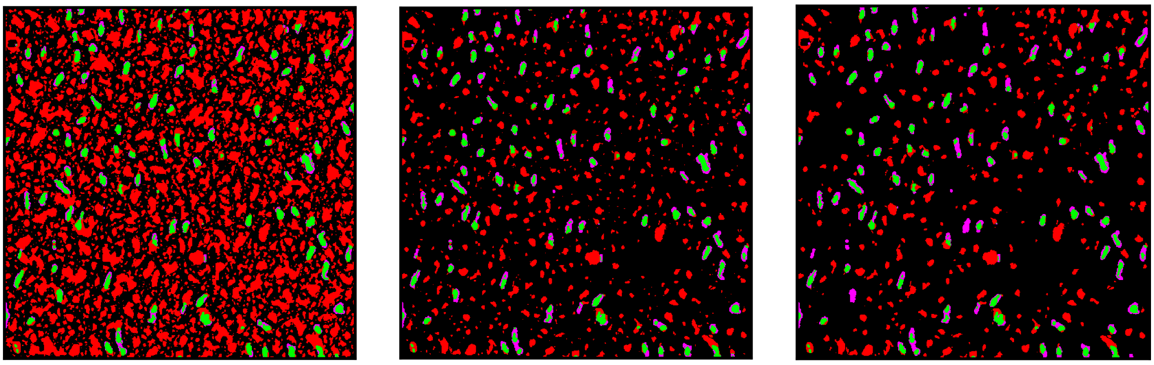
(a) PGs in original image

(b) Detected PGs



(c) Legend

Figure 2.15 Original image with marked Frangi filter response in comparison with ground truth (Figure 2.15a), for better clarity falsely detected proteoglycans are not depicted and a representation of all detected proteoglycans (after Hysteresis thresholding) in comparison with ground truth (Figure 2.15b). The legend in Figure 2.15c explains the displayed colours.



(a) Unprocessed response

(b) Thresholding

(c) Hysteresis thresholding

Figure 2.16 Comparison of unprocessed response (Figure 2.16a), response after simple thresholding (Figure 2.16b) and after Hysteresis thresholding (Figure 2.16c).

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

Figure 2.17 Application of Frangi filter with $\sigma \in \mathbb{Z} : \sigma \in [1, 5]$ on a “perfect tube”. Slice of the tube cut orthonormal to the tube direction (Figure 2.17a) and corresponding Frangi filter response (Figure 2.17b). Slice cut along tube direction (Figure 2.17c) and its Frangi filter response (Figure 2.17d).

(a) (b)

Figure 2.18 Problematic regions where tubularity assumption is not fulfilled are marked by yellow arrows presented on a slice of the “perfect tube” (Figure 2.18a) and its Frangi response (Figure 2.18b).

3 Segmentation of Proteoglycans and Collagen Fibrils based on Machine Learning

Results generated by previously described methods were not sufficiently reliable, so as a next step machine learning models were investigated.

As there was only one single three-dimensional image stack available, it was decided to start with a two dimensional approach. In total, 12 two-dimensional images with a resolution of 2048×2048 pixels with corresponding manually segmented ground truths were used as training and validation data. The images were split into 8 training and 4 validation images.

In the preprocessing step, the resolution of input and target images was decreased by two and split into 4 quadrants. The intensities of the input image were normalized between 0 and 1. Two different models were examined, both were implemented in PyTorch.

3.1 Background

3.1.1 Neural Networks in a Nutshell

Artificial neural networks are inspired by the structure and functioning of the brain. The general purpose of these mathematical models is to analyse data. Their basic entities are artificial neurons. In an artificial neuron of network layer l an input $\mathbf{a}^l$ is weighted by $\mathbf{w}^l$ and a bias $\mathbf{b}^l$ is added (equ. 3.1). The result $\mathbf{y}^{l+1}$ is activated by a non-linear activation function Φ (Eq. 3.2). This activation function enables the model to capture non-linearities.

$$\mathbf{y}^{l+1} = \mathbf{W}^l \mathbf{a}^l + \mathbf{b}^l \tag{3.1}$$

$$\mathbf{a}^{l+1} = \Phi(\mathbf{y}^{l+1}) \tag{3.2}$$

By connecting artificial neurons to each other, a neural network is constructed (Figure 3.1a). In order to determine if the mathematical model generated by the network is a good or a bad one, the output of the last layer has to be designed in a way such that it can be compared to a target. The discrepancy between output and target can then be measured by a so called loss function ℓ, which can be seen as a ”supervision” signal. If the model did not perform well, its parameters $\mathbf{w}^l$ and $\mathbf{b}^l$ have to be altered. This can be done by minimizing the loss of the output and the corresponding target. The gradient $\frac{\partial \ell}{\partial \mathbf{w}^l}$ measures the rate of increase of the loss function ℓ with respect to changes of $\mathbf{w}^l$. This means that (for a small region around $\mathbf{w}^l$) if the loss should be decreased, one has to move in the negative direction of the gradient. Equation 3.3 shows how the weight parameters are updated. τ is needed to control the extent of the adjustment.

$$(\mathbf{w}^l)^{t+1} = (\mathbf{w}^l)^t - \tau \frac{\partial \ell}{\partial (\mathbf{w}^l)^t} \tag{3.3}$$

The same procedure can be applied for the update of bias parameters.

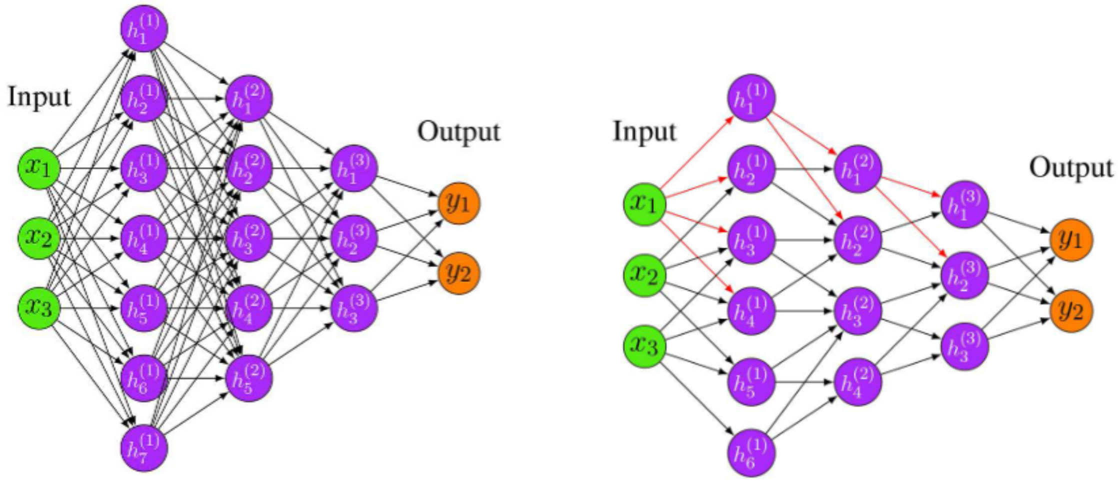
$$(\mathbf{b}^l)^{t+1} = (\mathbf{b}^l)^t - \tau \frac{\partial \ell}{\partial (\mathbf{b}^l)^t} \tag{3.4}$$

For processing images, connecting every neuron of one layer with every neuron of the next layer would lead to a huge amount of learnable (updateable) parameters. In convolutional neural networks a neuron of one layer is therefore only connected to nearby neurons of the next layer, see Figure 3.1b. In a convolutional layer, a convolution is calculated for each input region by sliding a kernel across the input of the convolutional layer. In regions where the kernel overlaps the input, values at the same location at kernel and input are multiplied with each other, the results are summed up to a final scalar value. The mathematical description can be written as given in equation 3.5 with a two-dimensional input feature map $I(i, j)$, a two-dimensional kernel $K(m, n)$ and a two-dimensional result

$C(i, j)$. Strictly speaking, the operation is a cross-correlation and not a convolution.

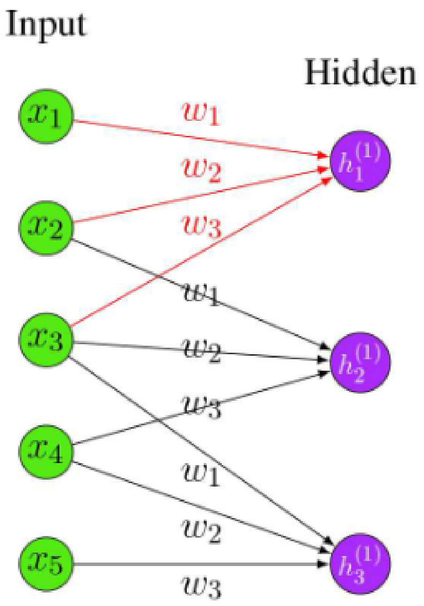
$$C(i, j) = (I * K)(i, j) = \sum_{m} \sum_{n} I(i + m, j + n) K(m, n) \tag{3.5}$$

The kernel shares its weights with each neuron of the layer, see Figure 3.1c for an illustration. Due to this weight-sharing characteristic, the reduction of parameters is not the only advantage of convolutional neural networks. In images, local values are often highly correlated and their statistics invariant to their spatial location. As a kernel shares its weights, patterns from different local regions in the image can be found. By using different kernels, different types of patterns can be extracted. Additionally, pooling layers are integrated into a network to reduce spatial dimensions. They can help to enhance the generalization ability of a network by introducing spatial invariance. Pooling layers do not require and therefore do not learn parameters. In a pooling layer, a pooling window is sliding over the entire input. In the overlapping region of pooling kernel and input, depending on the type of pooling, the average, the maximum or any other chosen value is computed. By doing so, the pooling operator maps all subregions into single numbers (Emmert-Streib et al., 2003; Wu, 2017).



(a) Fully connected neural network. Every neuron is connected to every neuron of the next layer.

(b) Convolutional neural network. Neurons are only connected to some neurons of the next layer.



(c) In a convolutional neural network neurons of one layer share kernel weights .

Figure 3.1 Illustration of fully connected network (Figure 3.1a) and convolutional neural networks (Figure 3.1b and 3.1c). Images taken from Emmert-Streib et al. (2003).

3.1.2 Activation Functions

ReLU (Rectified Linear Unit) and Leaky ReLU activation functions were tested to activate inner layers.

The ReLu function (Equation 3.6) is widely used in the neural network field, its calculations are inexpensive and it shows fast convergence behaviour. However, ReLU units can ”die” during the training process. If, for example, a strong gradient is backpropagated through the network, the weight update can be severely unfavourable and, from that point on, the gradient flowing through this ReLU unit will be constantly 0 because the activation gets ”stuck” in the horizontal plane on the left side where $x < 0$.

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{3.6}$$

The leaky ReLu (Equation 3.7) function attempts to fix this problem of ”dying units” by having a small negative slope instead of a horizontal plane at values for $x < 0$. As a result, gradients will not be zero in those region, they will contribute to the network.

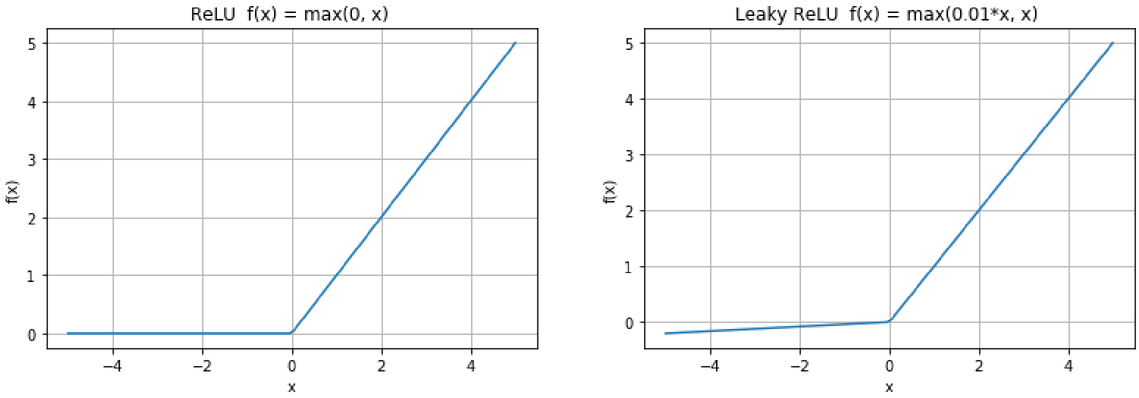
$$f(x) = \begin{cases} \alpha x & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{3.7}$$

It turned out that the leaky ReLU function worked better for both subsequently described networks and was therefore used. α was set to 0.01. The output layer of both networks was activated by a Sigmoid activation function.

The Sigmoid activation function is defined in the following equation:

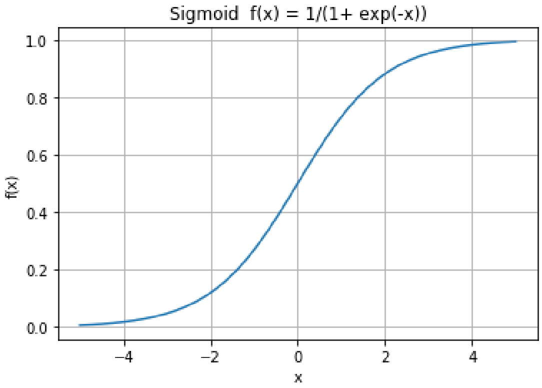
$$f(x) = \frac{1}{1 + \exp(-x)} \tag{3.8}$$

It takes a real-valued number and “squeezes” it into a range between 0 and 1, see figure 3.2c. For a large negative number the denominator will be large and the result therefore 0. For a large positive number the denominator and the result of the function will be 1 (Li et. al., 2016).



(a) ReLU

(b) Leaky ReLU



(c) Sigmoid function

Figure 3.2 Activation functions used and tested for the machine learning models: ReLu (Figure 3.2), Leaky ReLu (Figure 3.2b) and Sigmoid function (Figure 3.2c).

3.1.3 Loss Function

As all target images were binary images, the pixel-wise Sigmoid activation function was used on the final feature map and combined with a binary cross entropy loss.

With target t_c and activated network output p_c for class c, the cross-entropy loss of a pixel ij is defined as follows:

$$\ell_{CE,ij} = -\sum_{c=1}^{C} t_{c,ij} \log(p_{c,ij}) \tag{3.9}$$

For this application only two classes are needed, proteoglycan and non-proteoglycan class, or collagen and non-collagen class and the expression can be simplified:

$$\ell_{CE,ij} = -[t_{c,ij}\ \log(\sigma(s_{c,ij})) + (1 - t_{c,ij})\ \log(1 - \sigma(s_{c,ij}))] \tag{3.10}$$

One can observe that activating the last layer with a Sigmoid function is necessary in order to have reasonable values for $p_c = \sigma(s_c)$ between 0 and 1. The logarithm of $\sigma(s_c)$ will be close to 0 if $\sigma(s_c)$ is close to 1, and it will be approximating - ∞ if $\sigma(s_c)$ is close to 0. Since t_c is either 0 or 1, the loss will be low if target and estimated output are similar and large otherwise.

The loss is calculated per pixel. In order to get a final scalar value, the loss is averaged over all pixels.

3.1.4 Optimization

For both implemented networks, the Adaptive Moment Estimation (Adam) optimizer was used. Intuitively one can think of the optimizer as a heavy ball with friction.

Gradient descent optimization algorithms minimize an objective function (loss function) $J(\theta)$, where $\theta \in \mathbb{R}^d$ are the parameters of the model. As previously described, by updating the parameters θ in the negative direction of the gradient $\nabla_\theta J(\theta)$ one can follow the slope of the surface of the objective function "downwards" until a valley is reached.

If the "standard" gradient descent algorithm is used, updates are performed based on the gradient of the loss of the entire training dataset. Stochastic gradient descent updates the model parameters by determining the gradient of the loss of each individual training sample.

If momentum is added to the standard stochastic gradient method, a fraction γ of the gradient at the previous time step $t-1$ is added to the update of the current time step t:

$$v_t = \gamma\, v_{t-1} + \tau \nabla_\theta J(\theta) \tag{3.11}$$

$$\theta = \theta - v_t \tag{3.12}$$

Intuitively one can think of a ball, as it rolls downhill it will become faster along its way because it accumulates momentum. The parameter updates behave similarly. For dimensions whose gradients are oriented in the same direction, the momentum increases and severe updates are made. In the contrary, if gradients change directions, only slight updates are performed.

The Adam optimizer relies on an exponentially decaying average of past gradients m_t, similar to momentum:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla_\theta J(\theta_t) \tag{3.13}$$

Additionally an exponentially decaying average of previously determined squared gradients v_t is used:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta J(\theta_t))^2 \tag{3.14}$$

As m_t and v_t are biased towards 0, they need to be bias-corrected.

$$\tilde{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.15}$$

$$\tilde{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.16}$$

The Adam update rule can now be written as:

$$\theta_{t+1} = \theta_t - \frac{\tau}{\sqrt{\tilde{v}_t} + \epsilon}\tilde{m}_t \tag{3.17}$$

Intuitively one can think of the algorithm as a ball with friction: it will overshoot local minima but once it reaches a flat minimum it will remain there (Ruder, 2016).

3.1.5 Instance Normalization

The performance of both networks was significantly improved by adding normalization layers. Layer normalization and instance normalization layers were applied and the obtained results compared. Batch normalization would not have made sense since batches of size 1 were used. Instance normalization led to the best results for both networks.

Assuming that $\mathbf{x} \in \mathbb{R}^{N \times C \times W \times H}$ is an input tensor, with N batched images, C feature channels and a spatial resolution $W \times H$, then the $tijk$-th element can be denoted as x_{tijk}.

t is the index of the image in the batch, i is the feature channel and j and k are the indices of the spatial dimensions.

Mean μ_{ti} and variance σ_{ti} are computed for each sample and for each channel across both spatial dimensions (Wu et al., 2018):

$$\mu_{ti} = \frac{1}{HW} \sum_{l=1}^{W} \sum_{m=1}^{H} x_{tilm} \tag{3.18}$$

$$\sigma_{ti}^2 = \frac{1}{HW} \sum_{l=1}^{W} \sum_{m=1}^{H} (x_{tilm} - \mu_{ti})^2 \tag{3.19}$$

Using both one can compute a normalized version of the $tijk$ -th element of $\mathbf{x}$:

$$\hat{x}_{tijk} = \frac{x_{tijk} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \epsilon}} \tag{3.20}$$

ϵ is added to the denominator to increase numerical stability. Instance normalization is applied during training and validation (Ulyanov et al., 2016). Figure 3.3 shows a very nice illustration of different normalization strategies.

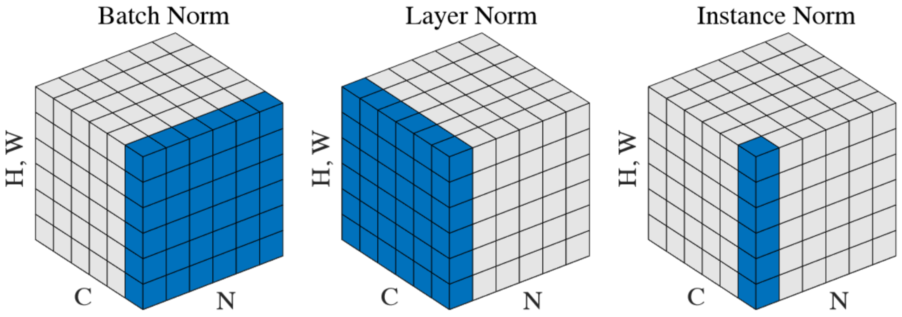
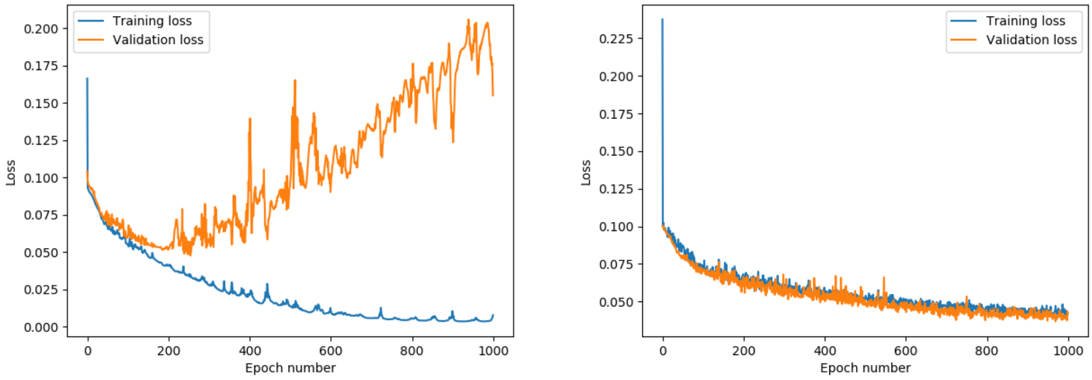


Figure 3.3 Illustration of mentioned normalization methods. H and W are spatial axes, C is the channel axis and N the batch axis of feature map tensors (Wu et al., 2018).

3.1.6 Data Augmentation

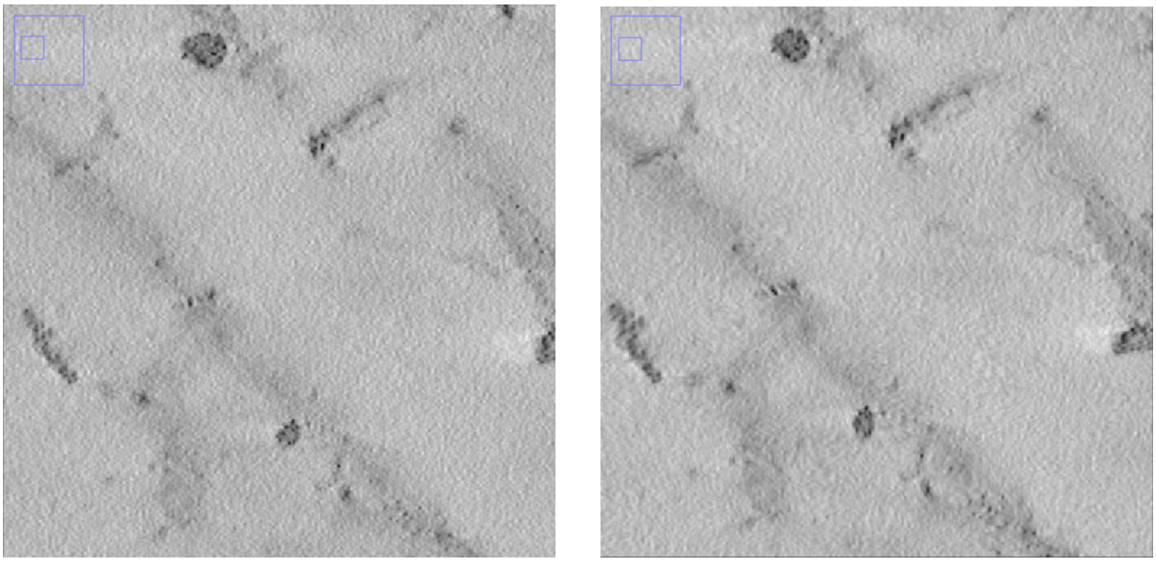
Due to the small amount of training data, severe overfitting was happening for both networks. Hence, data augmentation was crucial. Figure 3.4 shows the significant improvement of the learning behaviour of the U-Net model very nicely. Rotation, shifting, flipping and elastic deformation were implemented. Simard et al. (2003) suggest to implement random elastic deformation by generating displacement vector fields Δx and Δy which are chosen randomly from a Uniform distribution between -1 and 1. Δx and Δy are convolved with a Gaussian distribution with standard deviation σ. Setting σ to a relatively high value ensures that displacements remain rather low because the random value will average around 0. In order to control the extent of deformation, the displacement fields are multiplied with a constant parameter α. For this thesis, α was randomly chosen between 20 and 100, σ was set to 10.



(a) Loss without data augmentation

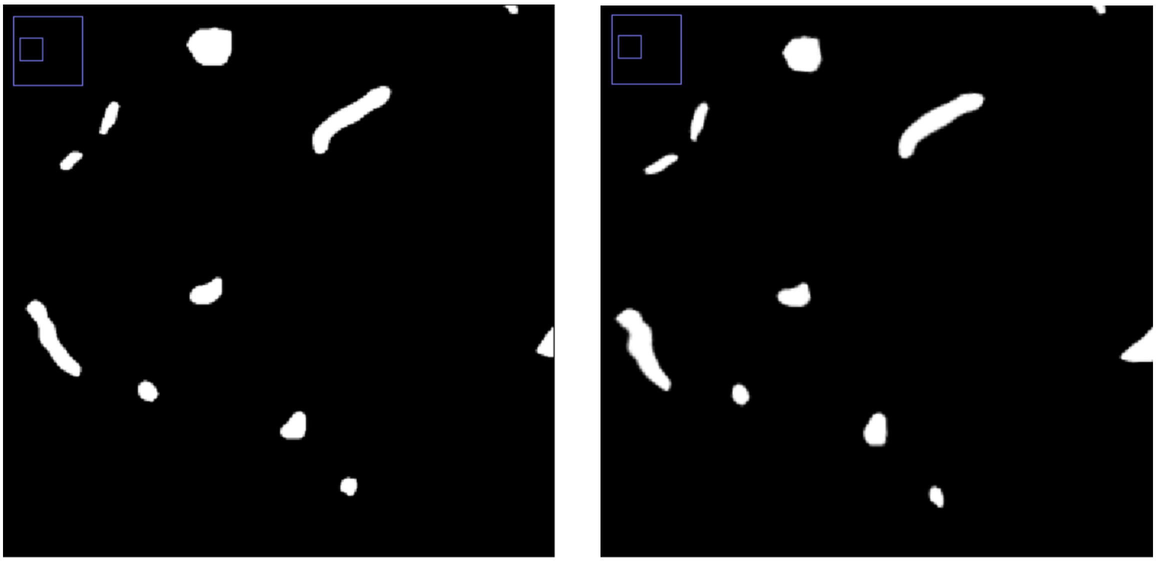
(b) Loss with implemented data augmentation

Figure 3.4 U-Net loss showing the impact of data augmentation by comparison between the loss without (Figure 3.4a) and with implemented data augmentation (Figure 3.4b) .



(a) Original input

(b) Deformed input



(c) Proteoglycan target

(d) Deformed proteoglycan target

Figure 3.5 Input image (Figure 3.5a) and corresponding proteoglycan target image (Figure 3.5b) and their elastically deformed versions (Figure 3.5c and 3.5d).

3.2 U-Net Model

3.2.1 Network Architecture and Parameters

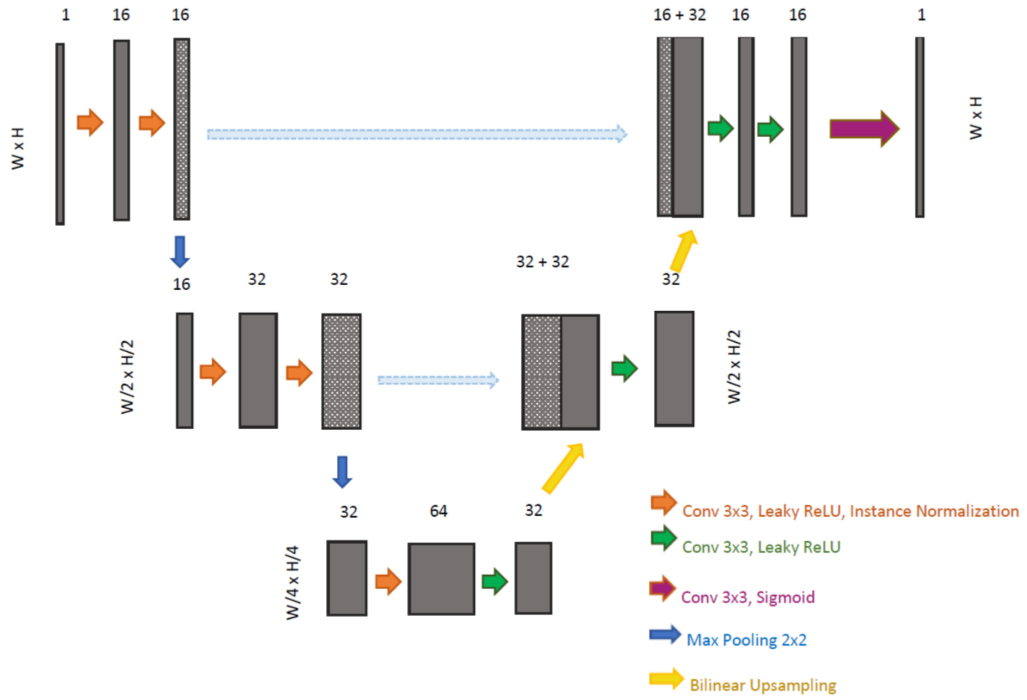


Figure 3.6 Illustration of network architecture, adapted from Figure 1 in Ronneberger et al. (2015).

The first network used for this thesis is based on the U-Net architecture developed by Ronneberger et al. (2015) and its modification for feature extraction used by Knöbelreiter et al. (2015), see Figure 3.6 for an illustrative representation.

It consists of a contracting path and an expansive path. In the contracting path two convolutions each followed by a Leaky ReLU activation function are applied repeatedly. The first convolution of each ”spatial resolution level” is doubling the amount of feature channels. A max pooling layer after the activation of the second convolution is used for downsampling to reach the ”spatial resolution level” below. The expansive path is composed of upsampling layers followed by two convolutions and Leaky ReLu activations. The first convolution is applied on the concatenated channels of the upsampled layer and the corresponding layer of the contracting path (Ronneberger et al., 2015).

Table 3.1 Network architecture.

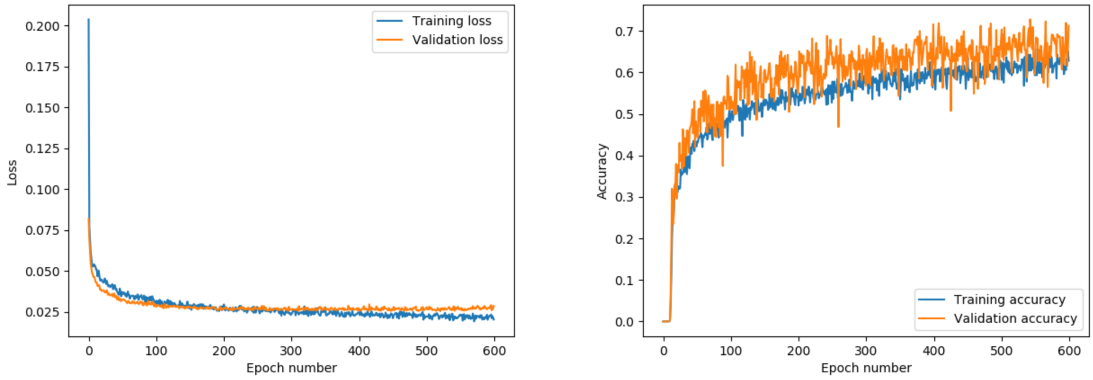
Layer	**KS**	**Resolution**	**Channels**	**Input**	**Activation fct.**
conv00	3	$W \times H/W \times H$	1/16	Image	Leaky ReLU
norm00	-	$W \times H/W \times H$	16/16	conv00	-
conv01	3	$W \times H/W \times H$	16/16	norm00	Leaky ReLU
pool0	2	$W \times H/\frac{W}{2} \times \frac{H}{2}$	16/16	conv01	-
norm01	-	$\frac{W}{2} \times \frac{H}{2}$ / $\frac{W}{2} \times \frac{H}{2}$	16/16	pool0	-
conv10	3	$\frac{W}{2} \times \frac{H}{2}$ / $\frac{W}{2} \times \frac{H}{2}$	16/32	norm01	Leaky ReLU
norm10	-	$\frac{W}{2} \times \frac{H}{2}$ / $\frac{W}{2} \times \frac{H}{2}$	32/32	conv10	-
conv11	3	$\frac{W}{2} \times \frac{H}{2}$ / $\frac{W}{2} \times \frac{H}{2}$	32/32	conv10	Leaky ReLU
pool1	2	$\frac{W}{2} \times \frac{H}{2}$ / $\frac{W}{4} \times \frac{H}{4}$	32/32	conv10	-
norm01	-	$\frac{W}{4} \times \frac{H}{4}$ / $\frac{W}{4} \times \frac{H}{4}$	32/32	pool1	-
conv20	3	$\frac{W}{4} \times \frac{H}{4}$ / $\frac{W}{4} \times \frac{H}{4}$	32/64	norm01	Leaky ReLU
norm20	-	$\frac{W}{4} \times \frac{H}{4}$ / $\frac{W}{4} \times \frac{H}{4}$	64/64	conv20	-
conv21	3	$\frac{W}{4} \times \frac{H}{4}$ / $\frac{W}{4} \times \frac{H}{4}$	64/32	conv20	Leaky ReLU
bilin1	-	$\frac{W}{4} \times \frac{H}{4}$ / $\frac{W}{2} \times \frac{H}{2}$	32/32	conv21	-
conv12	3	$\frac{W}{2} \times \frac{H}{2}$ / $\frac{W}{2} \times \frac{H}{2}$	64/32	{bilin1, conv11}	Leaky ReLU
conv13	3	$\frac{W}{2} \times \frac{H}{2}$ / $\frac{W}{2} \times \frac{H}{2}$	32/16	conv12	Leaky ReLU
bilin0	-	$\frac{W}{2} \times \frac{H}{2}$ /$W \times H$	32/32	conv12	-
conv02	3	$W \times H/W \times H$	48/16	{bilin0, conv01 }	Leaky ReLU
conv03	3	$W \times H/W \times H$	16/16	conv02	Leaky ReLU
output	3	$W \times H/W \times H$	16/1	conv03	Sigmoid

Table 3.1 shows the exact parameters of the network. $W = 512$ and $H = 512$ since the resolution of the two-dimensional images was decreased by two and the images were split into quadrants.

A sigmoid function is applied on the last convolutional layer which has one channel per pixel. In order to compare the result with the binary target to compute the accuracy and IoU, values greater than 0.5 are set to 1, all others are set to 0. The computation of accuracy and IoU are described in detail in chapter 4.

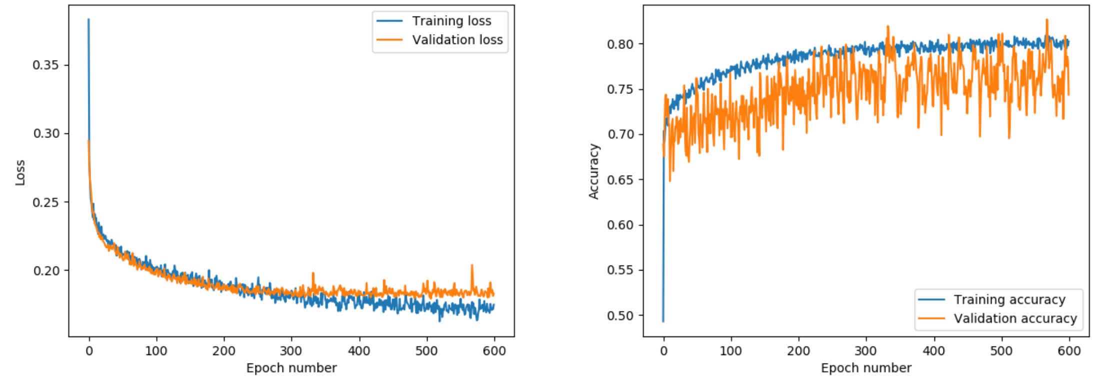
Additionally, layer normalization and instance normalization layers were added at different positions, e.g. before and after the activation function was tested. Instance normalization layers after the Leaky ReLU function in the contracting path turned out to lead to highest accuracies. The Adam Optimizer with a learning rate of 0.0004 was used for proteoglycan and collagen fibril segmentation trainings. Weights were initialized according to a Gaussian distribution with 0 mean and a standard deviation of $\sqrt{\frac{2}{N}}$. N is the number of incoming nodes of one neuron, hence $N = kernelsize * kernelsize * number\,of\,input\,channels$ (Ronneberger et al., 2015). Convolution bias was not used.

3.2.2 Results and Discussion



(a) (b)

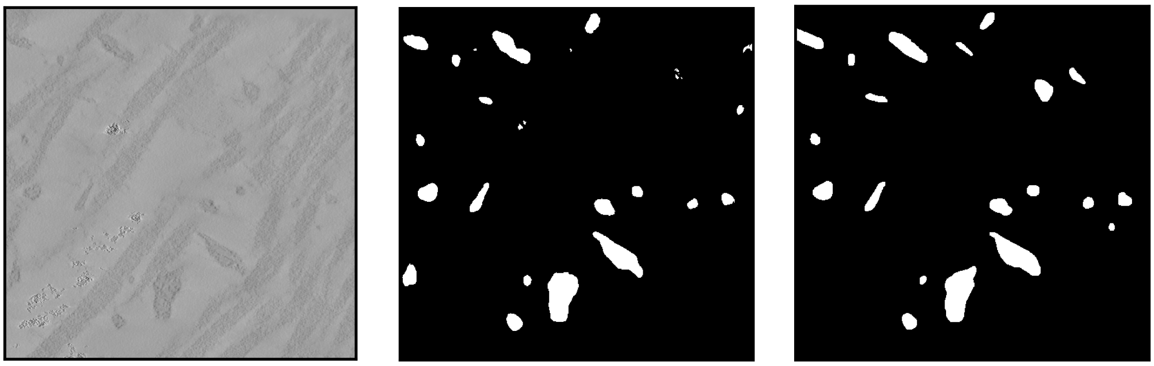
Figure 3.7 Training behaviour with data augmentation and instance normalization for the proteoglycan segmentation, shown by loss (Figure 3.7a) and accuracy (Figure 3.7b) during training.



(a) (b)

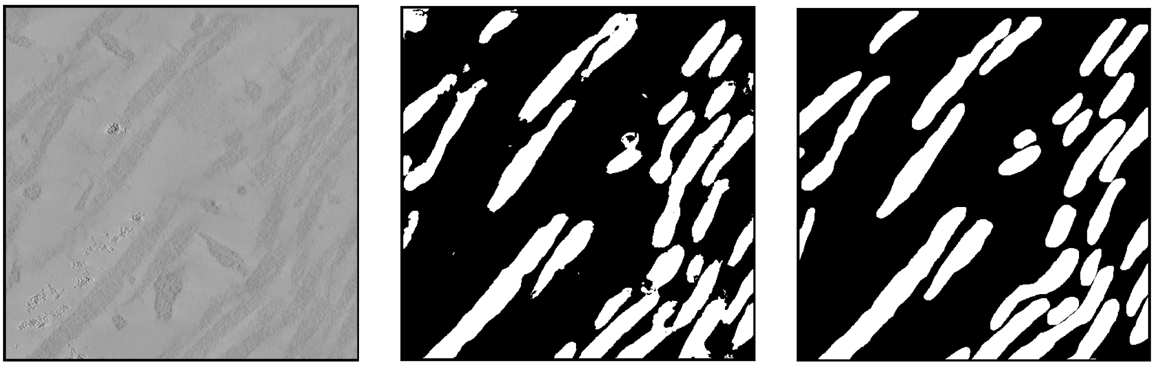
Figure 3.8 Training behaviour with data augmentation and instance normalization for the collagen fibril segmentation, shown by loss (Figure 3.8a) and accuracy (Figure 3.8b) during training.

Figure 3.7 shows the behaviour of the U-Net based model during training with augmented training data and additional instance normalization for the segmentation of proteoglycans. It can be seen that the accuracy increased rapidly at the beginning and remained between 0.6 and 0.7 at higher epoch numbers. The model with the highest accuracy (0.72) was saved. Figure 3.8 shows the training of the U-Net model for the collagen fibril segmentation. Again, the accuracy improved significantly at low epoch numbers and remained around 0.8. The best model with an accuracy of 0.82 was saved. Figures 3.9 and 3.10 show an example of both segmentations.



(a) Input (b) Output (c) Target

Figure 3.9 U-Net model segmentation of proteoglycans. The trained U-Net model is applied on a validation input image (Figure 3.9a). Its output (Figure 3.9b) can be compared with the target segmentation mask (Figure 3.9c).



(a) Input (b) Output (c) Target

Figure 3.10 U-Net model segmentation of collagen fibrils. The trained network is applied on a validation input image (Figure 3.10a). Its segmentation output (Figure 3.10b) can be compared with the target segmentation mask (Figure 3.10c).

3.3 ESPNet Model

3.3.1 Network Architecture and Parameters

The second Machine Learning approach is based on the ESPNet (Efficient Spatial Pyramid) developed by Mehta et al. (2018).

A standard convolutional layer takes $\mathbf{F}_i \in \mathbb{R}^{W \times H \times M}$ as an input feature map and $\mathbf{F}_o \in \mathbb{R}^{W \times H \times N}$ is the resulting output feature map. W and H are the width and the height of the feature maps, M is the number of input feature channels, N the number of output feature channels.

The ESPNet uses ESP modules, which decompose standard convolutions into two steps: a computation reducing point-wise convolution and a spatial pyramid of dilated convolutions which re-samples the feature maps to learn representations from receptive fields of different sizes, see Figure 3.11.

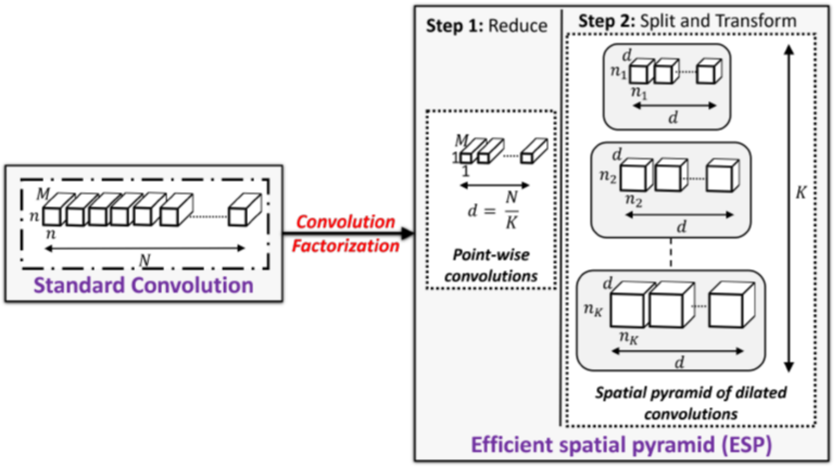


Figure 3.11 Decomposition of a standard convolution (Mehta et al., 2018).

The computation, power and memory efficient ESP module relies on the following strategy:

- *Reduce*: To shrink the dimensionality of the feature maps uniformly, a divider parameter K is introduced. For a given divider K, the ESP module reduces the feature maps from M-dimensional space to a $\frac{N}{K}$- dimensional space by using a point-wise convolution with a 1 × 1 kernel.

- *Split*: The resulting low-dimensional feature maps are then split across K parallel branches.
- *Transform*: In each branch these low-dimensional feature maps are then processed simultaneously with equally sized kernels but different dilation rates. For each kernel, the same number of pixels participate, but they cover K different receptive fields. If $n \times n$ pixels are considered, the effective spatial dimension of a dilated kernel is $n_k \times n_k$ with $n_k = (n-1)2^{(k-1)} + 1$, $k = 1, ..., K$.
- *Merge*: In the last strategy step, the output of all K parallel branches are concatenated in order to get an N-dimensional output feature map.

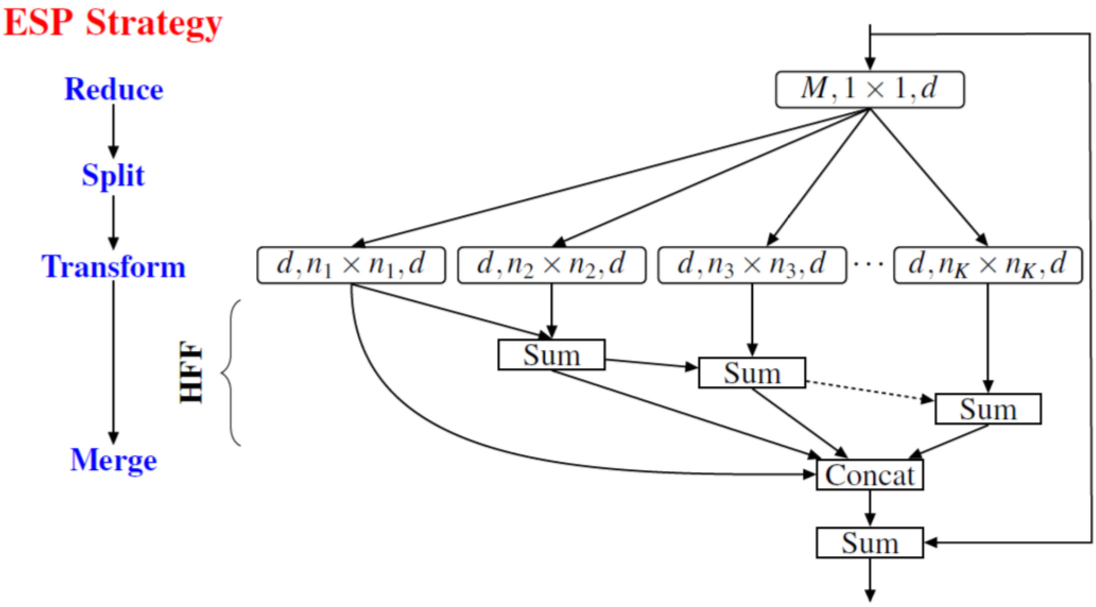


Figure 3.12 Illustration of the ESPmodule strategy and the HFF (hierarchical feature fusion) (Mehta et al., 2018).

Simply concatenating all branches might lead to unwanted gridding artefacts, therefore Mehta et al. (2018) suggest to add the outputs of differently dilated kernels hierarchically before concatenating them (HFF, hierarchical feature fusion). Branches with a larger receptive field are subsequently added to previously joined branches with smaller receptive fields.

Figure 3.12 shows the strategy of the ESPmodule in a block diagram. Figure 3.13 shows the architecture of the ESP-Net. It consists of an encoder path (depicted on the left side) and a decoder path (depicted on the right side). For down-sampling, strided ESP modules are used. The hyperparameter α is needed to control the depth of the network. At spatial level l, the ESP module is repeated α_l times. At low levels ($l = 0, l = 1$) a lot of memory is required because feature maps have high spatial dimensions. To keep the network memory efficient, ESP and convolutional modules are not repeated at these levels. Feature maps of the previous strided down-sampling ESP module are concatenated with the previous ESP module to improves the flow of information (depicted as curved arrow in Figure 3.13). To compensate the loss of spatial information due to down-sampling and convolutions, a down-sampled version of the input image is additionally concatenated at those network positions. The decoder path is based on a *reduce-upsample-merge* (RUM) principle, which make it possible to use the information at every spatial level. The *reduce* - step performs a projection of feature maps of spatial levels l and $l - 1$ to a C-dimensional space. C represents the number of classes in the dataset. As the Sigmoid function is used to activate the last layer of the network, $C = 1$ for this application. To match the spatial dimensions of the feature map at spatial level $l - 1$, the feature map of the layer below, l, has to be upsampled by a sampling factor of 2. This is achieved by applying a deconvolution kernel with kernel size 2×2 at the *reduce* step. The final *merge* - process is a concatenation of the previously upsampled feature map from spatial level l with the C-dimensional feature map of spatial level $l - 1$. The RUM principle is repeated until spatial dimension $l = 0$ is reached and the spatial dimensions of the final feature map are the same as the input image dimensions (Mehta et al., 2018). A Sigmoid activation function is applied on the final feature map to compute a binary segmentation mask. Mehta et al. (2018) used PReLU acitvation functions and batch normalization for inner layers, for this thesis Leaky ReLU and instance normalization turned out to work better. Divider parameter K is 5, depth hyperparameter α_2 was set to 2, α_3 was set to 8. The Adam Optimizer with a learning rate of 0.001 was used for proteoglycan and collagen fibril segmentation trainings.

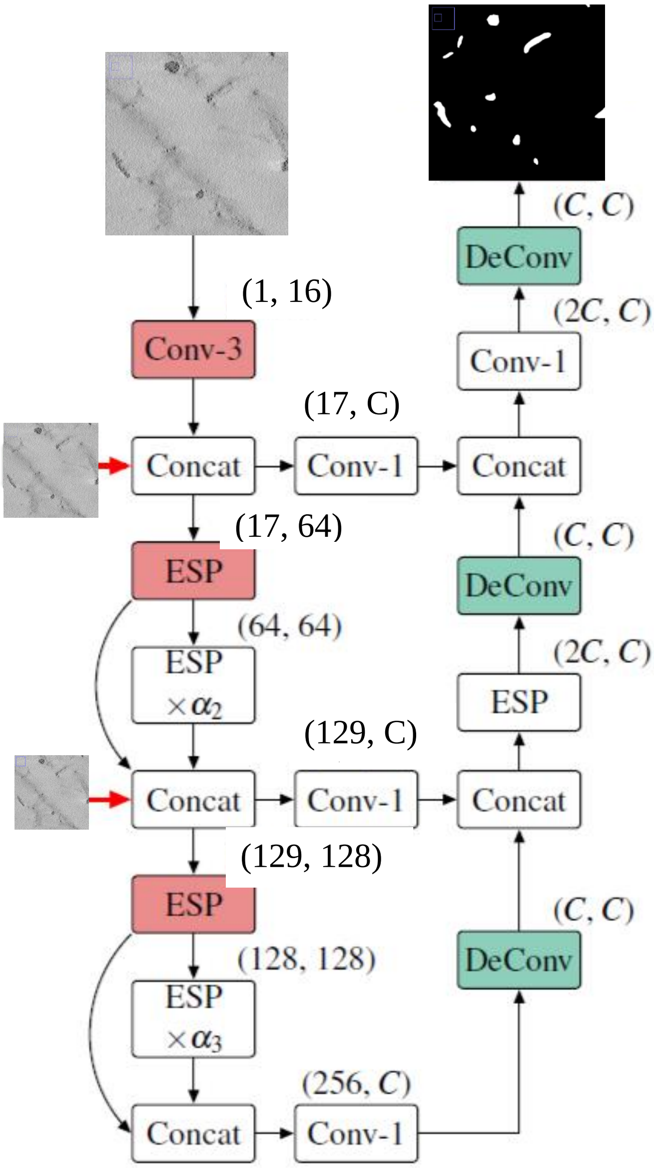
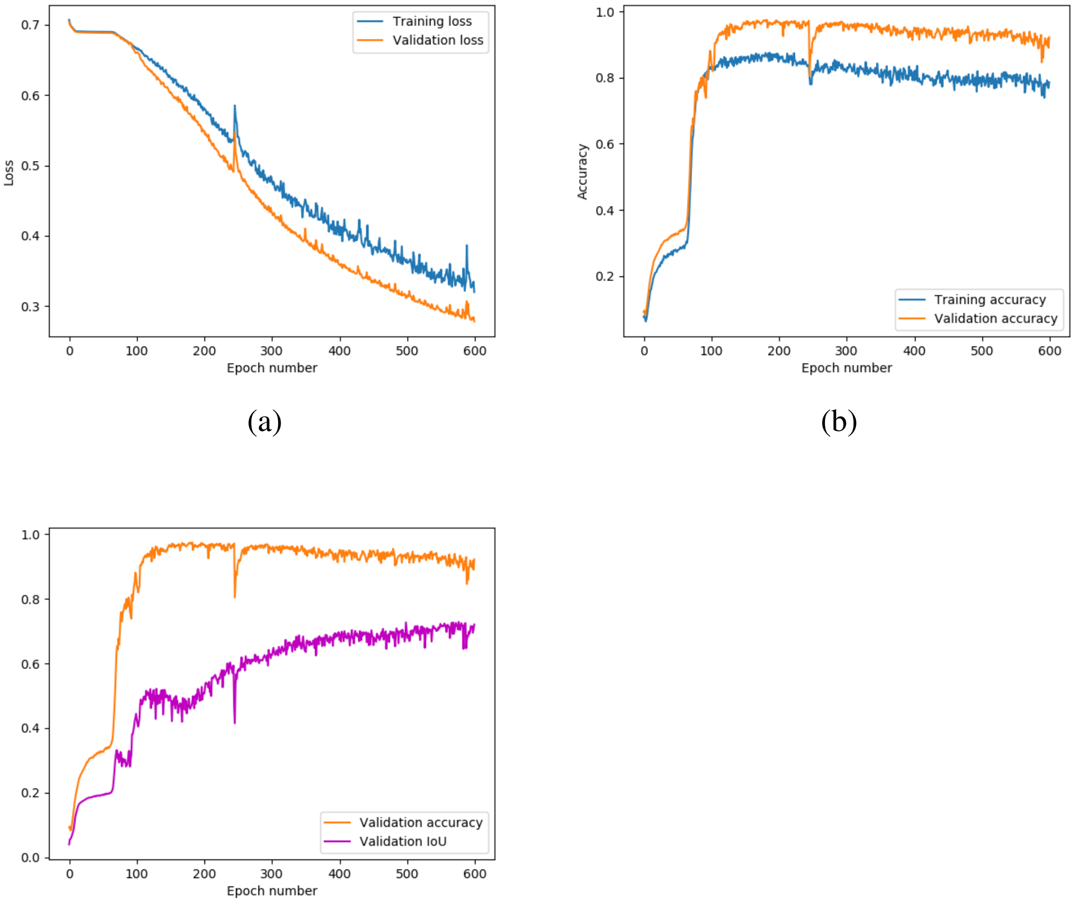


Figure 3.13 Illustration of the ESPNet architecture, adapted from Mehta et al. (2018). $C = 1$, since at the end a one channel output layer is activated by a Sigmoid function. Conv-n means that a convolution with an $n \times n$ kernel is performed. Adapted from Mehta et al. (2018)

3.3.2 Results and Discussion



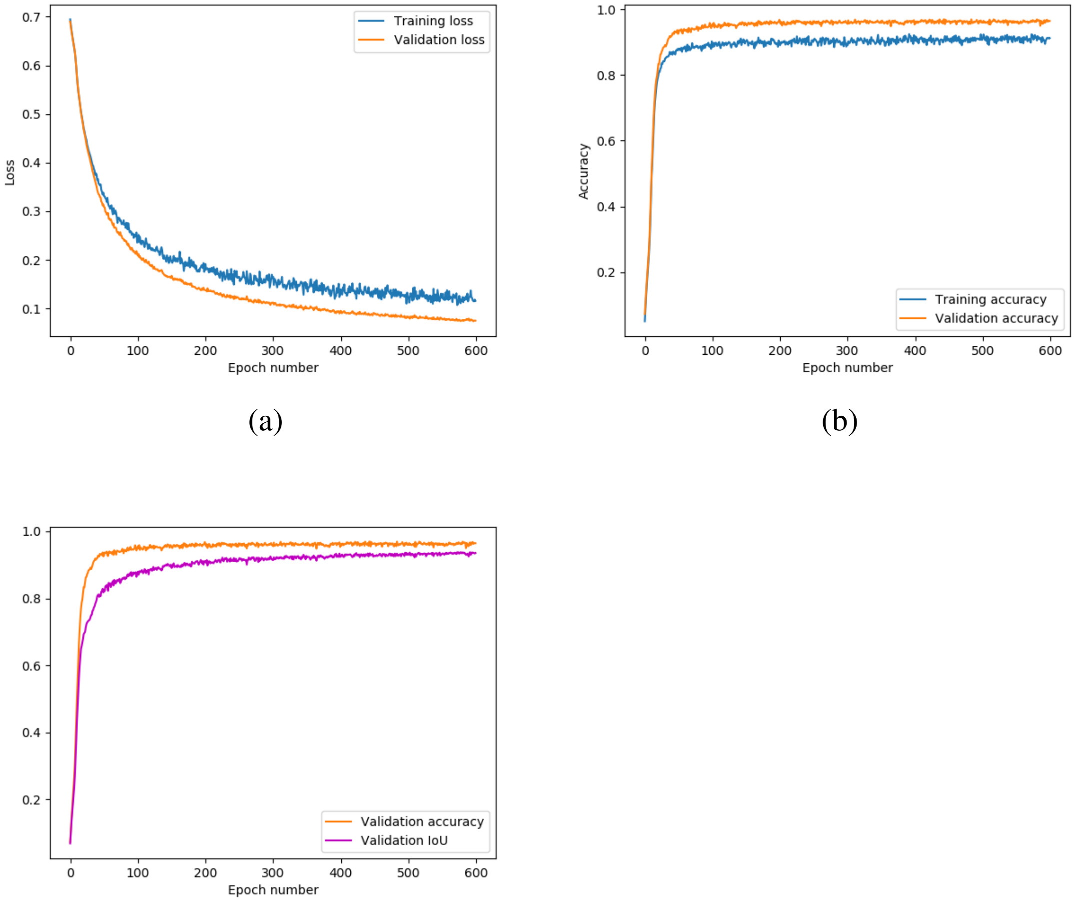
(a)

(b)

(c)

Figure 3.14 Training behaviour with data augmentation and instance normalization for the proteoglycan segmentation, shown by loss (Figure 3.14a), accuracy (Figure 3.14b) and the comparison of validation accuracy and IoU (Figure 3.14c) during training.

For the training of the segmentation model for proteoglycans, it was not sufficient to only consider the accuracy as an indicator. At low iteration numbers a high accuracy was measured although the IoU remained low (see Figure 3.14), therefore a model was only considered to be ”good enough” if both accuracy and IoU increased during a training epoch. The accuracy increased rapidly at low epoch numbers but starts to decrease again at approximately 100 epochs, while the IoU increases continously. The training process for the segmentation collagen fibrils behaves more evenly, both IoU and accuracy improve with more epochs. Figure 3.16 shows one example for a proteoglycan segmentation. The borders of the segmented structures are slightly ”edgy”, it can be assumed that this is due to the upsampling behaviour of the ESPNet.

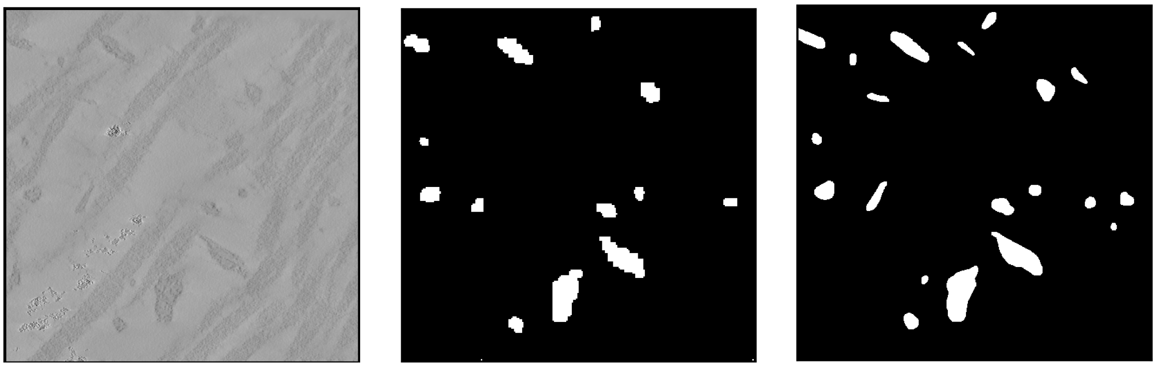


(a)

(b)

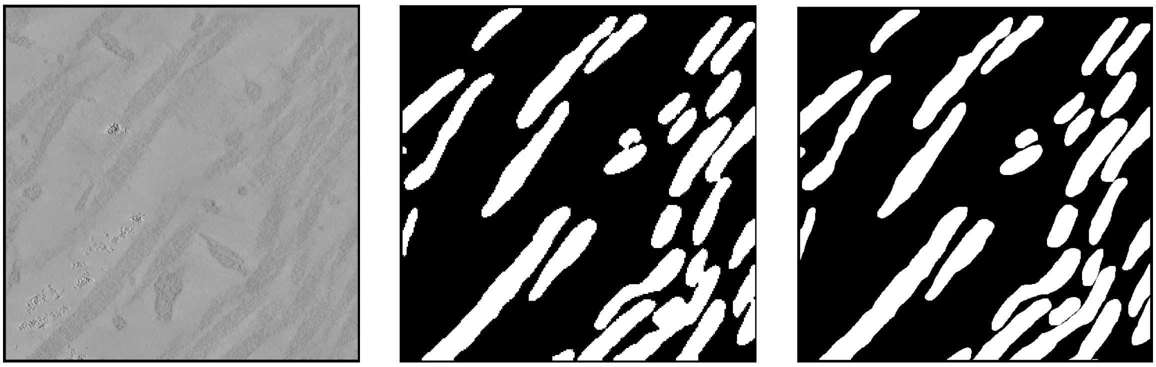
(c)

Figure 3.15 Training behaviour with data augmentation and instance normalization for the collagen fibril segmentation, shown by loss (Figure 3.15a), accuracy (Figure 3.15b) and the comparison of validation accuracy and IoU (Figure 3.15c) during training



(a) Input (b) Output (c) Target

Figure 3.16 ESPNet model segmentation of proteoglycans. The trained Network is applied on a validation input image (Figure 3.16a). Its segmentation output (Figure 3.16b) can be compared with the target segmentation mask (Figure 3.16c).



(a) Input (b) Output (c) Target

Figure 3.17 ESPNet model segmentation of collagen fibrils. The trained ESPNet model is applied on a validation input image (Figure 3.17a). Its segmentation output (Figure 3.17b) can be compared with the target segmentation mask (Figure 3.17c).

4 Comparison of different Approaches

4.1 Methods

4.1.1 Evaluation Metrics

In order to evaluate and compare different segmentation results, accuracy and intersection over union were computed for datasets where a manually segmented ground truth was available.

Accuracy

The accuracy is the average number of correctly classified voxels. $U \in \mathbb{R}^{M\times N\times J}$ is assumed to be the binary output of the segmentation. Voxels, which are part of a segmented structure have an intensity value equal to one. If not, their intensity is zero. $T \in \mathbb{R}^{M\times N\times J}$ is the manually segmented ground truth. Segmented voxels are labelled with one, others with zero. m is the total amount of segmented voxels in the ground truth image stack, so it is the number of voxels with an intensity value equal to one.

The accuracy is the percentage of correctly classified voxels and is computed as given in equation (4.1).

$$\mathcal{A} = \frac{1}{m}\sum_{i,j,k}\tilde{\delta}(U_{i,j,k},\, T_{i,j,k}) \text{ with } \tilde{\delta}(x,y) = \begin{cases} 1 & \text{if } x = y = 1 \\ 0 & \text{if } x \neq y \\ 0 & \text{if } x = y = 0 \end{cases} \tag{4.1}$$

Intersection over Union (IoU)

Additionally to the accuracy, the intersection-over-union is determined too. The intersection-over-union is the overlapping volume of segmentation result and ground truth divided by the volume of union between result and ground truth, see equation 4.2.

$$IoU = \frac{|U \cap T|}{|U \cup T|} \tag{4.2}$$

Again, $U \in \mathbb{R}^{M \times N \times J}$ is the binary output of the algorithm and $T \in \mathbb{R}^{M \times N \times J}$ is the manually segmented ground truth, segmented voxels have intensity value one, all other voxels are set to zero.

For a perfect segmentation, the segmented result matches exactly the ground truth. Their intersection is equal to their union and the determined intersection over union equal to one. If the segmentation involves many falsely classified voxels, the union of T and U would be rather large and as a result the intersection over union close to zero.

(a) Ground truth T. (b) Result U. (c) Intersection $|U \cap T|$. (d) Union $|U \cup T|$.

Figure 4.1 Demonstration of intersection-over-union for an illustrative structure. If Figure 4.1a and 4.1b are assumed to be ground truth T and result U, respectively, Figure 4.1c depicts their intersection $|U \cap T|$ and Figure 4.1d their union $|U \cup T|$.

4.2 Results and Discussion

4.2.1 Comparison of Proteoglycan Segmentation

Table 4.1 compares thresholding, the Frangi tube detection algorithm and both implemented Machine Learning approaches for the segmentation of proteoglycans. For the thresholding approach and the Frangi algorithm, values for accuracy and IoU were calculated based on one labelled 3D dataset. For both Machine Learning models, values of all 4×4 validation quadrant images were averaged. Both accuracy $\mathcal{A}_P$ and intersection over union IoU_P were calculated per-proteoglycan class.

Table 4.1 Comparison of per-proteoglycan class accuracy and IoU on validation set.

Method	**Validation set dimension**	$\mathcal{A}_P$	IoU_P
Thresholding	3D	0.4131	0.3018
Frangi	3D	0.5227	0.2025
U-Net Model	2D	0.7279	0.5961
ESPNet Model	2D	0.9669	0.6258

The machine learning models performed significantly better than the “classical” approaches, considering both accuracy and IoU. The ESPNet model delivered even better results than the U-Net model.

4.2.2 Comparison of Collagen Fibril Segmentation

Table 4.2 compares the implemented Machine Learning approaches for the segmentation of collagen fibrils. Values of all 4×4 validation quadrant images were averaged. Accuracy $\mathcal{A}_C$ and intersection IoU_C were calculated per-collagen-fibril class.

Table 4.2 Comparison of per-collagen-fibril class accuracy $\mathcal{A}_C$ and intersection over union IoU_C on validation set.

Method	**Validation set dimension**	$\mathcal{A}_C$	IoU_C
U-Net Model	2D	0.8269	0.7086
ESPNet Model	2D	0.9511	0.9048

Both machine learning models achieved good results. Again, the ESP-Net is performing significantly better. Compared to the segmentation of proteoglycans, better values for accuracy and IoU were achieved. There are more collagen fibril voxels than proteoglycan voxels in the images, the networks had more information to learn from.

5 Orientation and Diameter Evaluation of Proteoglycans

The best segmentation result obtained with the ESP-Net model was then used to evaluate orientation and diameter of segmented proteoglycans. The procedure is not done for collagen fibrils since most of them are only partially captured in the electron microscope image.

The output of the ESPNet model shows “edgy” boundaries due to the down- and upsampling character of the network’s architecture, see Figure 5.1. Additionally, since the model is applied on two-dimensional slices of the three-dimensional image stack, the stacked binary masks do not fit seamlessly together. In order to smooth the boundaries of the segmenation in the three-dimension image, a Guided Image Filtering published by He et al. (2013) was applied. The refined mask was then used to establish the radii and orientations of the proteoglycans in the images, again based on the Frangi algorithm.

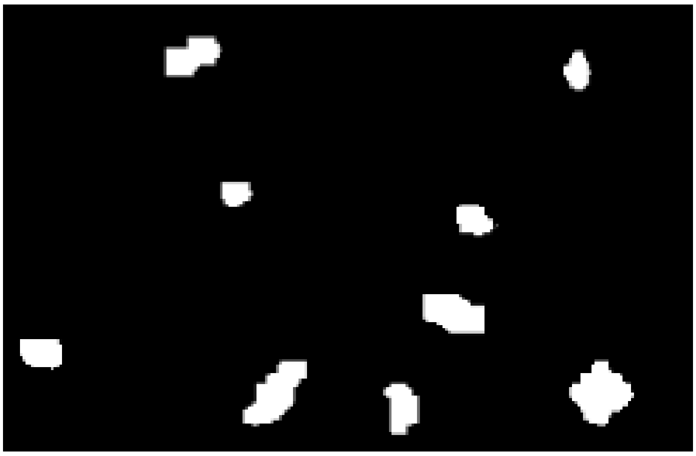


Figure 5.1 Edgy proteoglycan borders of the segmentation.

5.1 Methods

5.1.1 Guided Image Filtering

The Guided Image Filter filters an input image F by means of a guidance image I (see Figure 5.2). To use this filter as a ”mask refinement”, the unprocessed input image and the binary output image are the filter’s guidance I and input F, respectively.

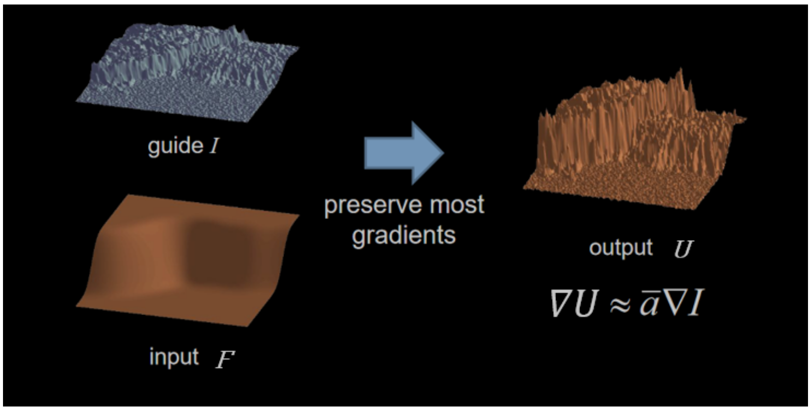


Figure 5.2 Refine input (unrefined mask) F with guidance (unprocessed original image) I to get a refined mask U, image adapted from He et al. (2010).

Definition

First a $(2r + 1) \times (2r + 1)$ square window w_p with radius r centered around pixel p is defined:

$$\omega_p = \{q : ||p - q||_\infty \leq r\} \tag{5.1}$$

A filter output pixel U_q inside this window w_p can be represented as a linear transformation of I inside this window. For the local linear model stated in equation 5.2 $\nabla U = a \nabla I$ is valid. Hence, U inherits edges from I and has an edge only if guidance I has an edge.

$$U_q = a_p I_q + b_p \qquad \forall q \in \omega_p \tag{5.2}$$

Linear coefficients a_p and b_p are assumed to be constant inside w_p and can be determined by solving the minimization problem stated in equation 5.3. The regularization parameter

$\epsilon > 0$ prevents a_p from becoming too large. It can be seen as a parameter to set "the extent of edge-preservation": the lower the chosen value, the more edges in the filter input image are kept.

$$E(a_p, b_p) = \frac{1}{2} \sum_{q \in \omega_p} ((a_p I_q + b_p - F_q)^2 + \epsilon a_p^2) \tag{5.3}$$

After solving the linear regression problem the parameters can be written as:

$$a_p = \frac{\frac{1}{|\omega_p|} \sum_{q \in \omega_p} F_q I_q - \mu_p m_p}{\sigma^2 + \epsilon} \tag{5.4}$$

$$b_p = m_p - a_p \mu_p \tag{5.5}$$

$|\omega_p|$ is the number of pixels in ω_p, μ_p and σ^2 are mean and variance of guidance I in ω_p and m_p is the mean of filter input F in ω_p. As the linear model is applied to the entire image, each pixel q is involved in all windows ω_p which include q, therefore $|\omega_p|$ different solutions U_q are computed. One strategy is to average over all possible values for U_q in order to come up with a final intensity value:

$$U_q = \frac{1}{|\omega_p|} \sum_{p: q \in \omega_p} (a_p I_q + b_p) = \bar{a}_q I_q + \bar{b}_q \tag{5.6}$$

In Equation 5.6, $\bar{a}_q = \frac{1}{|\omega_q|} \sum_{p \in \omega_q} a_p$ and $\bar{b}_q = \frac{1}{|\omega_q|} \sum_{p \in \omega_q} b_p$. Since $\bar{a}_q$ and $\bar{b}_q$ are outputs of an averaging filter operation, $\nabla U \approx \bar{a} \nabla I$ is still valid. This means that severe intensity changes in the guidance image I will be (mostly) present in the filter output U as well (He et al., 2013; Pock, 2019).

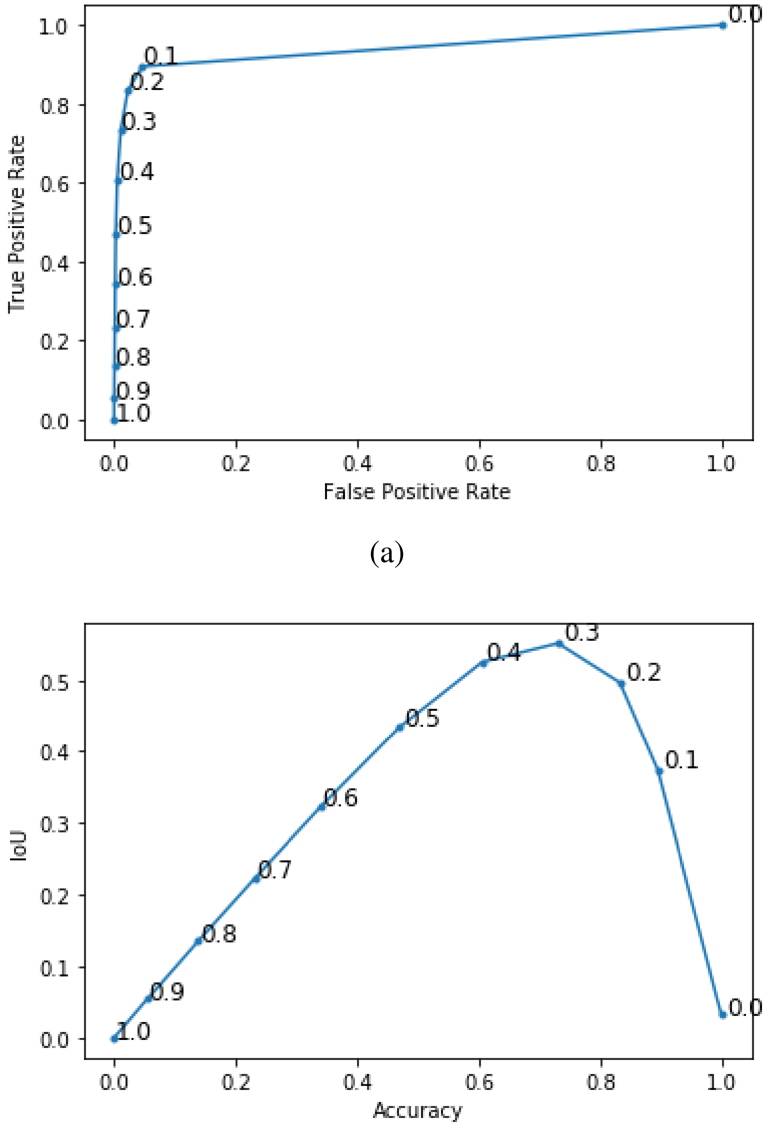
Guided Image Filter for Mask Refinement

Due to the preservation of severe intensity changes in the input image, gradients appearing in the original image at the boundaries between proteoglycan- / collagen fibril-structure will be incorporated in order to get a refined mask. For this thesis, $r = 8$ and $\epsilon = 0.05$ using the images with a resolution $50 \times 1024 \times 1024$ pixels. The output of the filter was then converted into a binary image. A good threshold was determined by computing a ROC (Receiver Operator Characteristic) Curve. The curve gives a relationship between the false positive rate (FPR) on the x-axis and the true positive rate (TPR) on the y-axis:

$$FPR = \frac{FP}{FP + TN} \tag{5.7}$$

$$TPR = \frac{TP}{TP + FN} \tag{5.8}$$

TP describes the amount of true positives, FP is the amount of false positives. TN and FN is the number of true negatives and false negatives, respectively. Additionally, a plot showing the accuracy and IoU for different threshold values was examined. The filter output was normalized between 0 and 1 before comparing different thresholds. It turned out that a threshold set to 0.3 led to the most reasonable results.



(a)

(b)

Figure 5.3 The threshold was set to 0.3 according to the evaluation of ROC (Figure 5.3a) and IoU / Accuracy curve (Figure 5.3b).

5.1.2 Orientation and Diameter Estimation based on Frangi Tube Detection

The Frangi algorithm was then applied on the three-dimensional previously refined binary mask with $\sigma \in \{1, ...14\}$. A non-maxima suppression was computed to extract the tubes' centerlines. It can be said that the multiscale response R_{ms} of a voxel on a tube's centerline is a local maximum. Assuming that the converse is true, local maxima in the multiscale response R_{ms} are located at the centerline and extracting local maxima leads to centerline detection.

The eigenvectors $\mathbf{v}_2$ and $\mathbf{v}_3$ of the Hessian Matrix $\nabla^2 I^{\sigma_{max}}(\mathbf{x})$ at scale σ_{max}, the scale where the maximum response was determined at that voxel **x**, is examined in order to compare multiscale responses in the normal plane. A voxel **x** can be considered to be a local maximum and therefore to be located at the centerline if following criteria are fulfilled (Krissian et al., 2000):

$$R_{ms}(\mathbf{x}) \geq R_{ms}(\mathbf{x} \pm \mathbf{v}_2) \qquad \text{and} \qquad R_{ms}(\mathbf{x}) \geq R_{ms}(\mathbf{x} \pm \mathbf{v}_3). \tag{5.9}$$

Evaluation of Diameters

σ_{max} at the centerline is used to estimate the diameter. The mathematical model states that the intensities in plane follow a Gaussian distribution with a certain standard variance. The highest response value corresponds to the Gaussian distribution whose standard deviation fits best to the actual tube cross section. Therefore σ_{max} can be used to estimate the radius of the tube. In order to find a mathematical relationship between scale and diameter, the diameter of a proteoglycan was measured in the available ground truth images (resolution of 100 × 2048 × 2048 voxel) and the corresponding scale value at the centerline of the Frangi filter response (resolution of 25 × 512 × 512 voxel) evaluated.

Figure 5.4 shows all 60 diameter/scale pairs and the linear function $d(\sigma_{max})$ (equation 5.10) which was fitted accordingly, diameters are given in voxels:

$$d(\sigma_{max}) = 7.88 * \sigma_{max} + 11.03 \tag{5.10}$$

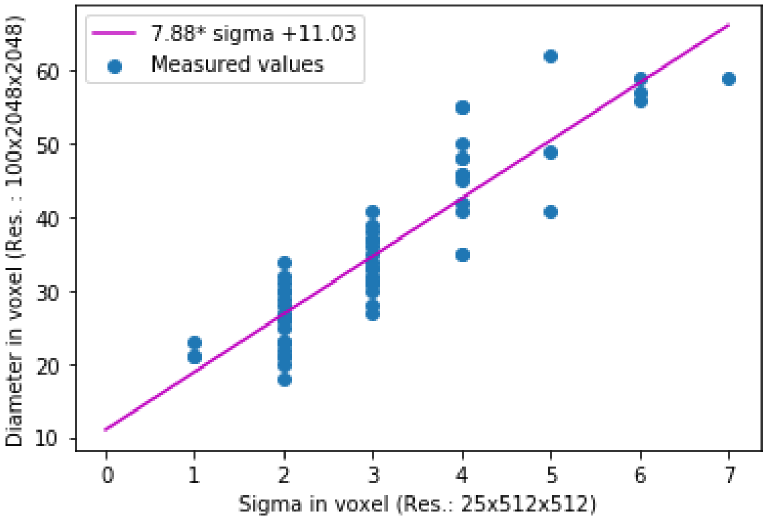


Figure 5.4 Scale - diameter relationship. The diameter is given in voxels.

Evaluation of Orientations

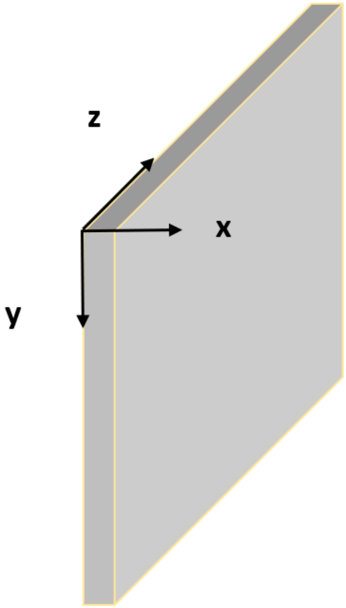


Figure 5.5 Image stack in coordinate system: x goes in “slice-direction”, y in “row-direction” and z in “column-direction”.

As it is assumed that $\mathbf{v}_1$ shows in tube direction, its coordinates could be used as an indicator for the orientation of proteoglycans. In order to interpret the orientations of the eigenvectors $\mathbf{v}_1$, their coordinates were converted into spherical coordinates as used in Holzapfel et al. (2015). With equations 5.11, 5.12 and 5.13, the direction of a proteoglycan can be defined in spherical coordinates as illustrated in 5.6.

$$r = \sqrt{v_{1,x}^2 + v_{1,y}^2 + v_{1,z}^2} \tag{5.11}$$

$$\Phi = \text{arctan}(\frac{v_{1,z}}{v_{1,y}}) \tag{5.12}$$

$$\Theta = \text{arcsin}(\frac{v_{1,x}}{r}) \tag{5.13}$$

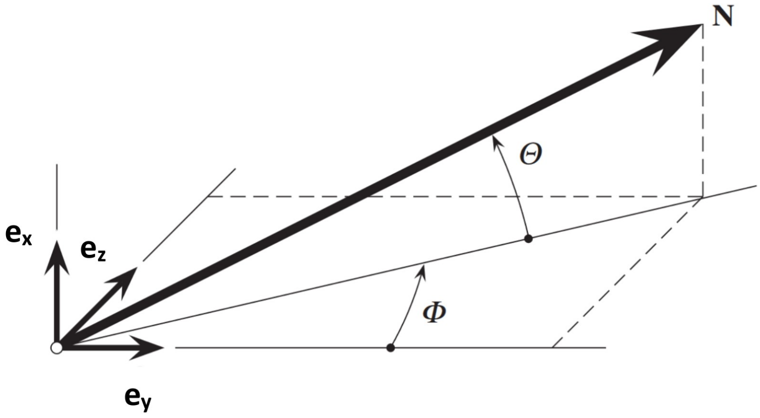


Figure 5.6 A unit vector $\mathbf{N}(\Phi, \Theta)$ can be defined by angles Φ and Θ in a rectangular Cartesian coordinate system. $\mathbf{e_x}$, $\mathbf{e_y}$ and $\mathbf{e_z}$ are unit vectors of the coordinate system, adapted from Holzapfel et al. (2015).

Biomechanical Interpretation

In order to interpret the measurements in a biomechanical context, it is necessary to convert the diameter from voxels to nanometers. In the reconstructed three-dimensional images with resolution 100 × 2048 × 2048 voxel, all voxels have a size of 0.5317 × 0.5317 × 0.5317 nm. This means that the estimated diameter d in voxel can be converted to a diameter $\tilde{d}$ in nm by multiplying it by 0.5317.

$$\tilde{d}(\sigma_{max}) = d(\sigma_{max}) * 0.5317 = 4.19 * \sigma_{max} + 5.86 \tag{5.14}$$

Tested aortas were cut in axial direction to get a planar tissue sample. These tissue samples were then treated biaxially, that means they are stretched in their circumferential and their radial direction. Figure 5.7 shows an illustration of vessel directions.

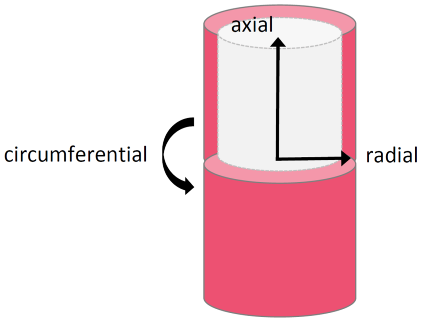


Figure 5.7 Simplified illustration of a vessel with depicted axial, circumferential and radial direction.

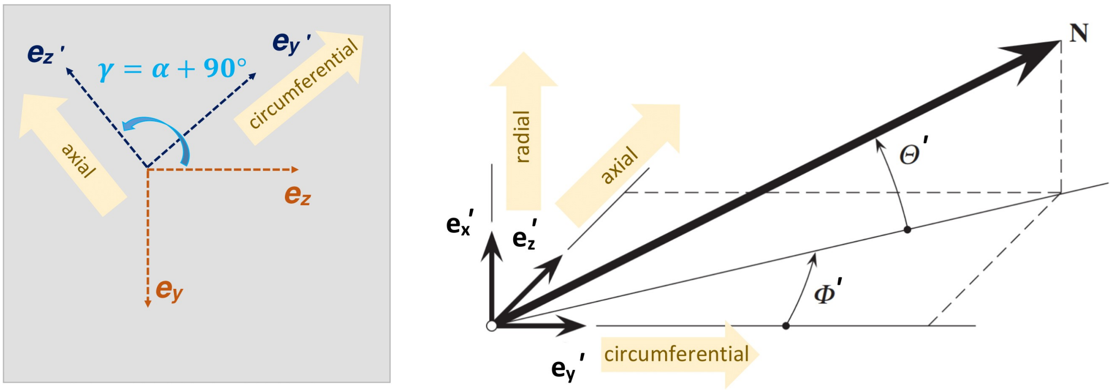
The radial direction is always aligned with the x-axis of the image stack, but circumferential and axial direction of the tissue sample do not have to be oriented along y- and z-axis of the image stack. The angle between z-axis and circumferential direction of the sample is already determined by the Institute of Biomechanics. Based on this angle α, the plane of the coordinate system spanned by y- and z-axis can be rotated around the x-axis (radial axis) by means of a rotation matrix. In order to align the rotated y-axis with the circumferential direction and rotated z-axis with axial direction, the coordinate system is additionally rotated by 90° . In total, the coordinate system is rotated by $\gamma = \alpha + 90°$.

$$\begin{pmatrix} v'_{1,x} \\ v'_{1,y} \\ v'_{1,z} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix} \begin{pmatrix} v_{1,x} \\ v_{1,y} \\ v_{1,z} \end{pmatrix} \tag{5.15}$$

Rotated axis y' is then aligned with the circumferential direction, z'-axis is aligned with the axial direction. The equations for spherical coordinates can be altered accordingly:

$$\Phi' = \arctan(\frac{v'_{1,z}}{v'_{1,y}}) \tag{5.16}$$

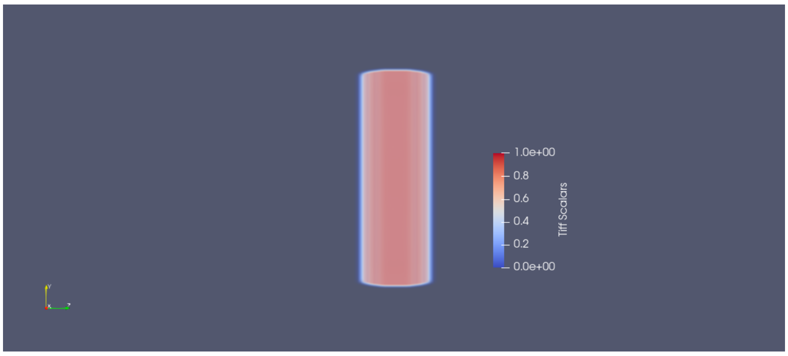
$$\Theta' = \arcsin(\frac{v'_{1,x}}{r}) \tag{5.17}$$



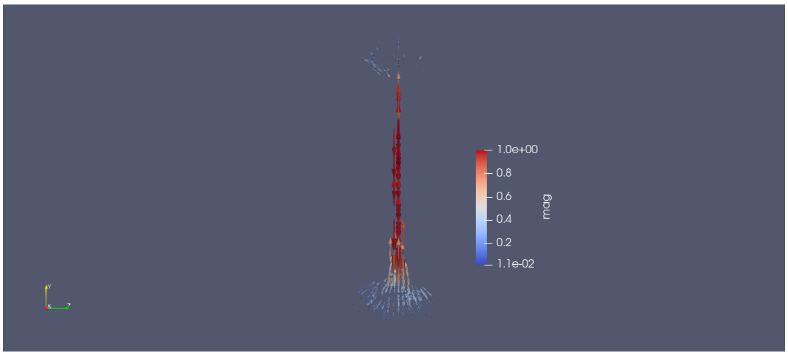
(a) (b)

Figure 5.8 y- and z-axis are rotated by γ in order to align them to axial and circumferential direction of the tissue sample. Θ' and Φ' can then be used to evaluate proteoglycan orientations with respect to axial, circumferential and radial direction. Figure 5.8b is adapted from Holzapfel et al. (2015).

Artefacts at Tube ”Endpoints” - Additional Thresholding



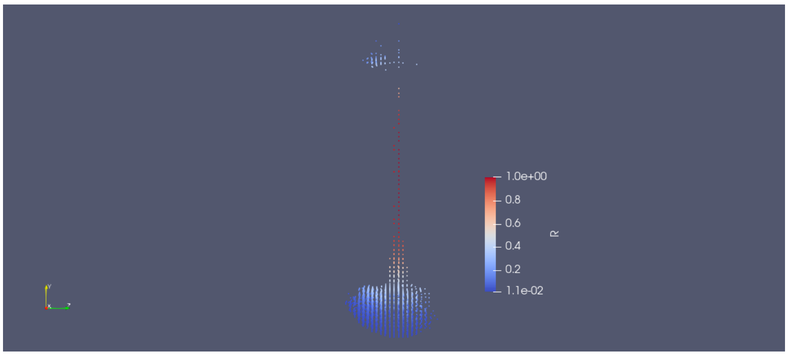
(a) “Perfect” tube



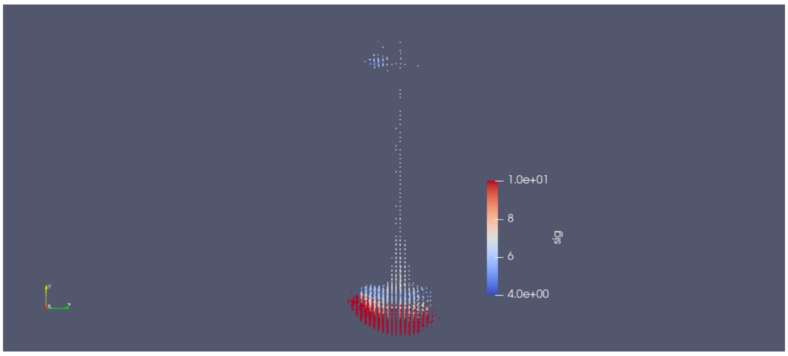
(b) Eigenvectors v_1

Figure 5.9 One can see that the determination of eigenvectors is not working that well at “end regions” of a perfect tube as it is depicted in Figure 5.9a. The assumption that intensities do not change in tube direction is not valid at the top and the bottom of the tube. The eigenvectors v_1 may not be oriented in tube direction at these “end regions” (Figure 5.9b).

In order to see if the non-maxima suppression and the estimation of orientation and radii makes sense, the evaluation was also performed on computer generated tubes, two-dimensional circles were stacked upon each other to create a three-dimensional tube. Figures 5.9 and 5.10 visualizes eigenvectors, response and scale values of a tube which is ”shorter than the image stack”. Abrupt intensity changes at the top and at the bottom contradict the mathematical basis which states that intensities do not change in tube direction and this then leads to miss-determination of tube orientations. One can see in Figure 5.10a that response values at voxels where the mathematical assumption does not fit properly are rather low and a thresholding is sufficient to eliminate those artefacts. Therefore, additionally to the criteria for the non-maxima suppression (Equation 5.9), it was checked if response values were higher than a chosen threshold $t = 0.1$.



(a) Response values after non-maxima suppression.

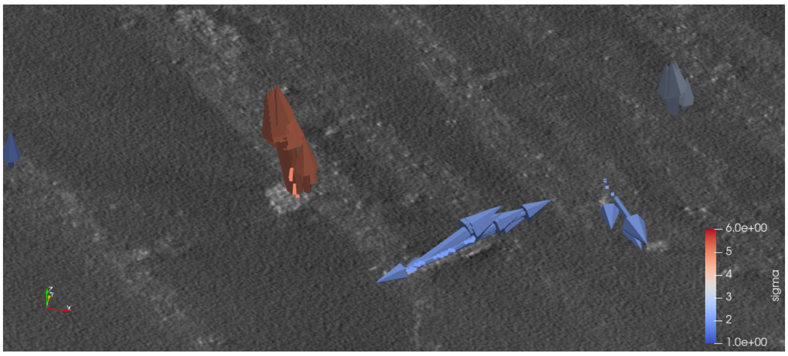


(b) σ after non-maxima suppression.

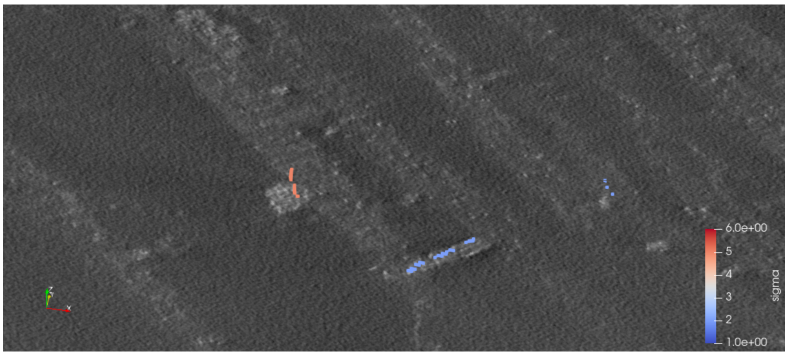
Figure 5.10 Visualization of multiscale response (Figure 5.10a) and σ (Figure 5.10b) after a non-maxima suppression. Again, one can see the artefacts at the top and at the bottom of the tube due to the deviation of the mathematical tubularity assumption.

5.2 Results and Discussion

(a) Image section of one slice of the image stack showing two proteoglycans. PG 1 is “standing” in the image stack, PG 2 is “lying”.



(b) Visualized main directions, the direction of PG 1 is oriented “upwards”, PG 2 is oriented more along the image slice plane.



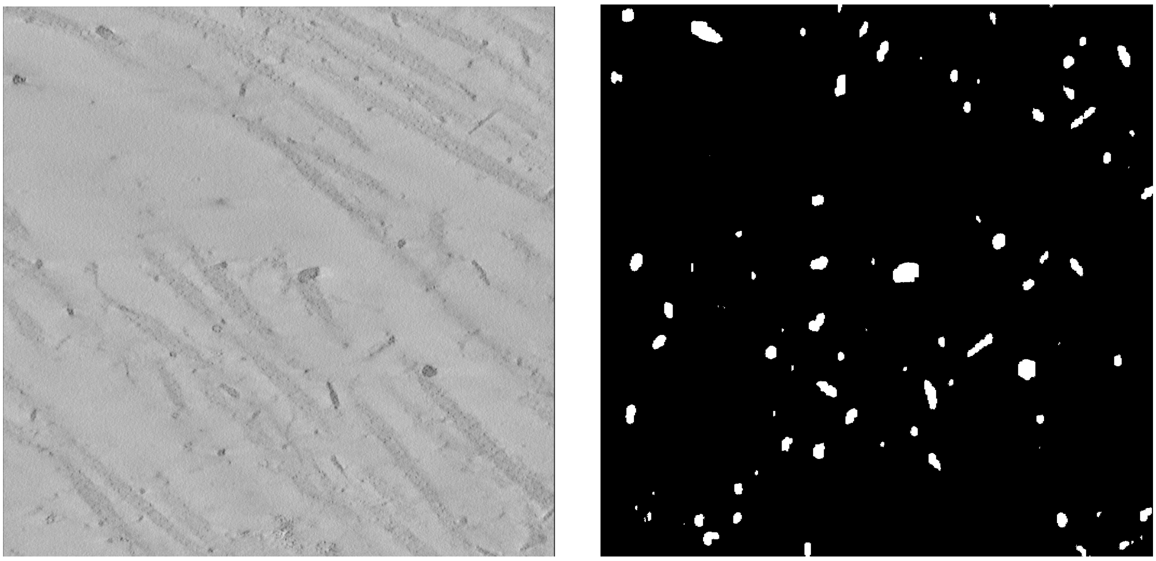
(c) Visualized scales σ_{max}. The scale of PG 1 is significantly larger than the scale of PG 2.

Figure 5.11 Two exemplary proteoglycans (Figure 5.11a) with depicted orientations (Figure 5.11b) and scales (Figure 5.11c) at their centerlines.

Figure 5.11 shows an example of determined main orientations and scales. For Proteoglycan 1, which can be described as to be ”standing” in the image stack, the estimated direction is oriented perpendicularly to the slice plane. The main direction of the ”lying” Proteoglycan 2 is estimated to be oriented more or less along the plane. Both estimated directions are very reasonable. For Proteoglycan 1, scale $\sigma_{max} = 5$. Using equations 5.10 and 5.14, a diameter $d = 50.43$ in voxels and $\tilde{d} = 26$ in nm was determined. For Proteoglycan 2 σ_{max} is equal to 2, resulting in a diameter $d = 26.79$ in voxels and $\tilde{d} = 14.24$ in nm. The proteoglycans' diameter was measured manually, both measurements matched the computations.

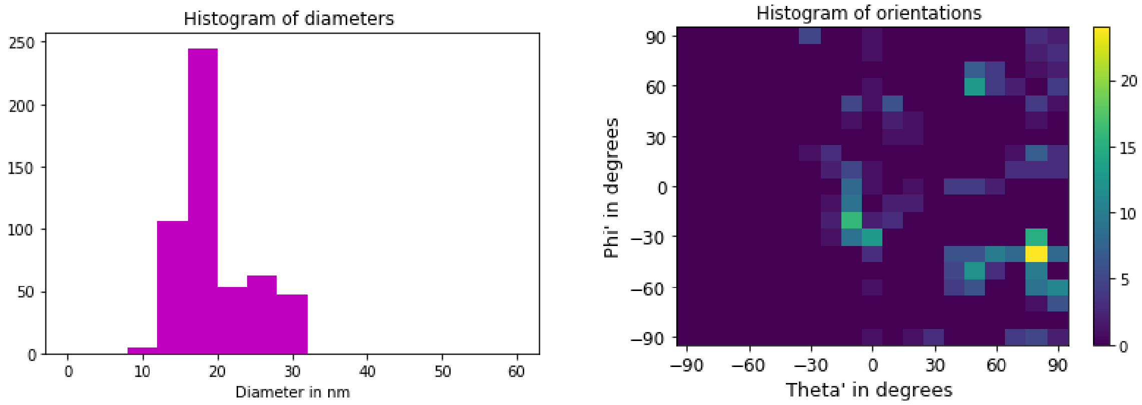
Figures 5.12 and 5.13 show the same slice of the original image stack and the segmentation of proteoglycans, the histogram of diameters and the histogram for the spherical angles for both datasets. One can see that the proteoglyans of the first dataset are generally smaller than the proteoglycans in the second image stack. This is also reflected in the histograms. The histogram of the first dataset suggests that most proteoglycans have a diameter around 18 nm. Proteoglycans with greater diameters are rarely present and proteoglycans with a diameter larger than 30 nm were not detected at all. In contrast, the histogram of the second dataset shows that there is a significant amount of proteoglycans with a diameter between 18 and 30 nm. The largest detected proteoglycans have a diameter of 34 nm.

Regarding the orientations of proteoglycans in the images, it is observable that for the majority of proteoglycans for the first dataset (Figure 5.12d), Θ' is close to 90 ° and Φ' is approx. -45 °, suggesting that they are oriented mainly out of plane along radial direction with a tendency to be directed “in between” circumferential and axial direction. In the second dataset, the histogram (Figure 5.13d) shows that although many proteoglycans are oriented out of plane as well, for a large extent of proteoglycans Θ' is approx. -45 °and Φ' around 0 °, which means that they are only slightly oriented out of plane and significantly directed in circumferential direction.



(a) Original image

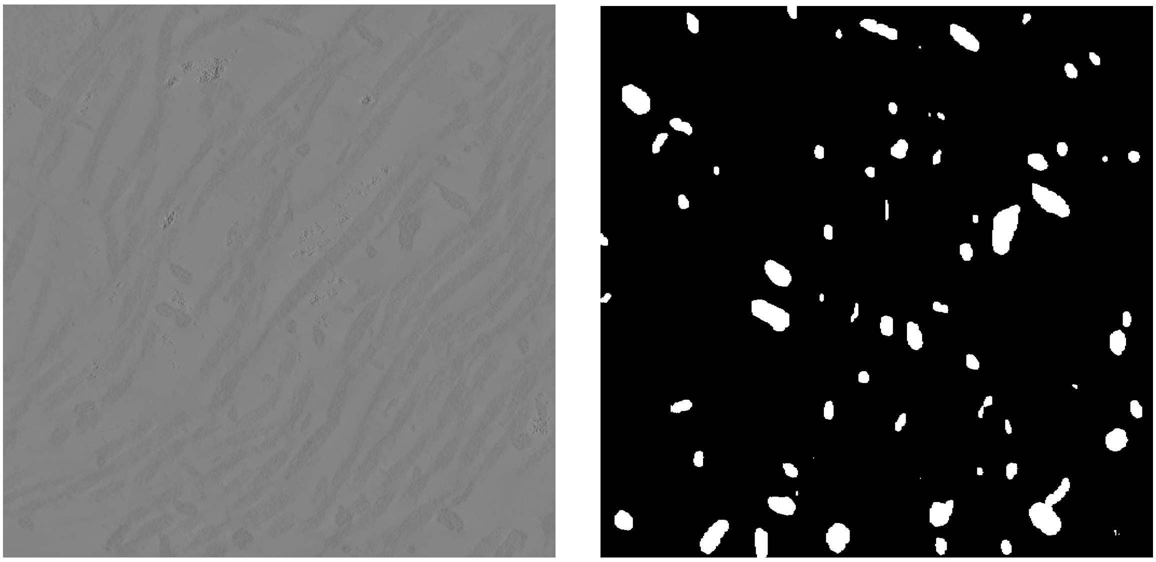
(b) Proteoglycan segmentation



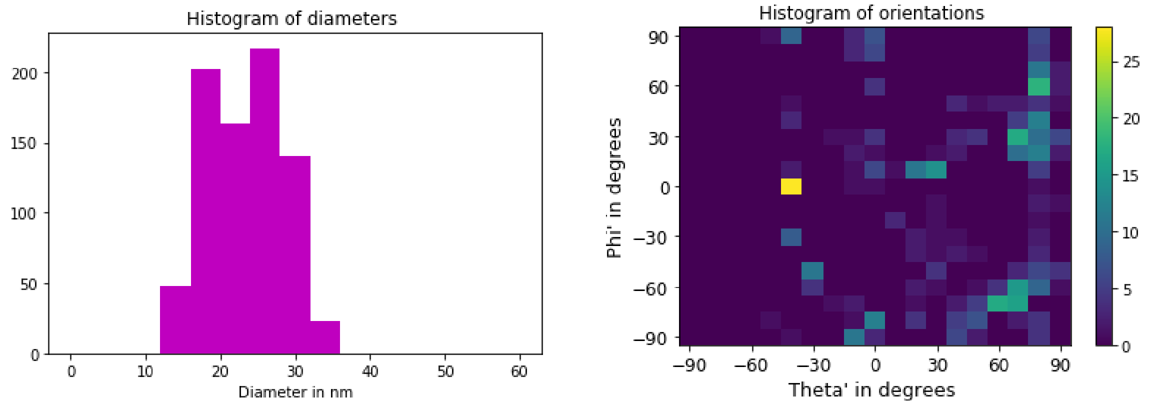
(c) Histogram of diameters

(d) Histogram of Θ' and Φ'

Figure 5.12 First exemplary dataset with one slice of the original image stack (Figure 5.12a) and its segmented proteoglyans (Figure 5.12b). The diameters of the segmented proteoglycans in the whole dataset were plotted in a one-dimensional histogram (Figure 5.12c) and angles Θ' and Φ' of the proteoglycan orientations were plotted in a two-dimensional histogram (Figure 5.12d).



(a) Original image (b) Proteoglycan segmentation



(c) Histogram of diameters (d) Histogram of Θ' and Φ'

Figure 5.13 Second exemplary dataset with one slice of the original image stack (Figure 5.13a) and its segmented proteoglyans (Figure 5.13b). The diameters of the segmented proteoglycans in the whole dataset are given in a one-dimensional histogram (Figure 5.13c), angles Θ' and Φ' are given in a two-dimensional histogram (Figure 5.13d).

6 Conclusion

The prerequisites for this thesis were quite demanding. With the naked eye, it is relatively hard to recognize and identify proteoglycans and collagen fibrils. Various structures are not assignable with certainty. Hence, segmenting images manually is highly laborious and cumbersome, reliable three-dimensional annotations are especially difficult to generate. As a result, only very little labelled training data for evaluating methods and training machine learning models was available.

Both “classical” approaches, thresholding and Frangi algorithm, have the big advantage of being transparent and plausible, one is able to interpret and comprehend their behaviour at any step and datapoint. Thresholding is computationally inexpensive and although a series of calculations are computed at every voxel for the Frangi algorithm, it is possible to run the program on a simple CPU in reasonable time if the resolution of the images is decreased. However, the underlying assumption for the tube detection algorithm is a severe simplification. In reality, the recorded data does not show infinitely long tubular structures and the intensity profile in the normal plane might not be described well by a Gaussian distribution. If the structures in the 3D dataset deviate too much from the tubular assumption, the algorithm can behave poorly. Furthermore, for both thresholding and Frangi algorithm it is often necessary to preprocess the images in some way, the preprocessing process itself can be quite demanding. Choosing appropriate parameters for both approaches is challenging too.

Both deep learning models outperformed previous approaches significantly. Although the models rely only on two-dimensional images, GPUs were absolutely essential for their training. However, the trained models are able to compute segmentation masks in real-time, no preprocessing is necessary.

The evaluation of diameters and orientations was done as an additional task because the segmentations achieved by the ESPNet model were considered to be sufficiently reliable.

The strategy used, based on the considerations of the Frangi algorithm, leads to reasonable findings and is a good starting point for further investigation of segmented structures. But as the mathematical tubularity assumption is violated at “end regions” of tubes and considering that proteoglycans and their “central regions” are rather short, other approaches could lead to more solid results, e.g. a medialness transform.

Although the segmentation of both structures turned out to be surprisingly good, the model could be improved considerably by training it on more data, especially on three-dimensional data. However, labelling three-dimensional data is labour-intensive and cumbersome. An approach to tackle this issue could be based on the method proposed by Koziński et al. (2020).

They developed a method to train deep neural networks to segment three-dimensional images of linear structures using only annotations of two-dimensional maximum intensity projections of the training data. They introduce a loss function which penalizes deviations between the output of the network and the annotated maximum intensity projections along all three axes. The annotation expense is decreased severely without any worsening of the model performance.

Bibliography

Antonin Chambolle, Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25: 161 - 319, 2016.

Emmert-Streib Frank, Yang Zhen, Feng Han, Tripathi Shailesh, Dehmer Matthias. An Introductory Review of Deep Learning for Prediction Models With Big Data *Frontiers in Artificial Intelligence*, 3: 2624 - 8212 2020.

Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken, Max A. Viergever. Multiscale vessel enhancement filtering. *Proc Medical Image Computing and Computer-Assisted Intervention*, 130 - 137, 1998.

Pascal Frey, M. De Buhan . Chapter 6: The finite difference method. *The numerical simulation of complex PDE problems (lecture notes 2008)*, 79 - 92, 2008.

Kaiming He, Jian Sun and Xiaoou Tang. Guided Image Filtering *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.35 (No. 6): 1397 - 1409, 2013.

Kaiming He, Jian Sun and Xiaoou Tang. Guided Image Filtering (ECCV Presentation Slides) *Presentation Slides for the European Conference on Computer Vision 2010* , 2010.

Gerhard Holzapfel. Collagen in Arterial Walls: Biomechanical Aspects. *Collagen Structures and Mechanics*, 2008.

Gerhard A. Holzapfel1, Justyna A. Niestrawska, Ray W. Ogden,Andreas J. Reinisch and Andreas J. Schriefl. Modelling non-symmetric collagen fibre dispersion in arterial walls. *Journal of the Royal Society Interface*, 12: 20150188, 2015.

David J. S. Hulmes. Collagen Diversity, Synthesis and Assembly. *Collagen, Structure and Mechanics*, Springer, Chapter 2, 15 - 47, 2008.

Karl E.Kadler, David F. Holmes, John A. Trotter and John A. Chapman. Collagen Fibril Formation. *Biochem. J.*, 316: 1 - 11, 1996.

Patrick Knöbelreiter, Christian Sormann, Alexander Shekhovtsov, Friedrich Fraundorfer, Thomas Pock. Belief Propagation Reloaded: Learning BP-Layers for Labeling Problems, 2020.

Karl Krissian, Grégoire Malandain, Nicholas Ayache, Régis Vaillant, Yves Trousset. Model-Based Detection of Tubular Structures in 3D Images. *Computer Vision and Image Understanding*, Elsevier, 80 (2): 130 - 171, 2000.

Mateusz Koziński, Agata Mosinska, Mathieu Salzmann, Pascal Fua. Tracing in 2D to reduce the annotation effort for 3D deep delineation of linear structures. *Medical Image Analysis*, 60: 2020.

Fei-Fei Li, Justin Johnson, Serena Yeung. Commonly used activation functions. *CS231n: Convolutional neural networksfor visual recognition 2017, Neural Networks Part 1: Setting up the Architecture* , 2017.

Kevin Linkan, Vu Ngoc Khiêm, Mikhail Itskov. Multi-scale Modeling of soft fibrous tissues based on proteoglycan mechanics. *Journal of Biomechanics*, 49 (12): 2349 - 2357, 2016.

Jeffrey M. Mattson, Yunjie Wang, Yanhang Zhang. Contributions of Glycosaminoglycans to Collagen Fiber Recruitment in Constitutive Modeling of Arterial Mechanics. *Journal Of Biomechancs*, 82: 211 - 219, 2019.

Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, Hannaneh Hajishirzi. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. 2018.

Tanmayi Oruganti, Jan G. Laufer, and Bradley E. Treeby . Vessel Filtering of Photoacoustic Images. *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*, 2013.

Thomas Pock, Reinhard Beichel, Horst Bischof . A Novel Robust Tube Detection Filter for 3D Centerline Extraction. *Proc 14th Scandinavian Conference on Image Analysis (SCIA 2005)*, 481 - 490, 2005.

Thomas Pock. Advanced Filtering *Lecture notes on Image Processing and Pattern Recognition*, Chapter 2: 18 - 32, 2019.

Thomas Pock. Total Variation. *Image Processing and Pattern Recognition (Lecture notes)*, 1 - 61, 2020.

Martin Urschler. Vascular structures. *Lecture Summary of Course Medical Image Analysis 710.076 (Lecture notes)*, 49 - 52, 2019.

Dmitry Ulyanov, Andrea Vedaldi and Victor Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. 2016.

Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, LNCS, Vol.9351: 234- 241, 2015.

Sebastian Ruder. An overview of gradient descent optimization algorithms. 2016.

Patrice Y. Simard, David Steinkraus, John C. Platt. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis *Seventh International Conference on Document Analysis and Recognition.* 958 – 963, 2003.

Jianxin Wu. Introduction to Convolutional Neural Networks. 2017

Yuxin Wu and Kaiming He. Group Normalization. 2018.

Statutory Declaration

I declare that I have authored this Thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material, which has been quoted by the relevant reference.

date

signature