Nikolaus Jaufer

Automated segmentation and morphometry of muscle fibers from haematoxylin-eosin-stained histological sections

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieur

Master's degree programme
Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Thomas Pock
Institute for Computer Graphics and Vision

Dipl.-Ing. Dr.techn. Martin Urschler
Ludwig Boltzmann Institute for Clinical Forensic Imaging

Graz, Austria, October 2017

Affidavit

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 indicated all material which has been quoted either literally or by content from the sources used.

The text document uploaded to TUGRAZonline is identical to the present master's thesis dissertation.

Date	Signature

Acknowledgments

Firstly I would like to thank my mentor Dr. Thomas Pock, for supporting my individual master program and the facilitation of writing my master thesis while working full time.

Secondly, I would like to thank my supervisor Martin Urschler and Philipp Kainz, whose guidance, support and feedback on the topic made this thesis possible.

Next, I would like to thank the members of the Institute for Computer Graphics and Vision for their altruistic commitment, constructive discussions and their significant hints, namely Darko Stern and Christian Payer.

Parts of this work were done in collaboration with the Division of Phoniatrics of the Medical University of Graz. I would like to thank Gugatschka Markus, Karbiener Michael and Gerstenberger Claus for providing image data, in-depth medical support and their contagious display of commitment and relevance of this topic.

Finally, I would like to express my gratitude towards Katrin for her love and understanding which she has shown during this period.

Contents

List of Figures

List of Tables

1 Introduction

Contents

1.1 Motivation

The proper functionality of all muscles in our body is of critical importance for our health and wellbeing. Unfortunately, as we grow older, we lose muscular mass and consequently strength, and function [82]. Muscle mass decreases approximately 3–8% per decade after the age of 30 and this rate of decline is even higher after the age of 60 [37, 52]. In the EU-28's, 18.9% of the population are 65 years old or older, a number that is expected to rise to 28.1% in 2050[1].

Vocal cord atrophy is most commonly caused by aging (presbyphonia) but also by nerve injury (viruses, trauma, intubation, tumors, etc.). The quantity of elderly who are affected by vocal cord atrophy is between 12 and 35% [34, 40]. Other studies show a range between 4.8 and 29.1%, considering the general population aged 60 years or older [20].

Symptoms include vocal fatigue, difficulty with projection, reduced vocal range, breathy voice, inability to hold a note for a long time or pitch breaks during a long note to name a few. These lead to a significant impact on the functional use of voice.

Furthermore the perception of listeners of affected speakers may be negatively influenced. Additionally the reduced quality of the voice may lead to social withdrawal [81].

[1] Eurostat, Population structure and aging, http://ec.europa.eu/eurostat/, 2017.06.29

The loss of muscular mass of the vocal cords in the larynx (vocal cord atrophy) is an important research question in phoniatrics, which is being studied by our medical collaboration partner[2].

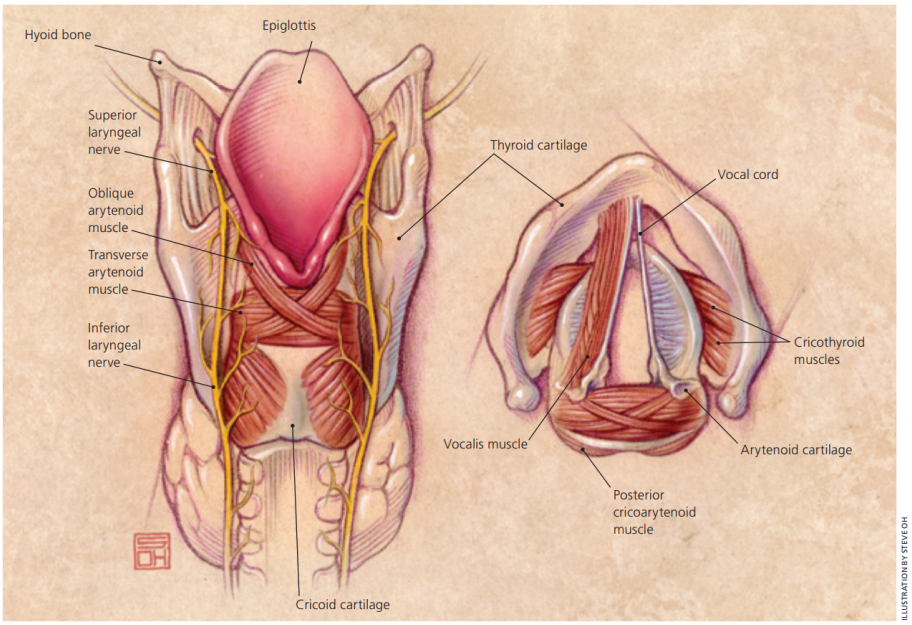


Figure 1.1: Laryngeal anatomy. (Left) Posterior view of larynx, and (right) cross-section of larynx from above (taken from Feierabend and Shahram [25, page 364]).

The larynx is a structure interconnected with many parts that serves several functions including protection, respiration, deglutition and vocalization. It extends from the trachea to the base of the tongue. It is composed of an underlying cartilaginous, bony, and membranous structure with an overlying mucosal lining (Figure 1.1). The foundation of the larynx is formed by the cricoid, thyroid, arytenoid, epiglottic, corniculate, and cuneiform cartilages, interconnected by ligaments and membranes. It is moved by extrinsic and intrinsic muscles. The vocal cords (vocal folds) are the primary source of human voice production. They are membranous structures attached to the arytenoid and thyroid cartilages, and stretched across the larynx. The larynx is innervated by the superior and recurrent laryngeal nerves, which are branches of the vagus nerve. Sounds are produced by the lung, when it produces an adequate airflow and air pressure to cause the vocal cord epithelium to vibrate, thus fluctuations in air pressure produce sound waves (vibrations) that form audible pulses. The edges of the vocal cord must be brought together close enough to vibrate from the flow of air through the larynx to generate sound. The arytenoid cartilages and attached muscles are responsible for movement and tension of the vocal cords. Resonance of the sound waves is influenced by the position and shape of the lips,

[2]Division of Phoniatrics, ENT University Hospital, Medical University of Graz

jaw, tongue, soft palate, and other speech organs [25, 73].

Vocal Cord Atrophy Vocal cord (fold) atrophy refers to a gradual change in the vocal cord as people age or due to a nerve injury. The vocal fold muscle(s) can become thinner and/or less taut overtime.

Current treatments of vocal cord atrophy are split into two major groups, surgical methods and voice therapy. Both aim at increasing vocal loudness, reducing vocal effort and therefore improving voice related quality of life.

Voice therapy requires periodic, tailored (choice of technique, order and length) sessions that meet the needs of each individual patient [88]. Outcome measures are limited to voice perception and stroboscopic ultrasound imaging and have a shortfall in measuring the success because of missing randomized control groups, or lacking consistent criteria for determining efficacy [10]. Additionally, since the structures of interest (especially the thyroarytenoid muscles [TAM]) are inaccessible in humans, no conclusion about changes on the muscular level can be derived. Surgical approaches aim to passively restore the glottal competence by altering the laryngeal anatomy [4, 28, 38]. Well established methods require thyroplasty [59], where the target is to antagonize vocal fold atrophy and thereby improve phonation and voice quality [38]. From these established treatments, it is difficult to identify the best approach for each individual patient, since there are no controlled settings. Doctors intend to start with the method that has the lowest risk for the patient, and continue with more invasive methods [9] until either the patient is satisfied or there are no further options left.

In contrast to these approaches, our medical partner proposes a novel treatment. By applying unilateral Functional Electrical Stimulation (FES) they target to reverse presbyphonia by inducing muscular hypertrophy. Current research on *FES* applications look promising, since they improve the effects as well the scope (i.e. invasive vs. non-invasive) of *FES* by altering frequency, amplitude, duration and wave form of the electrical signal [5, 47].

It has proven to be feasible to gain surpassing training results by applying (additional) electrical stimulated muscle contraction [7, 17, 50, 83]. To determine suitability of *FES* in regards to presbyphonia, our medical partner performs experiments in sheep. They apply *FES* to the recurrent laryngeal nerve (RLN) and therefore target the muscular glottal gap, which is the most prominent feature of the presbylarynx. An implantable pulse generator[3] connected to a cuff electrode[4] powered with a single lithium primary cell was used to apply stimulation patterns (Figure 1.3).

The implantation process (Figure 1.2) was done under general anesthesia. After a healing period of 5-7 days the implants were activated.

[3]Developed at the Centre for Medical Physics and Biomedical Engineering, Medical University of Vienna

[4]Developed at Ardiem Medical, PA, USA

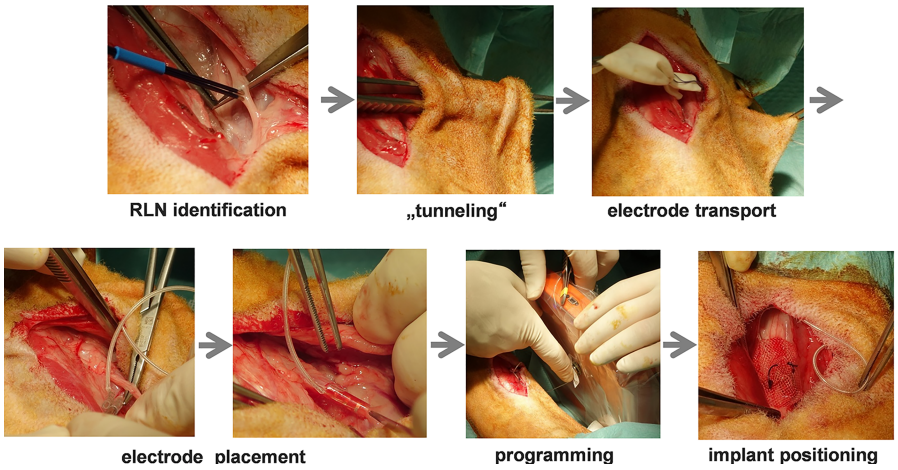


Figure 1.2: For implantation surgery a lateral skin incision was made in the skin of the neck and the vascular nerve sheet containing vagal nerve, carotid artery and internal jugular vein were lateralized. The recurrent laryngeal nerve (RLN) was identified visually and by direct electrical stimulation with an insulated needle. A cuff electrode was wrapped carefully around the RLN. (Taken from [42].)

The initial individual thresholds for *FES* were set to the lowest current amplitude (0.1-0.6 mA) at which changes in the laryngeal adductor pressure could be observed (adductor muscle twitch threshold), targeting a modest overload of the muscles [64]. The stimulation was applied at the right Thyroarytenoid Muscle (TAM) once a day for 29 days [42]. Stimulation patterns were implemented using a combination of predefined stimulation and pause blocks (see Figure 1.3).

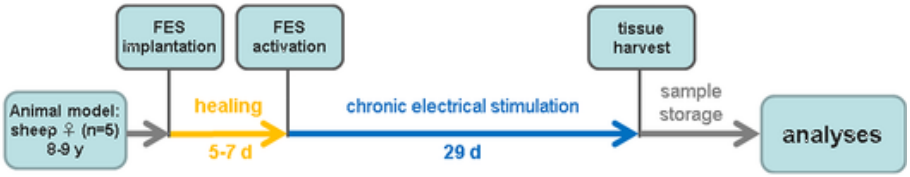
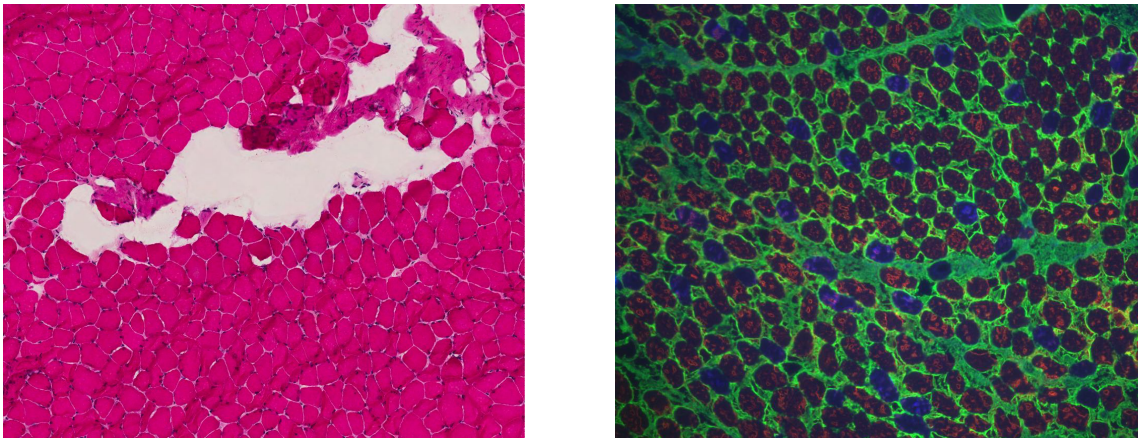


Figure 1.3: Overview of the experimental stimulation of vocal cord in sheep [42]. The daily stimulation pattern (after Functional Electrical Stimulation (FES) activation) consisted of 5 cycles (separated by 1 min) â 8 bursts (3s ON, 0.5s OFF). Frequency was 100Hz, amplitude 0.3 - 2mA (individually adjusted according to threshold).

After the 29-day stimulation period (Figure 1.3), the larynges were excised and the cricoarytenoid muscles were exposed to measure the outcome directly on the muscular level (fiber diameter). For microscopic anatomy analysis of the tissues (histology), a sample from the central region of approximately 4 x 8mm in size was excised and snap-frozen. The current semi-automatic evaluation of the sections requires specific triple immunoflu-

orescence labeling [77], which is done in specialized laboratories[5]. The procedure is time consuming (sending and preparing specimen for further processing) as well as cost intensive.



(a) Haematoxylin-eosin staining

(b) Triple immunofluorescence labeling (combination of all fluorescence images, green: Collagen, blue: MHC type I, red: MHC type II)

Figure 1.4: Different histological labeling: Figure 1.4a shows simple and economically priced standard labeling whereas in Figure 1.4b labeling with specific antibodies to extract special features (i.e. collagen surrounds fibers) was applied.

The protocol starts with verification of correct orientation of specimen via standard haematoxylin and eosin staining [48]. The next step is the preparation with triple immunofluorescence labeling to prepare for extraction of morphometric information (minimum Feret diameter). Therefore our medical partner performs the following steps [42]

1. Drying of frozen cryosections at room temperature for 30 minutes
2. Fixate section with methanol-acetone (1:1) at -20° C for 15 minutes
3. Rinse slides in phosphate-buffered saline (PBS) three times for 5 minutes
4. Incubation with rabbit anti-collagen 6 antibody (abcam, Cambridge, UK) at a dilution of 1:400 for 1 hour
5. Repeat rinsing
6. Alexafluor 488 goat anti rabbit diluted 1:1000 was added for 1 hour

[5] i.e. Department of Veterinary Clinical Sciences, Royal Veterinary College, London

7. Repeat rinsing
8. Incubation with two mouse monoclonal antibodies for 1 hour
    - anti-type 1 myosin heavy chain MHY7 (Merk Millipore Corp., UK) directly labelled with Zenon 350 (Fisher scientific, UK) diluted 1:50
    - anti-type 2 myosin heavy chain MY32 (Abcam. Cambridge, UK) directly labelled with Zenon 594 diluted 1:1000
9. Repeat rinsing
10. Fixate sections using 4% paraformaldehyde in PBS 15 minutes
11. Repeat rinsing

After mounting the specimen, sections are examined using a fluorescence microscope (with specific filters in respect to the emitting wavelengths). An experienced observer selects representative areas depending on their attributes regarding presence of artifacts, sectioning of the muscle and uniform tissue appearance. To extract morphometrics information, the manually selected regions are converted to gray scale and the minimum fiber diameter is measured for each fiber. After these prerequisite steps, this can straightforwardly be achieved, since the collagen antibody staining augments fibers from each other (colored green in Figure 1.4b) and therefore simplifies individual fiber identification in contrast to Figure 1.4a.

Our goal in this work is to reduce this elaborated time consuming routine by providing an automated segmentation and calculation of morphometrics information by directly evaluating Haematoxylin Eosin (HE) stained slides [48]. We aim to introduce a precise (separation of properly stained muscle fiber areas from connective tissue as well as from artifacts which occur during sample preparation) and user-friendly processing pipeline. Furthermore, we aim to reduce the delay, in contrast to triple immunofluorescence labeling in specialized laboratories (preparing specimen, sending and evaluation), from days (or even weeks) to minutes. Therefore we use medical image analysis and evaluate different image segmentation methods as well as a way to incorporate our medical partner to provide labeling expertise.

1.2 Medical Image Analysis

A medical image analysis pipeline for automatic evaluation of morphometrics parameters from histological sections constitutes an extremely helpful and time saving option that can be of broad relevance for various disciplines within medicine.

Medical image analysis [21] plays an important role in many clinical as well as preclinical applications involving data from various imaging modalities. It aims to acquire

useful information about physiological processes or organs to enhance diagnostics by improving manual or computer-assisted interpretation of medical images. Medical image analysis is a process by which meaningful information or measurements can be extracted from digital images. To create the digitized data, there is a variety of different imaging sources [21, 57]. X-ray radiography is the oldest and most commonly used imaging technique where electromagnetic radiation is absorbed, depending on the density and composition of the object, before being captured by a detector. Computed Tomography (CT) combines multiple X-ray projections to produce detailed cross-sectional images of areas inside a body. Magnetic Resonance Imaging (MRI) uses radio waves and a magnetic field to create images of organs or tissues. Positron Emission Tomography (PET) is a nuclear imaging technique which extracts information about how organs and tissues function. Ultrasound uses high frequency broadband sound waves and measures their reflection to produce images [57].

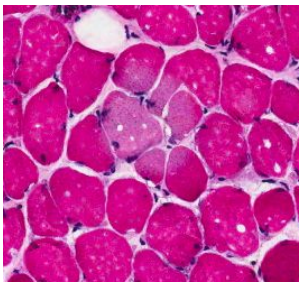
In contrast to the mentioned non invasive medical imaging modalities, histology is the study of sectioned (cut into a thin cross section with a microtome) specimen. To enhance differentiation of microscopic structures, use of histological stains is common. Our source for investigation are digitized *HE*-stained histological slices that we target to segment in order to distinctively identify, segment and measure fibers. Image segmentation is known as the process of partitioning a digital image into multiple regions or objects [30]. The intention of this process, is to change the representation of an image into one, which is easier to analyze. Since image segmentation is influenced by the problem formulation, the application and the basic image data, it is considered a very difficult task in image analysis. Conventional image processing methods for segmentation can broadly be classified into pixel-, edge- and region-based categories [21, 69]. Pixel-based methods apply heuristics or estimation methods derived from the histogram statistics of the image to form closed regions belonging to the objects in the image. Edge-based methods require edge information to resolve boundaries of objects. The boundaries are analyzed and modified as needed to segment the image. Region-based methods analyze pixels directly based on a predefined similarity criterion for a region expansion process until the image is sectioned. More advanced segmentation methods are rapidly developing as novel and more powerful approaches, which are able to cope with raising demand (number of image datasets, size of images, dimensionality). The integration of machine learning into image segmentation provides state of the art solutions to cope with these issues [56]. Supervised methods like classifiers require training data. A commonly used parametric classifier is the maximum-likelihood (Bayes) classifier. Unsupervised methods, like clustering, perform the same function without the need of labeled training data but need to iteratively alternate between segmenting the image and characterizing the properties of each class. Deformable models summarize techniques that use closed parametric curves or surfaces which align, under the influence of internal external forces, to a desired shape. Recently, special attention in the medical imaging domain is paid to deep learning methods, especially Convolutional Neural Networks (CNN) [45]. They present state of the art in semantic segmentation. A

CNN consist of an input-, multiple hidden- and an output layer. The hidden layers typically are composed of convolutional-, pooling-, fully-connected- and normalization layers. *CNN* have the ability to learn adaptively (adjusting the weight of the respective layer), to solve complex problems. Current weight calculation strategies for *CNN* rely on parallelizing the computational task which can be redirected to Graphics Processing Unit (GPU) (i.e. Caffe [39], Torch[6] [15], CNTK[7] [87]). Applying the calculated weights on images to generate segmentation predictions requires a fraction of the training time (i.e. seconds, depending on network configuration, *GPU* or Central Processing Unit (CPU), and image size). We want to use the auspicious capabilities of *CNN*, since a qualitative review of the provided image data in Figure 1.5 indicates that distinct fiber segmentation is no trivial task. Considering that the fundamental requirements to arrive at a digitized slide requires correct biopsy procedure, proper fixation and processing techniques, adequate sectioning and staining, it comes as no surprise that a number of artifacts [12, 53] (Figure 1.5) exist. At first it is necessary to fixate the tissue to avoid autolysis and putrefaction. The next step is to transfer the specimens to cassettes, which requires trimming so that the probe does not touch the edges. To gain a solidified block, tissue processing, which includes dehydration (with alcohol), clearing (removing of alcohol) and embedding (paraffin wax) is required. This block can be sectioned, after it was chilled, with the help of a microtome. Once cut, the tissues are labeled, dried up to melt excess paraffin wax, and placed on slides. Since most cells are transparent, histochemical (*HE*) staining is applied. These slides are scanned (digitized) and delivered to us in different, delayed packages (batches). Batch I consists of 31 images â 1036 x 860 pixels and 24 bits per pixel and batch II, after scaling to the same resolution, consists of 66 images, with an area range of 588 x 625 up to 1573 x 1413 pixels, â 24 bits per pixel. Batch I and II are combined in Dataset A. Another Dataset B, consisting of 270 images, ranging from 656 x 852 up to 3781 x 2762 pixels (24 bit per pixel), was only used in evaluation of our method. In all images, a pixel equals $0.98\mu m$[8] in length.

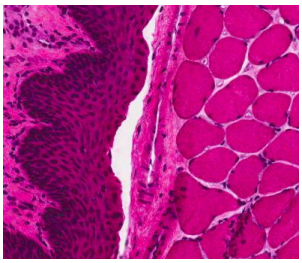
[6]Facebook's Torch, `http://torch.ch/`, 2017.06.29

[7]Microsoft Cognitive Toolkit, `https://www.microsoft.com/en-us/cognitive-toolkit/`, 2017.06.29

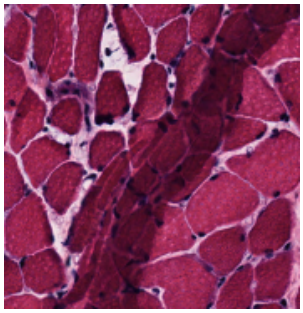
[8]Some images of batch II, and all in Dataset B, had an original resolution of $0.245\mu m$ per pixel, but were rescaled to $0.98\mu m$



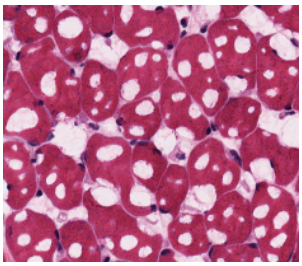
(a) S2-L-PCA-3, bright area in the center and small freezing artifacts (holes within fibers)



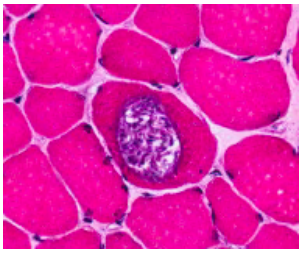
(b) S4-R-PV-3_3, leaking from a vessel at a border region region



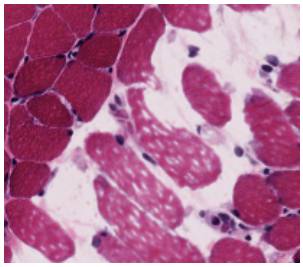
(c) Y2PCAL1_1_002, overlay of fibers



(d) S1-L-PV-3, freezing artifacts 'swiss cheese'



(e) S4-R-PCA-3, bessel within a fiber



(f) Y4 R PCA 1-1_002, stretched fibers

Figure 1.5: An overview of different artifacts throughout datasets. Due to their influence on the dimension of some fibers within an images (i.e. Figure 1.5d, Figure 1.5f) it is difficult to determine if morphometric information of such regions should either be evaluated or ignored.

An inspection of the digitized slides lead us to identify following (segmentation) challenges. The used *HE* staining varied in intensity which developed different coloring (light red to dark purple, Figure C.1 and Figure C.2). Overlapping regions (Figure 1.5c) likely happen while slicing or placing the thin section on a slide. Freezing artifacts (Figure 1.5a and Figure 1.5d) may occur by ice-crystal formation. Further artifacts include cut blood vessels (Figure 1.5b) or other similar colored vessels, bright areas (Figure 1.5a), likely because of a water drop on the slide, and other artifacts (Figure 1.5e or Figure 1.5f). Other obstacles were different resolutions (batch I, batch II) and varying image size (batch II).

Additionally, to assess segmentation with a supervised machine learning approach of *HE* stained slices, we require training data, which is not available. We want to create such, without the need of trained professionals to manually evaluate each pixel. We propose a semi-automatic approach to bootstrap label generation.

1.3 Related Work

A large number of image processing techniques are currently being developed to support medical domain experts in analysis of medical images in diagnoses and prognoses determination [21]. Histological images, obtained from biopsies, benefit from the recent changes in image processing, since repetitious tasks (e.g. counting objects) can be automated [31, 75]. Especially when segmenting such specimen, the difficulty depends on the applied staining [48, 77], the cumulation of artifacts that develop during the specimen preparation and the missing of a standard to evaluate results, since even expert pathologists may disagree [1, 19, 72, 80].

Early, semi-automatic approaches on computer aided segmentation of muscle fibers depend on pixel-intensity and gradient information [11, 16, 23].

Another concept was presented by Castleman et al. [11] who suggests a semi-automated approach based on active contour models. The required manual interaction of this method is a drop-out criterion for our use case since it requires pointing the approximate centroid of each fiber, and our smallest specimen contain close to a thousand fibers.

Building on the previous idea, the state of the art method by Kim et al. [43] proposes a fully automated segmentation where edge based active contours are extended by region based cues (color and texture difference of fiber and surrounding tissue). Depending on the quality of the prepared biopsies, the lack of visible strip-lines to adjacent fibers or low gradients lead to false segmentation results.

The watershed transform, as proposed by Beucher [6], is a region-based segmentation approach. The intuitive idea underlying this method comes from geography: It is that of a topographic relief which is flooded by water and watersheds being the divide lines of the domains of attraction of rain falling over the region. An alternative implementation is the marker controlled watershed [30], e.g. Xu et al. [85] use a template matching and a thresholding method and then the internal marker is determined by performing a distance transform and the external marker by morphological dilation.

By using the Gradient Vector Flow (GVF) of an image, which pulls the active contour towards the object boundary, Mula et al. [55] seek to improve upon active contour modeling. Initial seeds for the muscle fibers are detected within concave ares, after ridge detection enhances the muscle boundaries. The GVF deformable model is then applied to drive the contour to converge to the muscle fiber boundaries. The specimen required an immunohistochemical approach, targeting different antigens to allow identification of the tissue surrounding fibers.

Schenk et al. [63] perform segmentation of the glottis from high speed laryngeal videos for analyzing vocal fold vibrations. The method performs 3D segmentation on a spatio-temporal volume using a geodesic active contour approach extended to 3D and implemented on the GPU for efficient computation.

Interactive Graph Cuts (GC) algorithms, as introduced by Boykov and Jolly [8], aim to find the minimum cut between foreground and background seeds via maximum flow

computation. A user defines hard constraints for segmentation by indicating certain pixels (seeds) that absolutely have to be part of the object and certain pixels that have to be part of the background. The metrication error (blockiness) was addressed in subsequent work on GC by Unger et al. [78].

The usability of these methods is limited for our use case, since they require delineation of fibers or special histochemical stains or partially extensive user interaction.

Current segmentation methods utilize machine learning [36, 56, 58]. They can be split into supervised and unsupervised methods. Approaches for unsupervised methods commonly include clustering (i.e. k-means), neural networks or latent variable models [36] and are independent from labeled training data.

To identify follicular regions by classification of pixels, Sertel et al. [66] used the K-means clustering algorithm with the Euclidean distance considering four classes (follicles, interfollicular regions, lymphocytes and image background). In his later work, Sertel et al. [65] demonstrated that unsupervised segmentation methods can make systems more robust than the supervised approach of K-means, due to diversity in the color spectrum of the images.

The latest impressive segmentation results where achieved by supervised deep learning approaches, which outperformed other methods[9]. This can be seen for different challenges, like the 2012 ISBI 2D EM segmentation challenge [2], where the winning team had no prior experience with EM-images, or current 2016 ISBI challenges, like melanoma detection and identification[10] [32]. In the field of medical imaging, *CNN* (LeCun et al. [46]) have received extraordinary attention. The current scope of application ranges from semantic segmentation (Long et al., Ronneberger et al. [49, 61]), mitosis detection and classification (Ciresan et al., Malon and Cosatto [14, 51]) to blood cell counting (Habibzadeh et al., Xie et al. [33, 84]) to name a few.

The approach of Kainz et al. [41] for the GlaS@MICCAI2015 challenge[11] utilizes a combination of two *CNN* architectures. The first separates glands from background and the second *CNN* identifies gland-separating structures. The neural network prediction is regularized by a weighted total variation criterion which results in a figure-ground segmentation. Tissue classification accuracies of 98% and 94% are obtained on two test sets.

Since the most encouraging concepts require labeled training data, which we lack of, we have to first generate it. In case of only a limited number of training data available, additional synthetic label generation, like Barth et al. [3] and Cordier et al. [18] indicate, can improve the total segmentation results since only a small (manually) annotated dataset is required for fine-tuning. Another benefit of generating synthetic labeled data, is the reduction of labour intensive manual pixel-wise annotation to a minimum.

We target to iteratively improve our segmentation result, therefore we need to develop

[9] `http://brainiac2.mit.edu/isbi_challenge/leaders-board-new`
[10] `https://challenge.kitware.com/#challenge/560d7856cad3a57cfde481ba`
[11] `https://warwick.ac.uk/fac/sci/dcs/research/tia/glascontest/`

our labeled training data. Active learning, as summarized by Settles [67], is a concept which can be applied to different tasks, and generally asks an oracle (e.g., in our case a medical trained person) to label (or improve) training data. Gaur et al. [29] utilize pre-trained networks (copying the weights of the first three layers of a *CNN*) and expert feedback to segment in new domains with limited labeled data.

1.4 Contribution

This master thesis focuses on creating a fast, reliable segmentation for morphometric information extraction of *HE*-stained histological sections of muscle fibers.

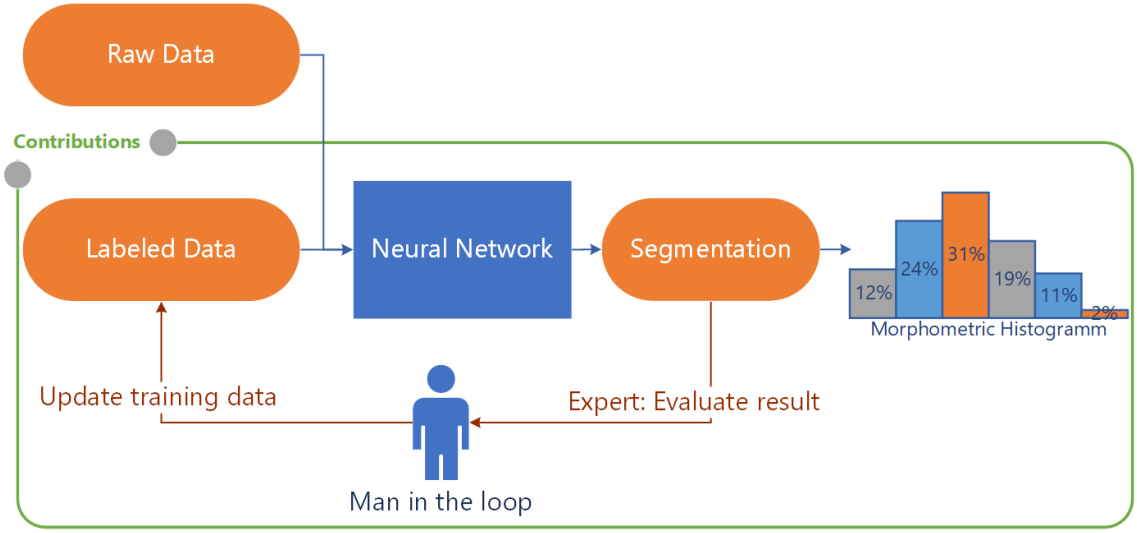


Figure 1.6: Essential segmentation and morphometric information extraction pipeline. This thesis makes contributions in the modules which are within the green component.

Since the data provided by our medical partner only consisted of digitized sections, and the most promising segmentation approaches in the literature require labeled data, we propose a way to construct them. We provide a road map from initial label bootstrapping, while identifying obstacles and proposing solutions, to an automated pipeline for the specific use case. Furthermore we show a viable pipeline for active learning (man in the loop).

We want to identify a feasible, fast and accurate information processing procedure which starts with the common necessity of creating labeled data when working with supervised machine learning approaches. We use and extend a neural toolbox [76] built upon Caffe [39]. By combining several low level segmentation techniques and manual annotation we are able to create initial segmentation results (bootstrapping). Because of the specific image data, we enhanced our Neural Network (NN) [61, 76] to support arbitrary image dimensions (limited by network definition), gain robustness against intensity shifts and

add elastic deformation capabilities to create artificial variations of training data (semantic image data). The parameters can conveniently be set in the *NN* configuration file. We automate the combination of *NN* predictions and watershed transformation to create our initial segmentation (binary ground truth) result and extract morphological fiber information. Thanks to the vastly improved computing speed, the evaluation of digitized slices is a matter of seconds, instead of transferring probes to other laboratories and waiting for results.

The iterative step of improving the resulting segmentation is done in a Graphical User Interface (GUI), which we provide to our histological experts in combination with the current segmentation outcome. Furthermore the *GUI* features statistical information extraction, which allows (grouped) display of extracted morphological information over sets of images. This leads to the following contributions:

- A process to create an initial segmentation (bootstrapping)
- A fast, fully automatic pipeline to segment, detect and extract statistical data from muscle fibers
- A tool to manually refine segmentation to improve the automated results and evaluate morphometric information

This thesis summarizes the developed approach for detecting muscle fibers, evaluates the detection algorithm based on supervised neural networks and qualitatively presents results on muscle fiber segmentation.

1.5 Outline

At first this thesis will explain the decision making by reviewing and comparing results of other image segmentation techniques (Chapter 2.1) and discuss obstacles which we want to overcome. After defining the processing-pipeline we will go into detail. The chosen machine-learning segmentation approach requires labeled training data, which our initial data lacks. In Chapter 2.2 we overcome the absence of labeled data with a semi-automatic process, where we initially use different classes for artifacts, fiber-centers and background, train an *NN* and combine the predictions to generate a binary segmentation output. Building upon that, in Chapter 2.3 we can reduce pre- and post-processing requirements. The reduction of complexity simplifies the integration of an iterative update that can be seen in Chapter 2.4 where experienced professionals further improve the neural network output and therefore the extraction of morphological information. In Chapter 3 we experimentally evaluate and discuss our concept and give an outlook to potential enhancements.

2 An Automated Muscle Fiber Segmentation Approach

Contents

This thesis aims for morphologic information extraction from histological muscle fiber slices. This chapter presents our contribution in segmentation and information extraction and puts it into context with existing approaches.

2.1 Approach

To ease evaluation of fiber size, we require a processable segmentation image. As a minimum requirement we need to distinctively identify fibers. Therefore, we initially compare different methods, like threshold (adaptive threshold is ignored, since microscopic images are evenly illuminated), watershed and total variation segmentation (see Figure 2.3).

After identifying the obstacles and limitations of these image processing operations, we propose a machine learning approach. Deep Neural Networks (DNN) promise extraordinary success in the field of image segmentation [2, 13, 44, 49]. A drawback is the required training data which depend upon labeling.

We demonstrate that distinct fiber identification is feasible and propose a bootstrapping process to create initial labeled data, without exhaustive manual labor.

Furthermore, we offer a way to include expertise of trained medical personnel to provide active learning input to iteratively improve the segmentation results in a simple active learning strategy.

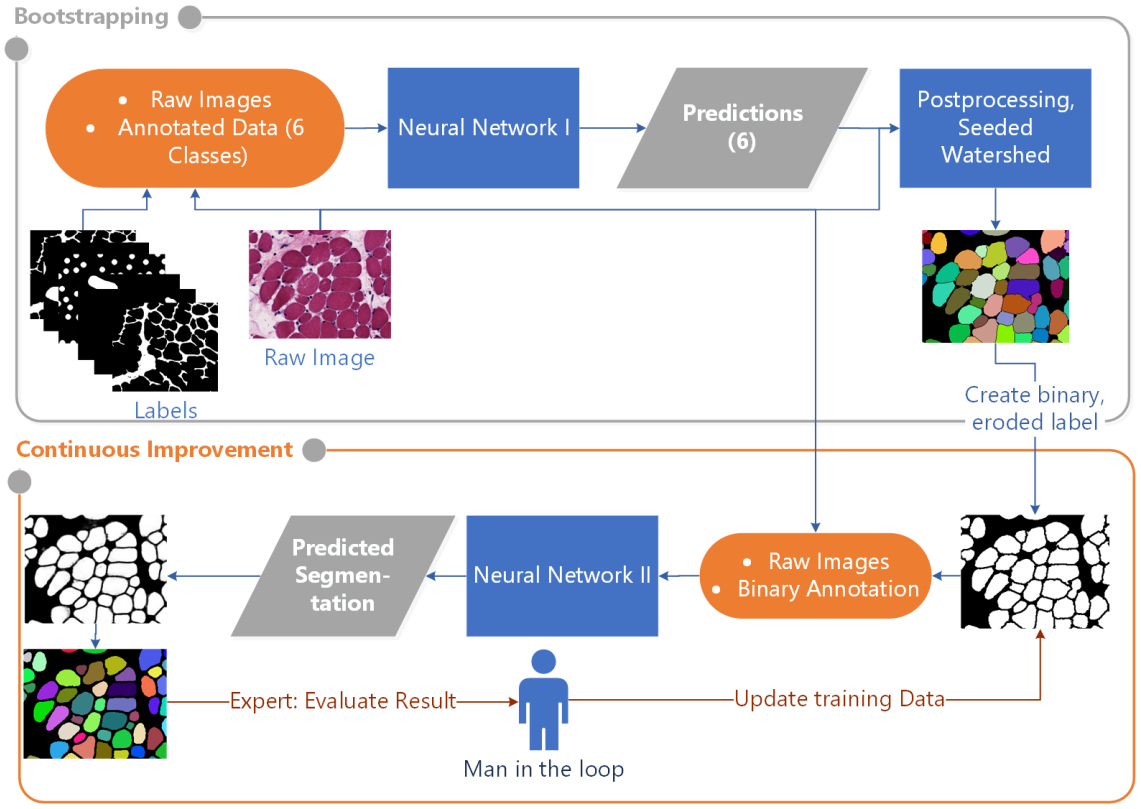
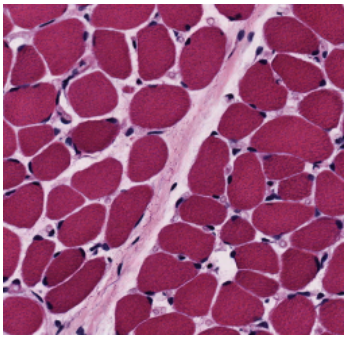


Figure 2.1: Our proposed approach consists of two steps: In the bootstrapping component (gray) the process for initial label generation is displayed. In the orange component the iterative pipeline for binary segmentation using a Neural Network (NN) is illustrated. The man in the loop symbolizes a supervised update of the binary labels, making most effective use of costly expert intervention.

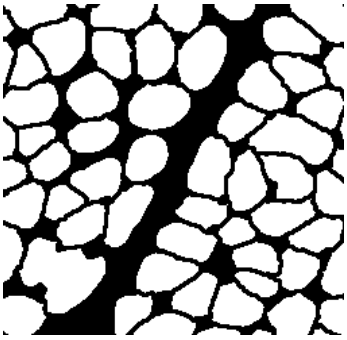
Our proposed solution includes two major steps to automatically segment and continuously improve segmentation results. The first is necessary to bootstrap labeled data for machine learning and can be omitted afterwards (top, gray container in Figure 2.1). The benefit of the first step is the great simplification of conditions for the second step (orange container in Figure 2.1) through reduction of the problem complexity by reducing the number of prediction classes from six to two.

Since the provided data only consists of histological fiber slides and no labels for them, the first step was to create labels from dot annotations (top, gray container in Figure 2.1). After examining the provided images, six label classes were identified (Figure 2.8) and annotation (labeling) was applied on a subset of the provided image set (label image). The matching pair of raw- and label image is used to train *NN* I. Thereby it can predict those classes on the remaining set. We combined the predictions into fiber area, fiber background and fiber centers and used them in the post processing step, where the output is a watershed transformed image, which already roughly identifies fibers. Further

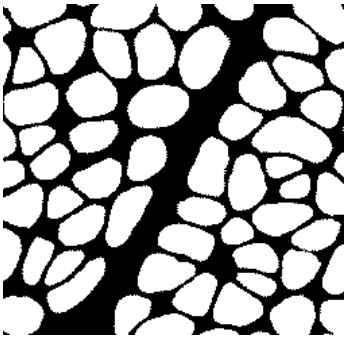
processing of these images provides a binary segmentation (fiber - white, no fiber - black, Figure 2.2b).



(a) Unmodified raw image



(b) Bootstrapping segmentation result



(c) Continuous Improvement segmentation result

Figure 2.2: Visual difference of bootstrapped- and continuously improved segmentation. (a) sample region (from Y6 L VOC 1-1_002). (b) is the result of watershed applied on center, background and artifact predictions in conjunction with (a) that displays serrated borders, whereas in (c) watershed was applied on the pixelwise segmentation prediction. A visual improvement of the topography can be seen from (b) to (c) i.e. when comparing the bottom left images.

Even though the created binary labels contain an error (i.e. serration at contours, Figure 2.2b), they are used in conjunction with the histological slides to train a second *NN*. The benefit of this process is a direct segmentation prediction, the smoothing of errors (i.e. serration and topographical error, Figure 2.2c) and an immensely reduced pre- and post-processing. This baseline segmentation result is achieved without the input of any domain expert.

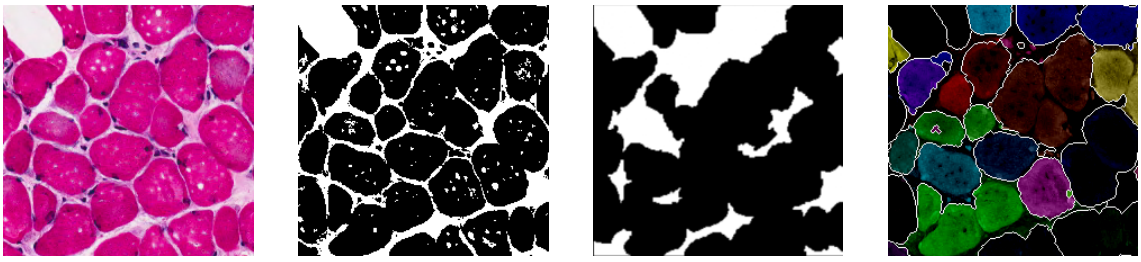
The processed prediction output in Figure 2.2c is a review candidate for a medical expert and continuously improved if necessary (Continuous Improvement component in Figure 2.1).

2.1.1 Obstacles for Segmentation

In collaboration with our medical partner we identified the following properties within Haematoxylin Eosin (HE) images.

- Fibers tend to have a gradient which helps to identify contours
- Boundaries are occasionally not clearly identifiable
- Artifacts are present

To evaluate an appropriate segmentation we investigated different image processing methods and their deficiencies, in regards of distinct fiber detection and identified hurdles, which lead us to a machine learning approach.



(a) Unmodified raw image S2-L-PCA-3_2,

(b) S2-L-PCA-3_2, Converted to gray-scale image$_{[0-255]}$ and applied threshold $t = 128$. Freezing artifacts (white dots) are visible.

(c) S2-L-PCA-3_2, TV-L1 supervised segmentation where foreground and background seeds are defined ($a = 10$, $b = 0.55$, $\lambda = 0.5$)

(d) S2-L-PCA-3_2's red channel *RGB* is used to create distance transformation for the seeds of the watershed

Figure 2.3: Review of segmentation methods using the raw image. (a) sample region (from S2-L-PCA-3_2), source for segmentation methods. In (b) thresholding was applied, resulting in visible freezing artifacts and (some) consolidated fibers. More computing intensive operations (c) or (d) yielded unsatisfying results as differentiation between fibers is even more arduous.

Thresholding To create a binary image of Figure 2.3a, the image is converted to gray-scale using OpenCV's RGB to Gray conversion[1]. In Figure 2.3b a fixed-level threshold to the previously created converted gray-scale image is applied. The automatically generated results, like in Figure 2.3b allow distinction of fibers up to a certain degree, but suffer from elementary classification error when artifacts occur. Freezing artifacts are labeled 'non-fiber' but artifacts in between fibers may be identified as fiber, depending on their gray-value.

Total Variation Segmentation The total variation segmentation is based on minimizing continuous non-smooth energy functional $E_{seg}(u)$ [35, 78]. It is a minimal surface segmentation approach formulated as

[1] `https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html`, 2018.05.28

$$\begin{aligned} \min_u E_{seg}(u) = \min_u \int_\Omega g(\mathbf{x})|\nabla u(\mathbf{x})|dx + \lambda \int_\Omega u(\mathbf{x}) \cdot w(\mathbf{x})dx, \\ \text{s.t.} \quad u \in C_{box} = \{u : u(\mathbf{x}) \in [0,1], \forall \mathbf{x} \in \Omega\} \\ \text{and} \quad g(\mathbf{x}) = e^{-\alpha||\nabla I(\mathbf{x})||^\beta}, \quad \alpha, \beta > 0 \end{aligned} \tag{2.1}$$

where

Ω ... image domain
u ... $u \in C^1 : \Omega \implies \mathbb{R}$ is smooth
g($\mathbf{x}$) ... Geodesic Active Contours (GAC) energy as edge function
λ ... trade off between data term and weighted TV semi-norm
w ... weighting map $u = \begin{cases} w < 0 & \text{foreground} \\ w > 0 & \text{background} \\ w = 0 & \text{second term vanishes, pure } GAC \text{ is computed} \end{cases}$
$\nabla I(x)$ ... gradient of the input image

The segmentation process works best, when a representative foreground and background region is marked. Due to the optimization of $E_{seg}(u)$ and occurring weak borders (or too narrow gaps) between fibers, the resulting contour leads to a segmentation where distinction of fibers is impossible (see Figure 2.3c).

Watershed The watershed transformation (see Figure 2.4 or Figure 2.3d, [54, 79, 85]) uses a topographical concept in analogy to pouring water from distinct sources (local minima, markers) that increase altitude in a topographical map with constant speed.

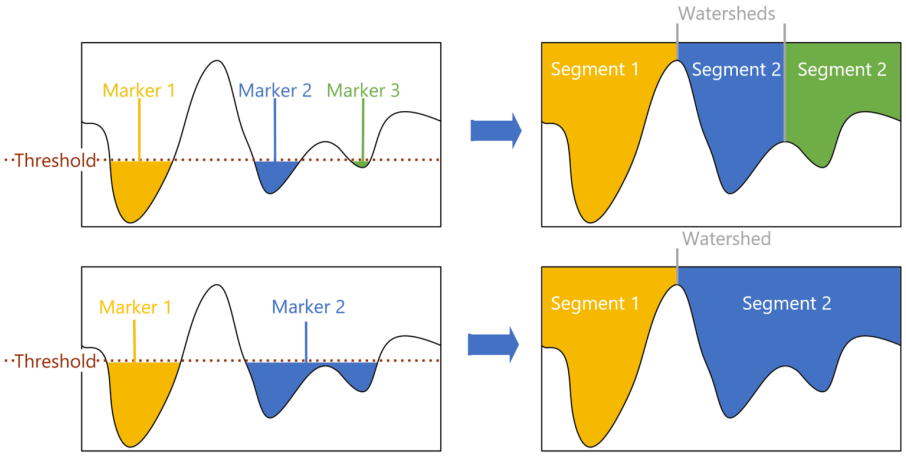


Figure 2.4: Two examples of the watershed transform applied to a 1-dimensional signal (e.g. intensity of a row of an image). Top row: Three distinct markers (seeds) are identified (see threshold level) which yield three segments divided by watershed lines. Bottom row: With a higher threshold, two markers are identified, segment 2 floods over the previously disjunctive peak and into the neighboring trough.

The watershed transformation can be formulated by evaluating the topographical distance T_f of an image f [54, 60].

$$\begin{aligned} T_f(p,q) &= \inf_{\gamma} \int_{\gamma} ||\nabla f(\gamma(s))||\, ds, \\ CB(m_i) &= \{x \in \Omega | \forall j \in I \setminus \{i\} : f(m_i) + T_f(x, m_i) < f(m_j) + T_f(x, m_j)\}, \\ \text{and} \quad Wshed(f) &= \Omega \cap \left(\bigcup_{i \in I} CB(m_i) \right) \end{aligned} \tag{2.2}$$

where

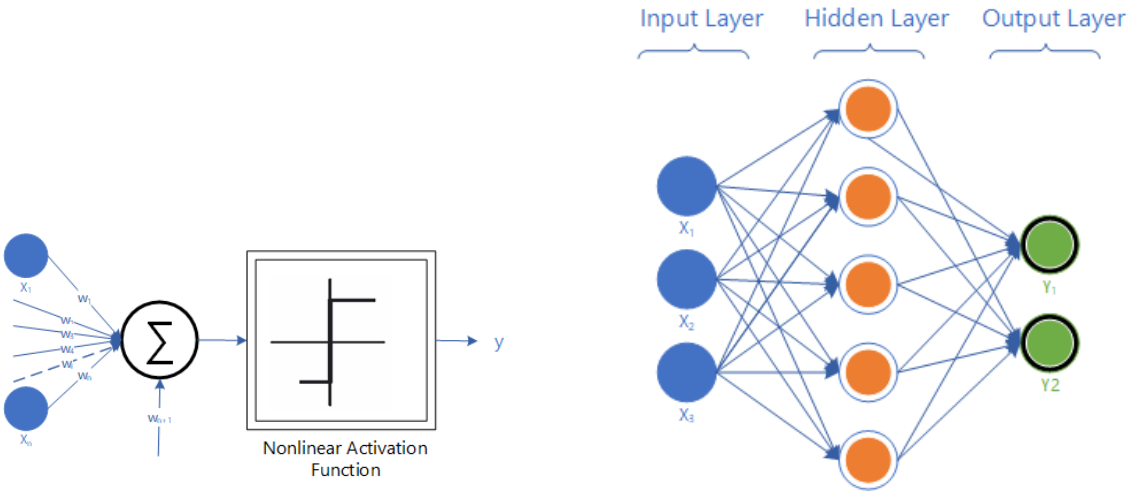
Ω	... image domain
$\gamma(s)$	... path (smooth curve) inside Ω
m_k	... a minima from $\{m_k\}_{k \in I}$ of f for some index set I
$T_f(p,q)$	... topographical distance between points $p, q \in \Omega$
$CB(m_i)$	... catchment basin: set of points topographically closest to m_i
$Wshed(f)$	... set of points which are not part of any catchment basin

The path with the shortest T_f-distance between two points (p,q) is a path of steepest slope.

In Figure 2.3d we used the red color channel of Figure 2.3a, since it contained the most relevant features. On that channel we applied a distance transformation to identify markers (fiber centers). With them as seeds, the watershed transformation could not avoid the merging of fibers in the resulting segmentation, since the gradient profile missed some

edges (leaking) or the identified initial markers already merged several fibers. Results were worse for images with other red intensities, since the applied threshold (on the distance transformation image) for markers strongly depends on it.

Neural Networks The idea of *NN* was embraced from natural biological neural networks (connected neurons) [21, 69].



(a) Basic computational neural element for classification

(b) An example of a feed forward neural network with 3 input nodes, a hidden layer with 5 nodes and a final output layer with 2 nodes

Figure 2.5: Neural Network Structure: (a) shows the computational output (y) of a neural element with an exemplary additional biased input (w_{n+1}). (b) shows a simple feed forward network structure.

The computational output of a neural element can be expressed as

$$y = F\left(\sum_{i=1}^{n} w_i * x_i + w_{n+1}\right) \tag{2.3}$$

where

F ... is the (non-linear) activation function (e.g. sigmoid)
w_i ... is the weight of the respective input (x_i or 1 in case of bias w_{n+1})

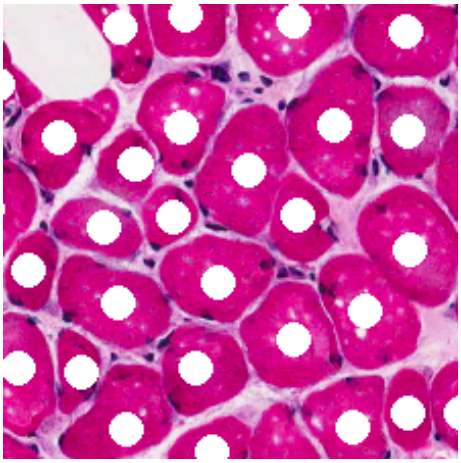
It is an interconnected assembly of simple processing elements (input layer, n-hidden layers, output layer as in Figure 2.5b) whose functionality we describe later. Convolutional Neural Networks (CNN) use convolution in at least one of their layers. Deep Neural

Networks (DNN), as their name suggests, have a deep structure, i.e. they have many hidden layers.

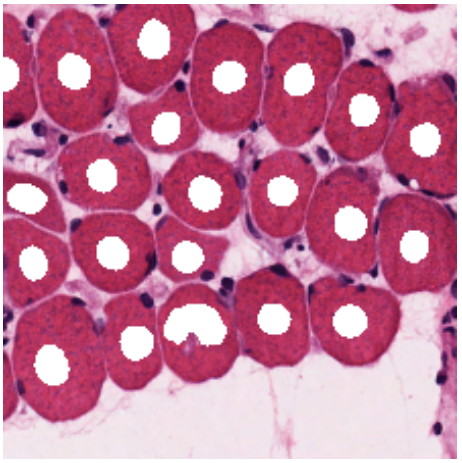
In the training phase the connections between layers receive a weight adaption, to reduce the error (loss) on the training data. After the training phase (weight calculation of neurons within the *NN*), they have the capacity to create predictions for various input images. In our case, after successful training, we want to create a ground-truth with an arbitrary number of new images.

For evaluation of the neural network approach, we use the neural toolbox as provided by Tschopp [76], built upon Caffe [39]. It offers a configuration interface to load the label and raw images for (patch) training. Additionally we can rely on template configurations for training and processing. A drawback is the required training data which requires labeling (Figure 2.6a).

To verify if a distinction is in principle achievable, we created training data where we annotated fiber-centers and trained a neural network model (Figure 2.11).



(a) S2-L-PCA-3_2, *ROI* with overlay of fiber-center label (white circles)



(b) S3-R-PV-2, *ROI* of example test image with overlay of fiber-center prediction $p(x) \in [0, 1]$ (applied T at 0.5)

Figure 2.6: Training-set consisted of 41, test-set of 55 images. Annotation was done manually. Qualitative result evaluation hints at a strong distinct fiber identification.

Our applied machine learning method shows that distinct fiber identification is feasible with fiber centers (Figure 2.6b, 3.2.1.2 Experiment), which is a mandatory step for an initial ground truth.

Summary Even though thresholding, total variation and watershed transformation are commonly used in image segmentation problems, they cannot be applied directly on the

HE images as the variations of color intensities, artifacts or dissolving boundaries prove to create erroneous segmentations. On the other hand, the prediction result of the neural network, which indicates that distinct fiber identification is possible, is promising (Figure 2.6), but does not tackle the problem of fiber boundaries or artifacts. The results demonstrate that a combination of watershed transformation with the detected fiber centers will reduce the watershed initialization error (markers) but the issue of dissolving boundaries, especially in artifact and no-fiber regions, remains. To avoid that we require identification of such regions which is a very similar problem to the distinct fiber identification problem which we solved. This leads to our segmentation bootstrapping approach.

2.2 Segmentation Bootstrapping

To bootstrap for fiber segmentation, we use our previously generated center annotation (which we used to evaluate the *NN* approach, see Figure 2.6) and extend it by identifying non-fiber regions through automatic thresholding with an additional rough manual review. Further, we annotate artifacts, which are only within a limited area of an image or not present at all (Table 2.1), thus they can quickly be manually identified. These annotation classes require no professional knowledge about muscle fiber properties and can be done by a nonprofessional. The trained *NN* (with modifications of [39, 76]) makes it possible to create predictions of the labels on untrained histological slides. By combining the output classes correctly, a marker based watershed is used to create a binary segmentation baseline.

The process is divided into the following blocks (Figure 2.7).

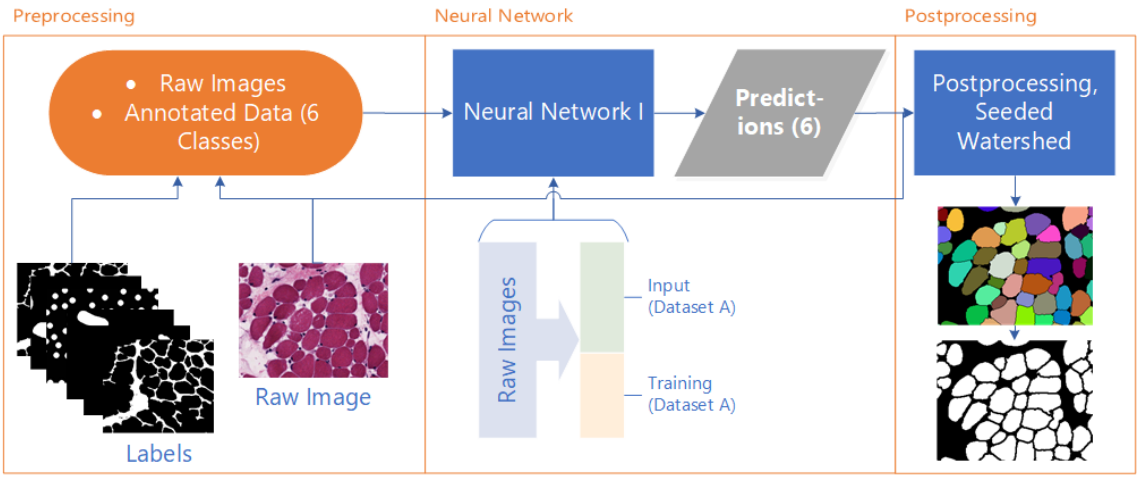


Figure 2.7: The segmentation bootstrapping pipeline consists of three steps: First, annotate a subset of all available images. The next step includes training of the annotated images and predictions on untrained. Finally the predicted classes are combined to apply the watershed algorithm.

1. Preprocessing: Annotation of labels for a subset of the provided slices. The extensions of labeling, by adding artifacts and background, targets to improve the segmentation result by providing additional information about invalid fiber regions.

2. Convolutional Neural Network: Learn from annotation to create prediction for unknown histological slices

3. Postprocessing: Combine Convolutional Neural Networks (CNN) predictions to apply watershed transformation and export binary image as segmentation baseline.

The labor intensive part of directly annotating fiber outlines was transformed into a partly automatic annotation. Some label-classes require manual labor, like fiber centers or artifact labeling (not present in all images, see Table 2.1) which is a trivial task that can be performed without histological knowledge by a layman. We expect our manual annotations to support the creation of an annotation baseline where distinction of fibers is possible.

The benefit of training a neural network using these annotations is that whenever new digitized histological sections are provided from our medical partner, we avoid the time intensive work of annotating since we can use the prediction output of the *CNN*. Additional labeling is only required, if a variation is not covered in the annotated images or image preprocessing in our modification of the *CNN* (e.g. further unknown artifacts).

Through our chosen label classes we can set the focus on targeting information (centers, artifacts, background), which allows us to set contour restrictions for the post-processing. Furthermore, it is already possible to extract morphological information from these images (Figure 2.19b).

2.2.1 Preprocessing

In preprocessing, a total of 41 images got annotated (sub-set of dataset A, see center of Figure 2.7). Annotation of each class has an error margin, since either the affiliation of a region, or the determination of constraints (i.e. fiber centers, contours of overlapping regions) or combinations of both, can be ambiguous tasks [1, 19, 72, 80].

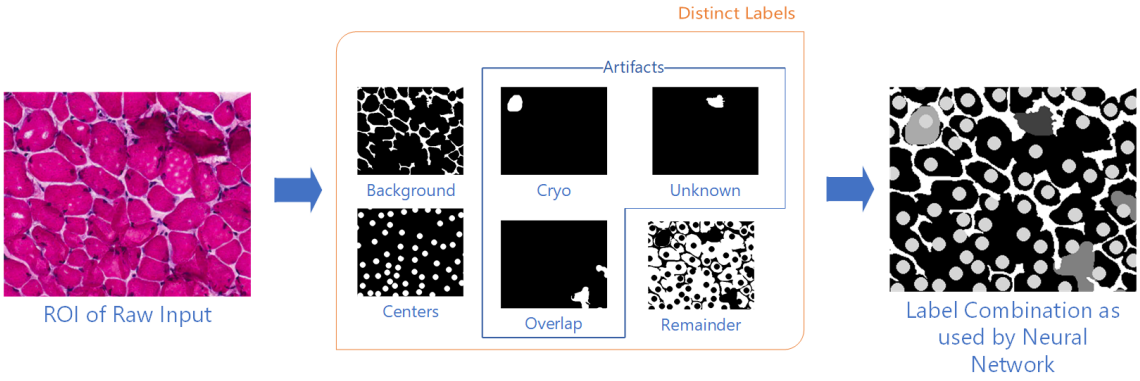


Figure 2.8: On the left a region of interest (ROI) of a raw image (from S1-L-PCA-3) as provided by our medical partner. In the center the distinct labels, which classify an image into background C_{BG}, centers C_c, artifact-cryo $C_{A,C}$, artifact-overlap $C_{A,O}$ and unknown artifacts $C_{A,U}$ are displayed. *Remainder* is the initial white label image, where all classes are subtracted from. The last image shows the combination of all classes as used for training.

The combinations of all labels is done by creating an output image D (Equation 2.4) which combines masks of each class C_i (Figure 2.8) with different intensities (i_c) to distinguish them. Priority of labels is determined by the order in which they are added, where the last one overrules others (Equation 2.4). The neural toolbox [76] prefers to draw patches where all classes are equally distributed, therefore a focus on rarely occurring classes is automatically implemented.

$$\begin{aligned} D = I_{white} \vee M_{A,U} \vee M_{A,O} \vee M_{A,C} \vee M_C \vee M_{BG} \qquad \text{if } M(x) > 0 \\ \text{and} \quad M_c = C_c \cdot i_c, \qquad \text{where} \qquad C(\mathbf{x}) \in \{0, 1\} \end{aligned} \tag{2.4}$$

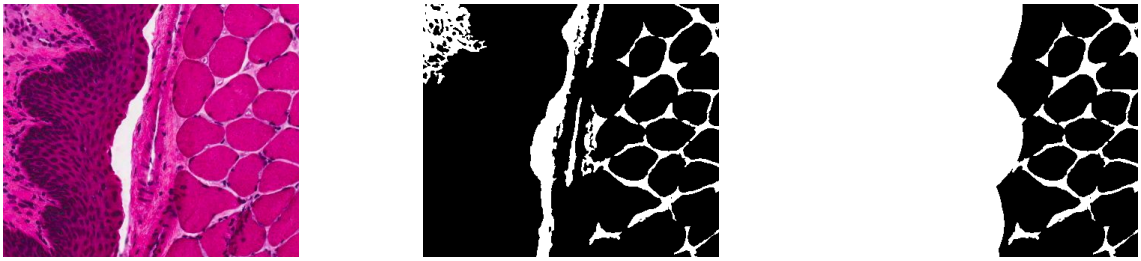
where

D ... destination image
I_{white} ... white initialized label image
C ... labeled binary image
M ... mask used to add label
i_c ... id to identify added mask
x ... coordinates of the particular image.

By applying Equation 2.4 in this order (left to right), we force that fiber centers M_C are never on a background class M_{BG}, but may occur in artifact-regions (Figure 2.8, right). Since there is only one class per pixel, the last added class overwrites any previously set.

The annotations in Figure 2.8 are individually stored and were created either automatically (Remainder), manually (Center C_c, Cryo-artifacts $C_{A,C}$, Unknown-artifacts $C_{A,U}$ and Overlap-artifacts $C_{A,O}$) or were a combination of both (background C_{BG}).

Background Annotation Background (a total area of $26,35\%$ over annotated images) was set by converting images to gray-values and setting a fitting T depending on color intensity. In case of some images, where blood vessels leaked (i.e. border regions, see Figure 2.9a) a quick annotation update, i.e. with a pencil tool as provided by image editing software like GIMP[2], was done.



(a) Unmodified raw image

(b) Automated threshold

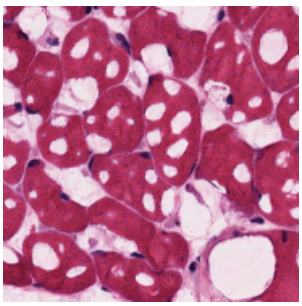
(c) Manual refined threshold

Figure 2.9: An example region (from S4-R-PV-3_3) to illustrate issues with automated background labeling e.g. a cut blood vessel in (a) (left). (b) displays the large erroneous region. (c) shows the manually updated annotation.

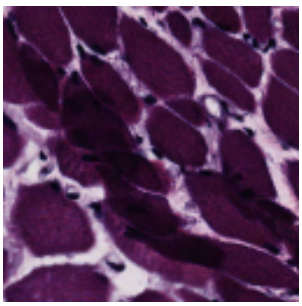
This manual step was only done for a limited number of images where threshold-background-labeling reached its limits and false detected regions were larger than *average fiber size* $*2$, since too detailed relabeling would contradict the goal of reducing manual labor.

Center Annotation The manual task of center labeling (a total area of $12,43\%$ over annotated images) was the most time consuming and repetitious task. Starting with a fixed diameter of 14 pixel variations including diameter of 1 pixel up to 20 pixel were tested. Too small areas reduced the number of found fiber centers whereas too large areas resulted in merged centers.

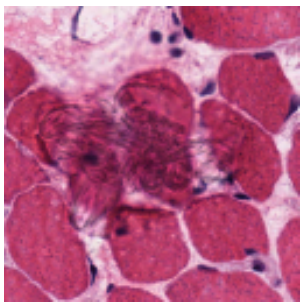
[2] `https://www.gimp.org/`, 2018.05.29



(a) An example region (from S1-L-PV-3) that displays cryo artifacts that may result while freezing.



(b) An example region (from Y1 VOC R 1-1_001) displaying overlapping fibers.



(c) An example region (from S5-R-PV-2) displaying artifacts, that are of unknown origin.

Figure 2.10: Exemplary regions that display artifacts which were sub-classified into cryo artifacts (a), overlapping artifacts (b) and unknown artifacts (c). The difference in color intensities is due to variations in the staining process (dehydration step).

Cryo Artifact Annotation The most common reasons for freezing artifacts are either a too high water content of the tissue or a too slow freezing process [12, 53]. Some of our data have such artifacts (see Figure 2.10a) to varying degrees (this reflects the learning curve of the morphologist preparing the specimen). We target to identify fiber morphology changing cryo-artifacts, to eliminate them from evaluation. Therefore, we mark the whole fiber as artifact (a total area of $0,51\%$ over annotated images), not just the part which has freezing artifacts.

Overlap Artifact Annotation Overlap summarizes wrinkling-, curling, nicks in tissue (see exemplary in Figure 2.10b), alternating thick and thin sections. We try to label these regions (a total area of $0,31\%$ over annotated images) since distinct fiber identification is error prone.

Unknown Artifact Annotation A combination of artifacts where the reason or origin are unknown to us (see exemplary in Figure 2.10c) and we know that they would influence the extracted morphometry (a total area of $0,17\%$ over annotated images).

Remainder All annotations get combined on an initially white image, therefore the remaining white area is the discount of the other classes (valid fiber bodies without centers, a total area of $60,23\%$ over annotated images).

Class Distribution per Annotation	C_R	C_B	C_C	$C_{A,C}$	$C_{A,O}$	$C_{A,U}$
	[%]					
Average	60,23	26,35	12,43	0,51	0,31	0,17
Average*	60,50	26,62	11,89	0,58	0,23	0,18

Table 2.1: Label distribution of ground truth which is used for bootstrapping the segmentation baseline (Average). Left to right by frequency. Labels that are used in training the bootstrapping neural network are excluded as training images for the binary segmentation baseline. Average*: Distribution for the binary segmentation baseline.

2.2.2 Convolutional Neural Network

We use the U-Net model as described by [61] for our CNN layout. Its special U-like structure is eponymous (Figure 2.11). U-Net involves a contracting path with two convolutions followed by a max pooling layer on each level. In the expanding path, compressed features from the contracting path are expanded again by deconvolution over the same number of levels as in the contracting path. Additionally, in each level of the hierarchical U-Net structure, the contraction path layers are copied to the deconvolved expansion path layers, to incorporate the original feature maps as well. Built on Caffe [39], we use the neural toolbox [76] providing patch-to-patch segmentation in an end-to-end manner (see right in Figure 2.8). We enhance this toolbox to support arbitrary image dimensions (limited by network definition and available GPU-memory), add robustness for intensity shifts (HE staining) and provide image deformation to synthetically augment the training dataset. We change the U-Net model to train for six different annotation classes which generate six prediction images. The layers[3] used in the U-Net structure are:

Convolutional Layer Convolves the input image with a set of learnable filters where each produces one feature map in the output image.

Pooling Layer Reduces the spatial size of the representation (reduces the amount of parameters, computation and overfitting). In case of max pooling, it takes the most responsive node of the given ROI.

Upconvolution Layer Halves the number of feature channels and concatenates with the responding feature channel (copy and crop layer, [61, chap. 2])

Other (Copy and Crop) Layer Crops the feature map from the contracting path to the extracting path of the same height.

Rectified Linear Unit (ReLu) Layer Computes the output as x, if $x > 0$.

[3] `http://caffe.berkeleyvision.org/tutorial/layers.html`, 2017.07.11

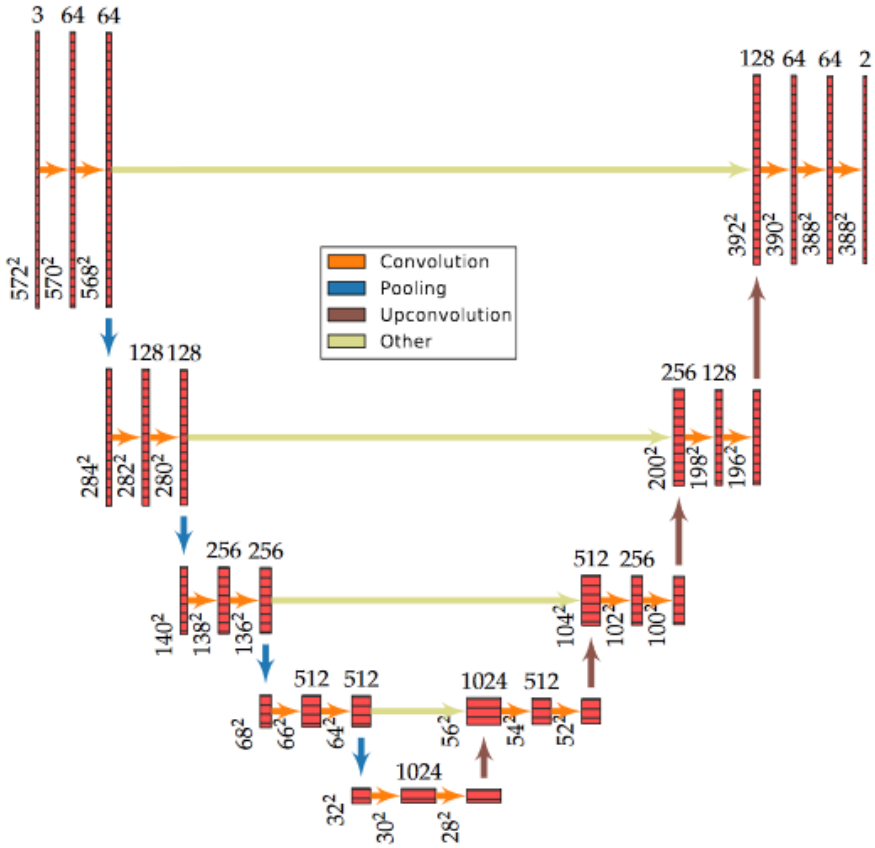


Figure 2.11: U-Net Model, as described by Tschopp [76, chap. 3.4]

2.2.2.1 Enhancements to the neural toolbox

The neural toolbox [76] supports a wide range of image input and output formats, pre-processing parameters and is available at Github[4]. It draws patches out of a provided image-set of a dimension that fits the used network model (e.g. U-Net: Figure 2.11). A benefit of the patch based approach is that the image can be of arbitrary dimensions, since the resulting prediction patches of an input images are stitched together to match the input image dimension. The patch-size lower limit is restricted by network definition and the upper limit by available memory. The preferred patches have an even distribution of the annotated labels, which helps to train to distinguish them. The major benefit of that process is, that labels (e.g. artifacts) which are barely available will be drawn with a higher probability than others [76, chap. 4.3]. We created a fork of the neural toolbox which allowed us to extend the functionality[5] [6]. A limitation, which we want to avoid, is

[4] `https://github.com/naibaf7/caffe_neural_tool`, 2017.07.11
[5] `https://github.com/pkainz/caffe_neural_tool`, 2017.07.11
[6] `https://bitbucket.org/derKlaus/`, 2017.07.11

that all training images need to have the same image size. Furthermore, in regards to our segmentation task, we want to gain robustness for varying staining intensities. Therefore we add augmentation functionality to the training data. As a second enhancement to artificially increase our training data we add image deformation capabilities.

Different Image Size The original toolbox implementation did not support a varying image size, which is a necessity if we want to avoid cropping our images (see different sizes of Figure C.2). A modification to the original source allows different image sizes as long as they meet the minimum criteria for the U-Net model.

Varying Intensities *HE* stained slices vary in their hue-characteristics depending on the staining process. To gain robustness, we converted *RGB* images to a Hue Saturation Value (HSV) cylindrical coordinate representation [27], thereby we can modify the hue value. For every drawn patch out of our training raw images we use a random variation of *Hue (h)* as described in Eq. 2.5 (i.e. Figure 2.12).

$$\mathrm{h} = \mathrm{h} \pm random(0, h_{max}) \quad \text{where } h_{max} = 30 \tag{2.5}$$

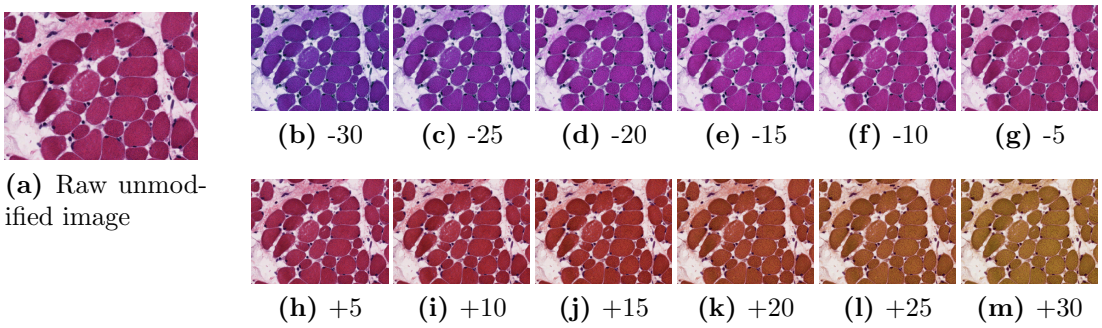
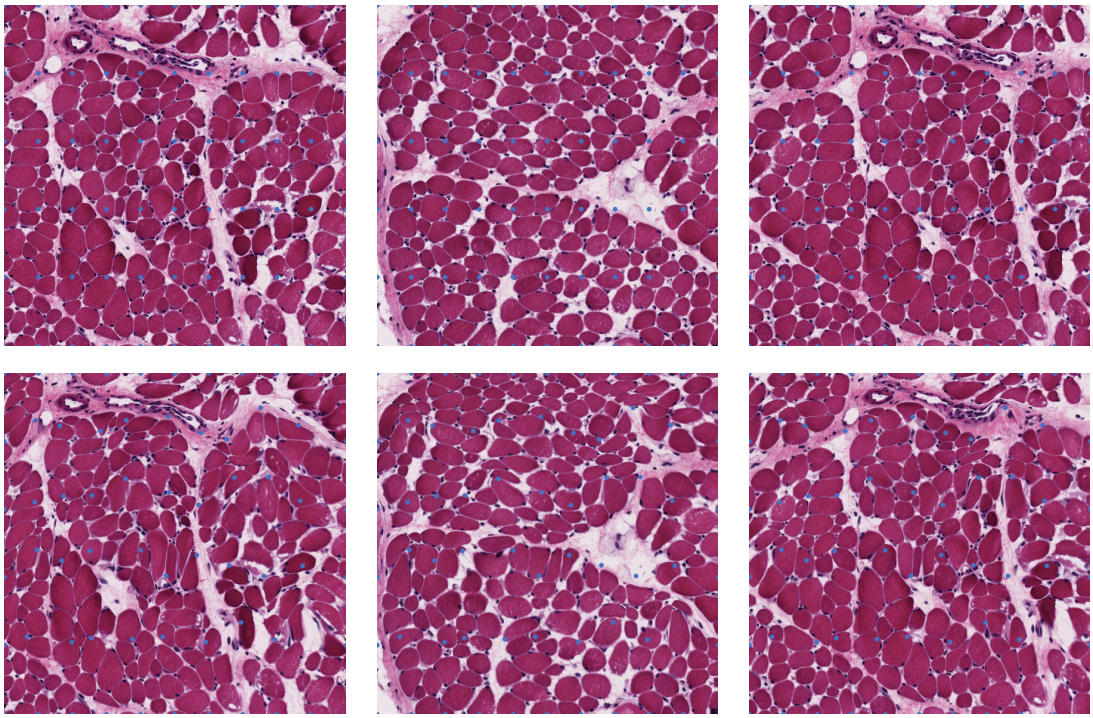


Figure 2.12: Illustration of the effect of manipulating the hue value (region from Y3 L VOC 1-1_002). We gain robustness to staining induced variations in intensities, as we artificially recreate the diversity ((b) to (m) with their particular hue variation)

With this modification we are able to cover all experienced distributions within one patch in regards to known variations in our dataset, by simply modifying our introduced $intshift$ parameter in the neural-toolbox's train configuration file (for configuration see B.1.2).

Image Deformation Ordinarily, it requires thousands of images for training deep neural networks. Differentiation for fiber classification can be influenced by small areas within our training set (e.g. see label distribution table 2.1) whereas other regions are no contribution as source of information. Image deformation provides us an easy access to introduce

a focus on areas, where our approach performed poorly, by adding reviewed data of regions of interest to the training set. Additionally, by drawing random patches out of an image, using existing rotation (by a multiple of 90° degree) or mirror functionality (which are functionalities of Caffe [39], respectively of the neural toolbox [76]) we can artificially enhance our training dataset. Especially for our use case of continuously improving the segmentation result, where we expect to only gain limited modified annotation patches from an already limited updated image set, we expect it to improve our results. The deformation is applied to the input raw patch and correspondingly to the label image.



(a) Affine *MLS* **(b)** Rigid *MLS* **(c)** Similarity *MLS*

Figure 2.13: Illustration of implemented different deformation methods. Top-row shows initial 50 points (Rows = 10, Cols = 5), evenly distributed in a random drawn patch from Y3 L VOC 1-1_002. Border regions have fixed positions (see top and bottom row) to avoid artifact creation because of missing image information. Bottom row shows results from different transformation methods, with random variation in the destination point movement. The maximum variation of a destination point here is half of the source point distance.

We use linear moving least squares deformation (Equation 2.6) that targets to produce globally smooth deformations [62]. We use collections of points which are randomly set (within limits defined by the user see Figure 2.13) to control the deformation.

$$\begin{aligned} \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 \\ w_i = \frac{1}{|p_i - v|^{2\alpha}} \\ \hat{p}_i = p_i - p_*, \quad and \quad \hat{q}_i = q_i - q_* \end{aligned} \tag{2.6}$$

where

M ... linear matrix
w_i ... weights, dependent on the point of evaluation v
v ... point of evaluation
p_i ... control point
q_i ... deformed target point
p_*, q_* ... weighted centroids

The applied geometric transformation of each control point in $\mathbb{R}^n$ are maps $\mathcal{F} : \mathbb{R}^2 \to \mathbb{R}^2$ of the form

$$M = RHST = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \tag{2.7}$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 & \delta x \\ 0 & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix}$$

where

M ... transformation matrix combined of R, S, H, T
R ... rotation matrix
S ... scale matrix
H ... shear matrix
T ... translation matrix

Not applied transformations are replaced by the identity matrix. The parameters are *mode* which defines interpolation mode to be either *MLS*-similarity[7], *MLS*-rigid[8] or *MLS*-affine[9] [62] (Equation 2.7), *cols* and *rows* whose intersections define the contact points to be moved and a maximum *variation* $\in [0, 1]$, where 0 is equivalent to no deformation and at 1 contact points could overlap (for configuration see B.1.2).

[7] 4 Degree of Freedom (DoF): Translations, Rotations and Uniform Scaling
[8] 3 *DoF*: Translations and Rotations
[9] 7 *DoF*: Translations, Rotations, (Non-Uniform) Scaling and Shearing

2.2.3 Postprocessing

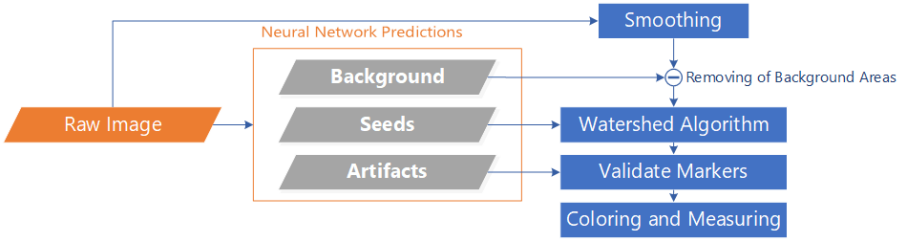
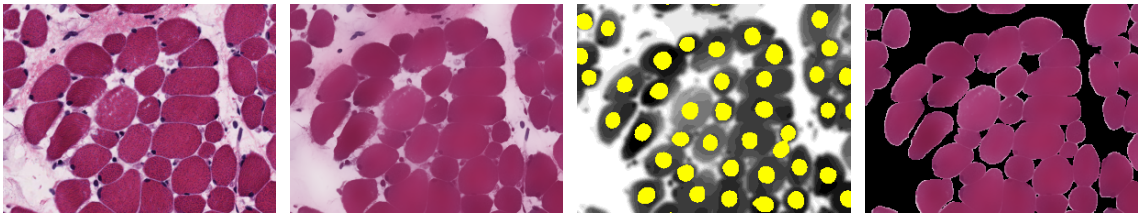


Figure 2.14: Postprocessing steps from trained neural net to extracted morphological fiber information.

The generated predictions by the *CNN* require further processing (2.14) to create a segmentation which separates individual fiber instances. Therefore we combine the prediction output with the smoothed raw image.



(a) Unmodified raw image **(b)** Applied bilateral filter **(c)** Topographic image with seeds (yellow) **(d)** Smooth image with removed background

Figure 2.15: An example region of our histological sections (from Y3 L VOC 1-1_002). (a) Raw input image. (b) Edge-preserving smoothing results. (c) Thresholded fiber-center prediction from U-Net, used as seed input for watershed segmentation. (d) Input image for watershed segmentation with plateaus (black) to avoid leaking.

The first step to prepare for watershed segmentation is smoothing the input image, while still preserving important edge information. For this step we use a bilateral filter, which provides a nonlinear filter in both, the photometric and the geometric domain (see Figure 2.15b). We define a range kernel for smoothing differences in intensities with a size of 150 and the spatial kernel for smoothing geometric differences with a size of 10 pixels. We interpret the background prediction C_{BG} as unreachable plateaus (watersheds) and combine it with the smoothed image (Figure 2.15d). Combining the resulting image with the predicted fiber centers as seeds (Figure 2.15c), we apply the watershed transformation.

(a) Unmodified raw image

(b) Combined Artifacts (white)

(c) Watershed (artifact fibers are white)

(d) Binary eroded watershed

Figure 2.16: *ROI* of S5-R-PV-3_4. Combined artifact image (Figure 2.16b), watershed which identifies these fibers as to be ignored (Figure 2.16c) and final eroded binary fiber display (Figure 2.16d)

The generated catchment basins are compared with the predicted artifact regions (overlap, cryo, unknown), which eliminate fibers that have an intersection with any of them (Figure 2.16).

Smoothing Out goal is to smooth the input image but to preserve the edges to a certain extent. Therefore, we applied a bilateral filter (see Tomasi and Manduchi [74]), which is defined as

$$I^{filtered}(\mathbf{x}) = \frac{1}{W_p} \sum_{\mathbf{x}_i \in \Omega} I(\mathbf{x}_i) f_r(\|I(\mathbf{x}_i) - I(\mathbf{x})\|) g_s(\|\mathbf{x}_i - \mathbf{x}\|) \tag{2.8}$$

where the normalization term

$$W_p = \sum_{\mathbf{x}_i \in \Omega} f_r(\|I(\mathbf{x}_i) - I(\mathbf{x})\|) g_s(\|\mathbf{x}_i - \mathbf{x}\|) \tag{2.9}$$

ensures that the filter preserves image energy and

$I^{filtered}$ ... is the filtered image
I ... is the original input image to be filtered
$\mathbf{x}$ ... are the coordinates of the current pixel to be filtered
Ω ... is the window centered in x
f_r ... is the range kernel for smoothing differences in intensities
g_s ... is the spatial kernel for smoothing differences in coordinates.

The nonlinear combination of close image values is used to smooth the image while preserving edges (Equation 2.8). Based on combination of the geometric closeness and photometric similarity, the resulting image inclines towards near values over distance values in domain and range. A further benefit, in contrast to standard filtering, is that it produces no phantom colors along edges in color images and reduces phantom colors where they appear in the original image [74].

The diameter of each pixel neighborhood which is used during filtering is proportional to g_s. We used a range kernel size of 150 and a spatial kernel size of 10 pixel (Figure 2.15b).

Background The predicted background allows us to define areas within an image, which definitely are not fiber substance. With this knowledge we mark these areas as invalid for watershed spreading (boundaries which have infinite height). By thresholding the background prediction from our *CNN* we create a binary image, where we apply noise removal with the morphological transformation *open* (erosion followed by a dilation). The final background area is removed from the smooth input image (Figure 2.15d).

Seeds A general approach for the watershed algorithm is the distance transformation of an image to identify all objects within an image. This does not perform well since the accuracy of discrimination is poor (Figure 2.3d). By using our *CNN* predicted seeds, we are able to define initial marks for the flooding technique (Figure 2.15c). This enables identification of each individual fiber. We threshold and apply noise removal to the seed predictions and extract their location information as input parameters for the watershed algorithm.

Watershed The principle of the watershed algorithm can be understood as the flooding of a topographic surface. An image, converted to gray-scale can be seen as such a surface, when we interpret high intensities as peaks and low intensities as valleys. The flooding starts from its *minima* and we prevent merging of water basins when they come from a different source. This partitioning of the image is done until all peaks are under water. To avoid some of the over-segmentation it has proven useful to smooth the image due to noise or local irregularities in the (gradient) image. Another major enhancement is to define the flooding points by a previously defined set of markers, which limits any over-segmentation (Figure 2.15c).

Artifacts We combine all artifact predictions to a mask (Figure 2.16b), which we use after we apply the watershed transformation. Any fiber intersecting with an artifact gets marked as invalid (white fibers in Figure 2.16c). With that process we are also able to create a binary mask which simplifies annotation classes to two class problem (fiber no fiber, Figure 2.16d).

2.3 Direct Fiber Segmentation

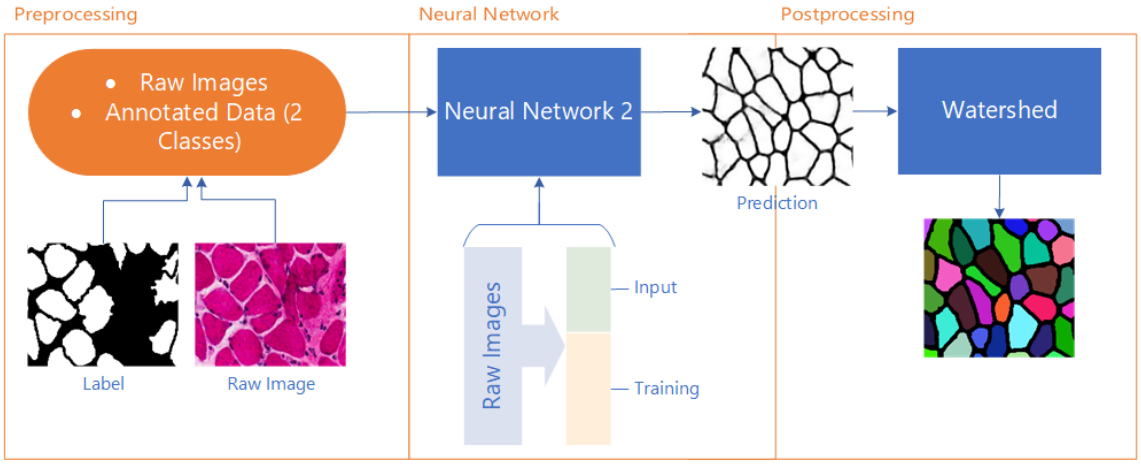


Figure 2.17: The pipeline consists of three steps: First, take the annotation created by segmentation bootstrapping. Next, train the Convolutional Neural Network with the annotated images and predict on images not used for training.. Finally, apply watershed transformation on CNN's prediction-output.

The previous section, which bootstraps our annotation baseline, requires a considerable number of steps to create a segmentation. If we identify any major errors in our annotation we have to rerun the label combination- and training process to create the predictions and apply the post-processing procedure, which is a time consuming and unsatisfactory process.

We build upon the hypothesis that the majority of the detected fibers are correct. To reduce the number of required steps for updates in the annotation sets, we use the previously generated (binarized) output as labels (fiber and no fiber) to train another *CNN*.We want to directly predict a segmentation of individual fibers. With the simplified network input approach we generate a binary fiber-no-fiber pixelwise prediction, on which we want to apply the watershed transformation and extract morphological information. Improvement of segmentation results will only require editing of a binary image, which greatly reduces and simplifies the required number of steps:

1. Preprocessing: The annotated labels are provided fully automatically by the annotation bootstrapping.

2. Convolutional Neural Network: Learn from annotation to create pixelwise segmentation prediction for unknown histological slices

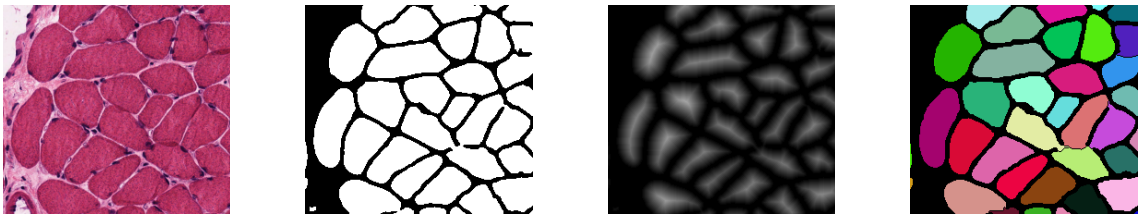
3. Postprocessing: Use only *CNN* prediction to apply a watershed segmentation and export morphometric information

The postprocessing is independent of the raw input image, since the watershed is applied on the *CNN* segmentation prediction. All fiber centers are calculated by distance-transforming the input image and thresholding the result.

Preprocessing A preprocessing is not necessary. We can use the binary output from 2.2 or any binary annotation. In case of obvious errors we can use an arbitrary image editing software to change the annotation of an image directly in the label image. Visible errors like serration are ignored, since they are random and will be smoothed out by the neural network's prediction.

Neural Network For the training and processing of image data, in comparison to 2.2.2 we reduced the number of classes to two, instead of six, other parameters stayed alike. Variations using image deformation show better results with our introduced label data augmentation activated. Any new input image automatically creates a segmentation. If we want to identify each fiber and extract morphometric information the segmentation prediction requires postprocessing.

Postprocessing In contrast to the previous section, we use fiber and no fiber predictions of the *CNN* to apply the watershed transformation (Figure 2.18b, Figure B.3). Artifacts were automatically defined as 'non-fiber' by the labeling procedure in 2.2.3, which we benefit from now. The created prediction of the *CNN* is a pixelwise segmentation, which can be conveniently processed to extract morphological information.



(a) Raw image **(b)** Thresholded segmentation **(c)** Distance Transformation **(d)** Applied Watershed

Figure 2.18: S4-R-PV+2: 2.18a-2.19c show *ROI*, starting with raw area, thresholded(t=150) prediction, distance transformation, resulting watershed transformation.

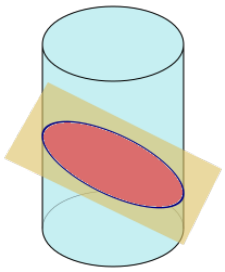
Before we threshold the prediction, we apply a circular Gaussian function ($\sigma = \sigma_x = \sigma_y$, kernel size is 5, Equation 2.10) to smooth the prediction and reduce otherwise resulting holes, very small fibers or artifacts.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x-\mu_x}{2\sigma_x^2} - \frac{y-\mu_y}{2\sigma_y^2}} \tag{2.10}$$

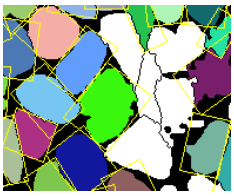
where we assume a circular Gaussian function with

σ . . . the standard deviation
μ . . . the expected value.

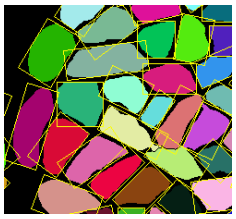
As next step, we threshold the smoothed image to create a binary image. This is the input for the distance transformation [26] (Figure 2.18c), where we, again, apply a threshold to obtain a seed map. To reduce unwanted holes within that mask, we apply a dilate operation (kernel size is 3). Finally, we use our markers as seeds and apply the watershed transformation on the thresholded segmentation prediction (Figure 2.18b).



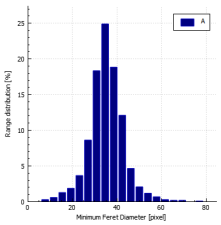
(a) Simple muscle fiber model (cylinder: muscle model, yellow: intersection plane, red: muscle fiber cut)



(b) Bounding boxes around fibers in the bootstrapping process (S5-R-PV-3_4)



(c) Minimum bounding boxes around fibers in the direct segmentation approach (S4-R-PV+2)



(d) Minimum Feret Diameter-Distribution (direct segmentation, S4-R-PV+2)

Figure 2.19: (a) Simple 3D muscle model to illustrate the minimum Feret diameter F_{min}. (b) Minimum bounding rectangles used for automated extraction of F_{min} for S4-L-PV-2_2 from a 2D segmentation result.

Minimum Feret Diameter The Feret Diameter (F) is a measure of the maximum dimension of an object in a given direction. We are interested in the maximum extension in the perpendicular direction to the muscle fiber, the Minimum Feret Diameter (F_{min}). As a simplified muscle model a cylinder can be used, and the histological slices which we work with, as slices of the cylinder. We apply a minimum bounding box around the fibers within the slice, and the minimum diameter (horizontally vs vertically) is the F_{min}. This is done because it is unlikely that the histological slice is absolutely perpendicular to the muscle, and therefore the plane results in a cylinder shaped fiber.
We measure F_{min} by fitting a minimum bounding rectangle around identified fibers (Figure 2.19b). The shorter length (width or height) is F_{min}. We create a histogram of F_{min} distribution where we can evaluate shifts in average fiber size while comparing different sets of *HE*-stained slices (Annotation GUI (2.4.1)).
The output of this bootstrapping approach (Figure 2.16d) is the binary segmentation

which is being used for continuous improvement. Since we trained a *CNN* and our processing pipeline is an automated setup, generation of new annotations of raw images is straightforward. Even though the resulting fiber segmentations show serrated borders, they are already able to distinguish individual fibers. Furthermore, regions are identified which contain artifacts where an unambiguous assignment is not possible (see 2.16). With these predictions processed to a binary mask, we target to train another *CNN*, which we expect to have few errors where either the impact is low or the falsely classified regions can be corrected with little effort by a domain expert. By identifying the minimum bounding rectangle (Figure 2.19c) we extract the morphological information from any fiber (Figure 2.19d). Since only fiber-no-fiber prediction is required to apply the postprocessing, we greatly reduce the dependencies which further eases evaluation and editing of segmentation.

2.4 Domain Expert Interaction

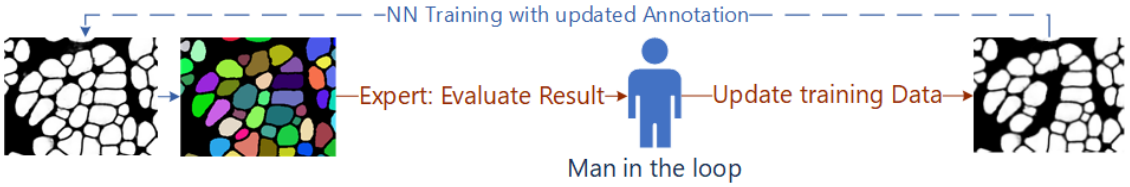


Figure 2.20: The pipeline builds upon the Direct Fiber Segmentation (2.3). The man in the loop is an histological expert, who corrects prediction errors.

We want to iteratively improve the fiber segmentation result. Consequently we require expertise review and input of a domain expert. In contrast to labeling an entire digitized HE-specimen, our medical partner examines the results from Postprocessing (2.3) and modifies them wherever necessary. Therefore we provide a dedicated Graphical User Interface (GUI) to streamline the annotation update work flow. Depending on the regions to be updated this may still be a time consuming task, ranging from minutes for minor updates (i.e. a region of an image containing few fibers) up to several hours or even days if an image requires complete rework (i.e. manual labeling of thousands of fibers).

Depending on the changes applied by the man in the loop, we can either update our calculated network weights or restart training from scratch. The following three steps are performed involving the expert:

1. Provide segmentation results from our binary pixelwise classification *CNN*
2. Review of segmentation and update if required
3. Provide updated annotation $\rightarrow$ 1.

Following this approach, we obtain new training data such that a retrained *CNN* is able to predict a segmentation result for any new histological section, similar to as if the domain expert would segment the image manually.

2.4.1 Annotation GUI

Since a correction to the binary segmentation result only requires differentiation between fiber- and non-fiber, we ask our medical collaboration partner to examine the results. We want to cover the use case of updating a segmentation with as little as possible distraction of the task, therefore we identified necessary (editing) and useful (zooming, scrolling, statistic extraction) features which we implemented in our dedicated fiber annotation *GUI*.

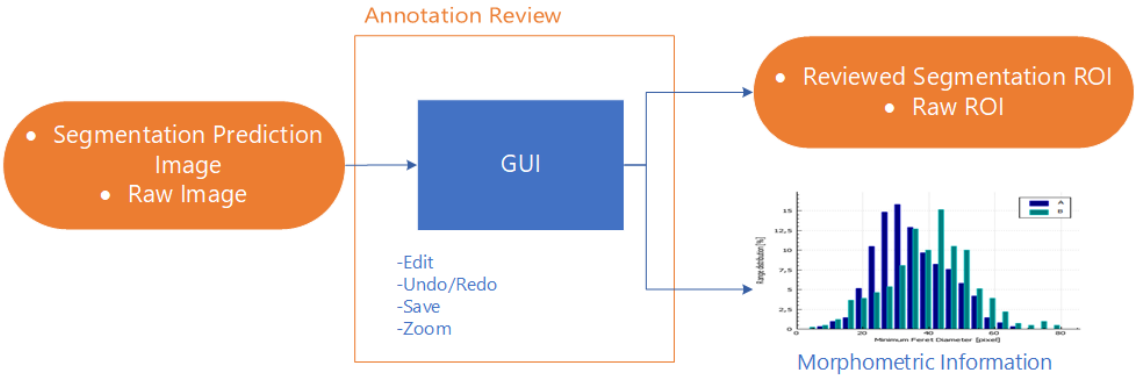


Figure 2.21: The dedicated graphical user interface (GUI) expects a pairwise input (raw image and segmentation prediction). The user of the GUI can edit the annotation and/or view the statistical data

With the focus of updating annotations in mind, we provide overlay functionality of raw image and prediction image (0-100%), preview of created segmentation, automated pairwise (raw, label) storing and morphological information extraction and visualization. The current version supports saving and continuing from previous annotation progress. All statistic data can additionally be exported in a Tab Separated Value (TSV)-file to support import to other applications.

Input As input we require a gray-scaled segmentation prediction which is generated by the *NN* output and the raw images corresponding to them. We use the postprocessing step mentioned in Postprocessing (2.3) as a preprocessing step, with additional modularity like an individual variation in thresholding, blur-kernel size or hole-fill kernel size which may improve resulting segmentation. It is possible to provide inputs from other sources as long as a pairing is provided.

Modifications We allow white and black annotation, with different diameters, where white represents a valid fiber and black everything else. Additionally, the *GUI* offers undo/redo functionality.

Output We store edited images pairwise to ease training (raw and label). Only edited regions are saved, with a minimum size of $572x572$ pixels, which is the required minimum dimension of the *CNN*. Furthermore, we can assign images to different sets and compare their F_{min} distribution to each other. This helps to validate the expected result like mentioned in [42] (see 3.2.3.2 Results).

2.5 Discussion

With the provided *HE*-stained slides and the missing of annotation information we introduced a process to create a segmentation and morphological information extraction and offer a basic active learning process to effectively include domain expert knowledge.

In Segmentation Bootstrapping (2.2) we show our approach to create suitable labels for distinct identification of fibers. The process requires a considerable number of preprocessing steps where the semi-automated annotation is the most time consuming. Even though some of them are time-consuming, they can be done by a layman and only require a fraction of the time of manual fiber segmentation. With these preparations we were able to identify fiber instances on untrained slides, and can create a binary segmentation for further training. Additionally we can measure the minimum Feret diameters and extract the data for statistical evaluation.

The Direct Fiber Segmentation (2.3) builds upon the generated annotation data from Postprocessing (2.2.3). We target to reduce the required steps in the process of editing annotations by reducing the annotation classes down to fiber and non-fiber (training and processing), thus our medical partner can use this setup to update training data directly.

The applied modifications to the *CNN* (Convolutional Neural Network (2.2.2)) enable training on even a small set of images, with focus on *HE*-stained histological images. This is especially useful in case of iterative updates, where the number of images edited may be rather limited in contrast to the number of training images used prior.

With the reduced complexity in the pipeline of Direct Fiber Segmentation (2.3) in contrast to Segmentation Bootstrapping (2.2) we introduced an effective man-in-the-loop principle in Domain Expert Interaction (2.4), where we present our segmentation result to domain experts who improve the annotation where necessary.

3 Experiments and Results

3.1 Overview

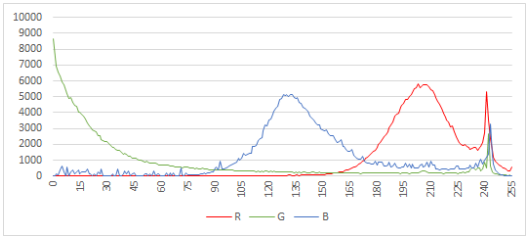
We applied the following experiments to verify our approach. We start from feasibility tests through quantitative and qualitative validation, depending on available data. They display, in order of our research progress, in combination with the respectively provided image data from our partners, how we achieved muscle fiber segmentation, including a simple active learning loop, while starting from no annotation data at all. For our experiments several extensions to the existing neural toolbox [76] were necessary. The medical image data we work with, was made available to us in different batches (datasets, appendix B).

3.1.1 Datasets

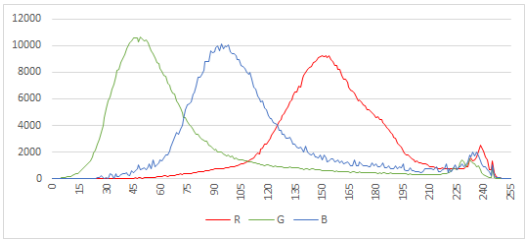
Our medical partner provided the images, from the vocal cords of sheep, on three different occasions (I,II,B). We provided the result of our first iteration to our medical partner and asked them to improve the segmentation on 15 images (randomly selected from A). Finally, to enable independent quantitative evaluation of our bootstrapping approach, our medical partner provided entirely manually generated segmentation labels for 10 images, arbitrarily selected from Dataset B.

Batch/Dataset	Image Count	Avg. Size (min, max) W x H [pixel]			Avg. Intensities (min, max) R G B $\in [0, 255]$		
I	31	1036.00 x 860.00	(1036 x 860,	1036 x 860)	207.32 45.71 148.19	(186 14 108,	240 106 188)
II	66	1296.58 x 1107.08	(588 x 625,	1837 x 1635)	157.83 72.68 113.61	(98 42 87,	201 113 155)
A	97	1215.15 x 1029.87	(588 x 625,	1837 x 1635)	173.65 64.06 124.66	(98 14 87,	240 113 188)
B	270	2360.77 x 1778.09	(656 x 852,	3781 x 2762)	164.13 81.72 123.31	(113 40 72,	202 157 186)

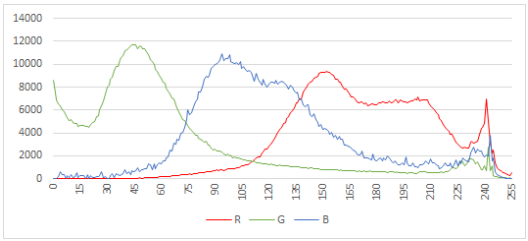
Table 3.1: The batches and datasets as they were provided and are used in the experiments. Dataset *A* is the combination of the batches *I* and *II* as it is eases understanding, when discussing the experiments.



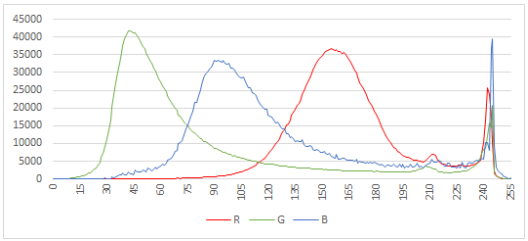
(a) Histogram of batch I



(b) Histogram of batch II



(c) Histogram of dataset A



(d) Histogram of dataset B

Figure 3.1: Color histograms of provided datasets. Y-axis is the pixel count and therefore indirectly reflects the amount of images provided. Especially the green-channel of the RGB representation indicates differences of the provided image sets

Initially, there was no annotation data provided to us from our medical partner. We received raw images, where the sets varied in size (number of images), staining intensities, resolutions (see Table 3.1, Figure 3.1) and occurring artifacts (see Table 2.1). Depending on the experiments, if required, the annotations were created by a layman.

Batch I Batch *I*, which consists of 31 images, all with the same size (see Table 3.1), was the first set that was provided by our medical partner (a pixel equals $0.98\mu m$). There was no annotation available (*raw data/image*). The images are homogeneous and have very low variation in color intensity (see Figure C.1) which is likely due to the fact that they were all prepared in the same staining session. They do contain artifacts (i.e. freezing, see Figure 1.5).

Batch II In a second batch, 66 images were provided (batch *II*, see Figure C.2). In contrast to *I* they have a large variation in size and color intensities (see Figure 2.10). Additionally the images were provided in a different resolution, which required a scaling operation to resize the images to align with batch *I* (batch I has $\frac{1}{4}$ the resolution in comparison to *II*).

Dataset A For convenience, as it is used in the experiments later, we introduce dataset *A* which is the sum of $I + II = 97$ images. In the progress of our experiments we selected

41 of these images and created our dot annotation.

Dataset B This is an entirely independent dataset (270 images). It was never used for training of any Neural Network (NN). Dataset *B* is exclusively used to evaluate our result with the current widespread method which relies on triple immunofluorescence staining and a manual annotation applied by a domain expert.

3.1.2 Evaluation

Our results and experiments tackle the initial feasibility test of identifying fiber instances, testing for training parameters, distribution of minimum fiber diameter, evaluation of segmentation performance and point out parameter influence for the annotation Graphical User Interface (GUI), quantitatively or qualitatively (depending on the available data). Our metrics align with the confusion matrix as mentioned by Fawcett [24]. We present our results in the following order:

- Dot Annotation Detection: Feasibility of Distinct Fiber Identification
- Segmentation Experiments
  - Qualitative evaluation of iteration steps for training our Convolutional Neural Networks (CNN) to bootstrap label data
  - Quantitative evaluation of morphometric extraction from HE-images (bootstrap) with reference data
  - Three-fold cross validation of direct fiber segmentation
  - Comparison to expert annotation
- Annotation *GUI* parameter influence

Focus in this thesis is on the evaluation of the proposed image processing pipeline, specific optimizations i.e. on the *CNN* parameters [86], or changing the depth of the *CNN* [68, 71] were omitted.

3.1.3 Implementation

As we are working in a specific field of image analysis, with a restricted set of images in contrast to general object detection/segmentation, the available image processing modules required extensions or modifications. Especially in regards to color intensity and training data availability we extended the existing functionality. Furthermore, to evaluate our segmentation results and ease annotation updates we provided a *GUI* to our medical partner. Both extensions can run independently, but the *GUI* requires raw input image and a corresponding ground-truth (prediction) to start with, which can be provided by the binary pixelwise prediction from the *CNN* used in the direct fiber segmentation.

We will give a short overview of the setup requirements, for details refer to appendix B (Implementation Detail).

- Neural Toolbox Modifications: Enhancements for robustness

  We use the existing neural-toolbox provided by [76] built upon Caffe [39] which offers a huge degree of parallelism by training and evaluation on Graphics Processing Unit (GPU) as well as extended imaging support. Our extensions[1] to the existing code[2] were written in C++.

- Annotation *GUI*: User friendly improvement of annotations

  The *GUI* is written in C++ and relies on OpenCV[3], Qt 5[4] and supports Linux and Windows.

3.2 Experiments

We started with a feasibility test where we evaluate distinct fiber (center) identification. We use a subset of 41 images from dataset A and extended it with fiber center annotation (one class, Dot Annotation Detection (3.2.1)).

Continuing, in conjunction with the previously tested center-annotation, we semi-automatically annotated additional classes and create a mapping from our six annotation types to a binary segmentation image and evaluated the necessary iteration steps in training the *CNN* (Segmentation: Qualitative Evaluation for Neural Network Parameters (3.2.2)).
After our processing pipeline produced the first segmentation results, our medical partner provided us with a large image set (dataset B) which is from sheep where reference morphometric information through the established method of triple immunofluorescence labeling is available. We applied our segmentation and morphometric extraction approach to compare results (Segmentation: Quantitative Evaluation of Morphometric Information Extraction (3.2.3)).

With our binary output from Segmentation: Qualitative Evaluation for Neural Network Parameters (3.2.2) we re-use dataset A to train a segmentation baseline (Segmentation: Cross Validation (3.2.4) and Segmentation: Comparison to Expert Annotation (3.2.5)).

Finally, we show the influence of parameters in the annotation *GUI* (Annotation GUI parameter influence (3.2.6)) which we provided to our medical partner, which he used to create the segmentation for Segmentation: Cross Validation (3.2.4).

[1] https://bitbucket.org/derKlaus/, 25.10.2017
[2] https://github.com/naibaf7/caffe_neural_tool, 25.10.2017
[3] https://opencv.org/, 25.10.2017
[4] http://doc.qt.io/, 25.10.2017

3.2.1 Dot Annotation Detection

The goal of this experiment is to verify that it is viable to distinctly identify fibers. Our quantitative evaluation is performed using the non-expert dot annotations of muscle fiber centers on 41 images (from dataset A). While this evaluation is based on a ground truth annotation that may contain errors, it is still valid to evaluate pure detection performance of an automatic U-Net center prediction algorithm, without taking the physiological interpretation into account.

3.2.1.1 Metrics

Detection evaluation was done by calculating the centroid of each fiber for each image in the ground-truth and measuring their distance in the hypothesis (source for candidates): For each centroid in the ground-truth the k-nearest neighbors were identified (in the hypothesis, within the defined threshold T of 20 pixel). Centroids (candidates) with the least distance to their corresponding centroid (ground-truth $\leftrightarrow$ hypothesis) are counted as true positive. If a candidate has no match, it is counted as false positive. If the distance of all k-neighbors to a ground-truth centroid is above T, it is labeled as false negative (undetected ground truth centroid). Candidates that do not match the current evaluated centroid, may be assigned to another centroid (as candidate) if the threshold allows it. For fiber center (feasibility-) detection we used the following specific evaluation of the k-nearest neighbors (defined by a distance threshold T)

$$d = \sqrt{(p_x - p'_x)^2 + (p_y - p'_y)^2)} \tag{3.1}$$

where:

d ... Euclidean distance
p ... fiber center in the manual annotation
p' ... fiber center in the prediction

- T = 20 pixel: Distance threshold to identify TP/FP/FN
- True positive (TP): Euclidean distance $d_{min} <$ T
- False positive (FP): Euclidean distance $\forall d(d < T \wedge d \neq d_{min})$
- False negative (FN): Rest
- Recall (sensitivity):
$$REC = \frac{TP}{TP + FN} \tag{3.2}$$
- Precision:
$$PRC = \frac{TP}{TP + FP} \tag{3.3}$$

- F1 score (harmonic mean of precision and sensitivity):

$$F1 = \frac{2 * PRC * REC}{PRC + REC} \tag{3.4}$$

3.2.1.2 Experiment

All fiber-center annotation images ($n = 20$) mentioned in Table 3.2 are used for testing, the remaining for training ($n = 21$). The annotated images are referred to as ground-truth and the (thresholded) net prediction as hypotheses. The fiber center annotations have a diameter of 17 pixels.

							Mean Euclidean Distance
Image	TP	FN	FP	REC	PREC	F1	TP [Pixel]
S1-L-PCA-3	582	49	53	0,9223	0,9165	0,9194	5,1484
S1-L-PCA-3_3	591	28	23	0,9548	0,9625	0,9586	5,1042
S2-L-PCA-3	371	18	32	0,9537	0,9206	0,9369	5,1082
S2-L-PCA-3_2	444	10	14	0,9780	0,9694	0,9737	4,5774
S2-L-PCA-3_3	367	16	24	0,9582	0,9386	0,9483	5,3287
S2-R-PCA-3	472	20	19	0,9593	0,9613	0,9603	4,9577
S3-R-PCA-3	371	4	46	0,9893	0,8897	0,9369	5,0919
S4-L-PV-3_2	278	23	37	0,9236	0,8825	0,9026	5,8900
S4-L-PV-3_3	328	12	12	0,9647	0,9647	0,9647	5,1791
S4-R-PCA-3	377	7	19	0,9818	0,9520	0,9667	5,3584
S4-R-PV-3	322	15	3	0,9555	0,9908	0,9728	4,9540
S4-R-PV-3_2	539	6	3	0,9890	0,9945	0,9917	4,2218
S4-R-PV-3_3	290	10	70	0,9667	0,8056	0,8788	4,8275
S4-R-PV-3_4	523	12	1	0,9776	0,9981	0,9877	3,7584
S5-L-PCA-3	406	33	73	0,9248	0,8476	0,8845	5,9823
S5-L-PV-3	308	8	36	0,9747	0,8953	0,9333	5,5069
S5-L-PV-3_4	418	16	5	0,9631	0,9882	0,9755	4,2946
S5-R-PCA-3_3	318	5	32	0,9845	0,9086	0,9450	5,9462
S5-R-PV-3_3	522	30	15	0,9457	0,9721	0,9587	5,4605
S5-R-PV-3_4	414	17	7	0,9606	0,9834	0,9718	3,7666
Average	412,05	16,95	26,2	0,9614	0,9371	0,9484	5,0231

Table 3.2: Fiber Center Identification: Feasibility validation of automated detection. 41 Images were split into training and test set. Threshold for fiber-center-matching was set to 20 pixel.

This evaluation shows an average recall of $96,14\%$ and consequently an F1-score of $94,84\%$. The average Euclidean distance of fiber centers among the test set was 5.02 pixels (see Table 3.2). These results indicate that fiber identification is a feasible task.

3.2.1.3 Conclusion

The manual annotation of the fiber centers was a rather trivial but repetitious task. Since it only required approximately center annotation the total time of annotation sums up to 20 hours for 41 images[5]. Minor deviations in the annotation process barely have an impact, since the count of annotated fiber centers is high enough to balance out outliers. Furthermore, we targeted to identify individual fibers and are not interested to locate their centers. The chosen dot annotation size has an impact for the neural network evaluation, since the diameter size directly influences the prediction outcome. It is a trade off between losing center points (lower diameters) and merging center points (higher diameters) in the prediction image. This may be optimized by changing the U-Net structure. The resulting scores for recall and F1-score prove that the detection performance of the automatic U-Net center prediction algorithm is a valid starting point for distinct fiber localization.

3.2.2 Segmentation: Qualitative Evaluation for Neural Network Parameters

Building on the localization experiment, we target to create a segmentation baseline. We extended the annotation of the 41 images of dataset A, as mentioned in Segmentation Bootstrapping (Table 2.1). We desire to identify suitable parameters for the *CNN* to avoid over- and under-fitting. As no reference data is available, we employ a qualitative evaluation.

3.2.2.1 Metrics

For the qualitative evaluation, we review the resulting segmentation in a specific Region Of Interest (ROI). These *ROI* are snapshots taken after different iteration steps. They include artifacts (freezing, unknown, overlap) as our intermediate goal is to identify those.

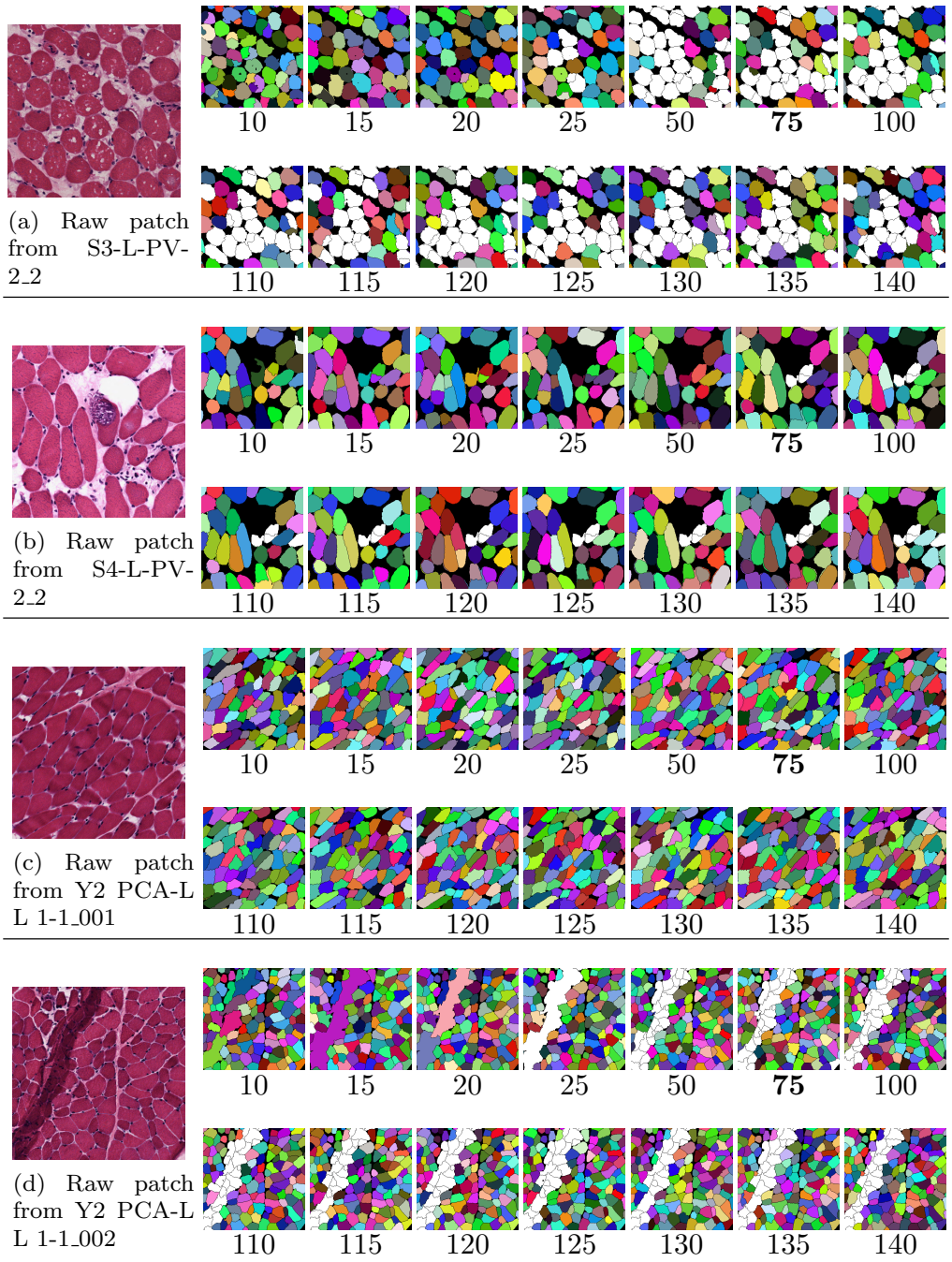
3.2.2.2 Experiment

Since no reference data was available for the first iteration, to compare our intermediate results to, we did a qualitative review of the resulting segmentation. The image data we evaluated were images not used in the training set.

To ease comparison and identify fiber instances which our pipeline characterized as faulty (white), we used the colored version of the pipelines proposed segmentation results (Figure 2.7). We applied fixed post-processing parameters and compared different *CNN* training iterations (Figure 3.2).

The qualitative evaluation was done by taking snapshots of sample regions, e.g. regions that contained artifacts (Figure 3.2). We empirically see that training the U-Net for 75000 iterations is enough to detect artifacts, which contributes to marking fibers as invalid that influence the morphometry.

[5]On average about 16 fiber centers per minute per person.



(a) Raw patch from S3-L-PV-2_2

(b) Raw patch from S4-L-PV-2_2

(c) Raw patch from Y2 PCA-L L 1-1_001

(d) Raw patch from Y2 PCA-L L 1-1_002

Figure 3.2: For a set of images, a region was selected and qualitatively evaluated over several iterations (Note: steps in thousands). White fibers are ignored for binary segmentation creation, i.e. artifacts are filtered.

3.2.2.3 Conclusion

With combination of the different prediction classes that define valid and invalid regions we are able to create a segmentation by using the watershed algorithm. The distinctively identified fibers are segmented (see the differently colored fibers in Figure 3.2) but have serrated borders due to the applied algorithm. A benefit of the resulting segmentation is that artifacts are considered so that fibers which overlap artifact regions are marked invalid (white fibers in Figure 3.2). We identified that annotation classes with low representation (Table 2.1) require 75000 iterations to be identified.

3.2.3 Segmentation: Quantitative Evaluation of Morphometric Information Extraction

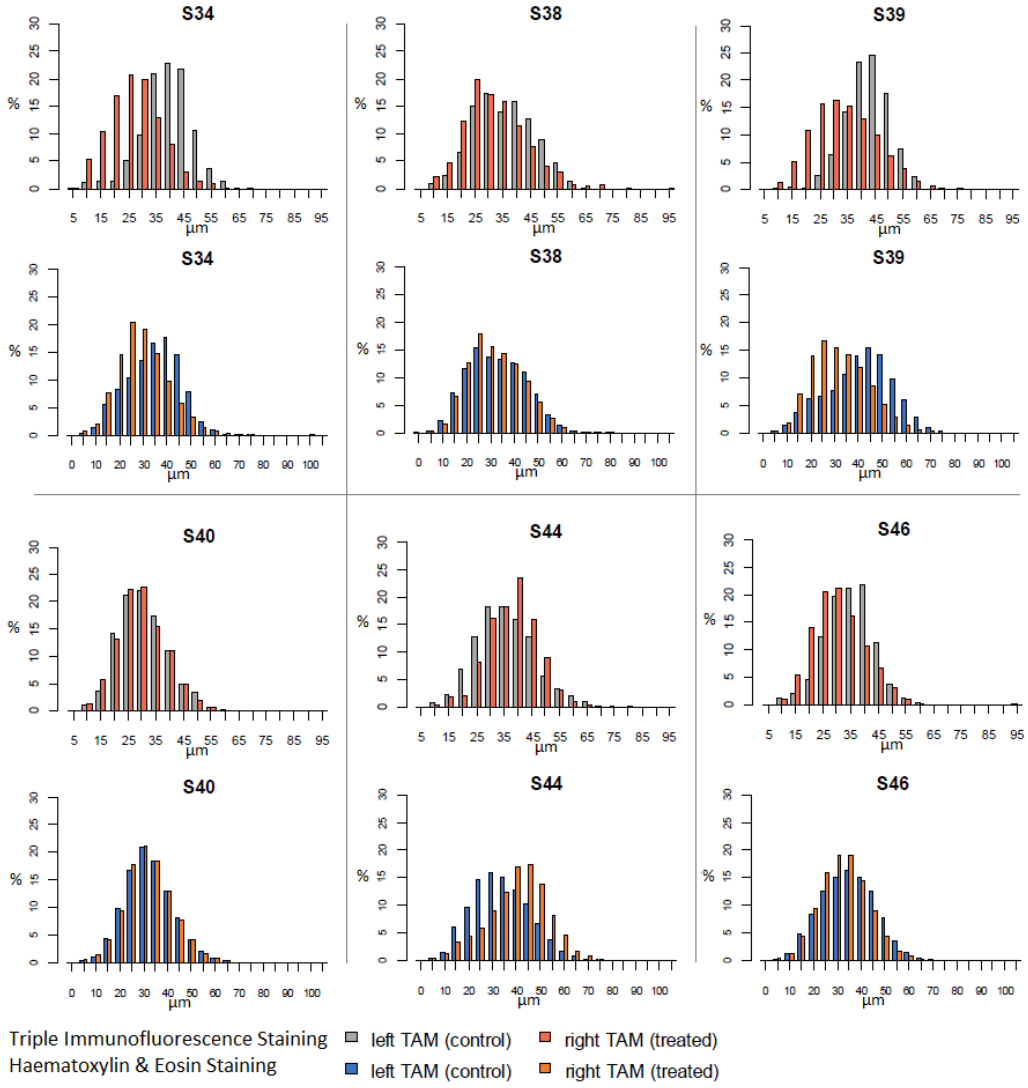
To quantitatively evaluate our first segmentation result, our medical partner provided us with a new set of 270 images (dataset B, on 12 sheep in total). They were grouped into fibers where Functional Electrical Stimulation (FES) was applied and control groups, where the affiliation of them is unknown to us. We expect a matching from the distribution created from our morphometric information extraction to the reference data distribution. We ran this experiment with the trained net from Direct Fiber Segmentation (2.3). After postprocessing (extraction of morphometric information), we provided the resulting data (Minimum Feret Diameter (F_{min}) of each fiber within each section for each sheep) to our medical partner. They prepared a quantitative comparison of our data to, from triple immunofluorescence stained extracted F_{min}, since we did not know which muscle of a sheep was part of the control or treatment group, and therefore what had to be expected. We compared our statistical distribution (even rows in Figure 3.2) with the provided triple immunofluorescence provided by our medical partner (odd rows in Figure 3.2) and expected aligned shifts, to changed muscle fiber size, when *FES* was applied in the treatment group.

3.2.3.1 Metrics

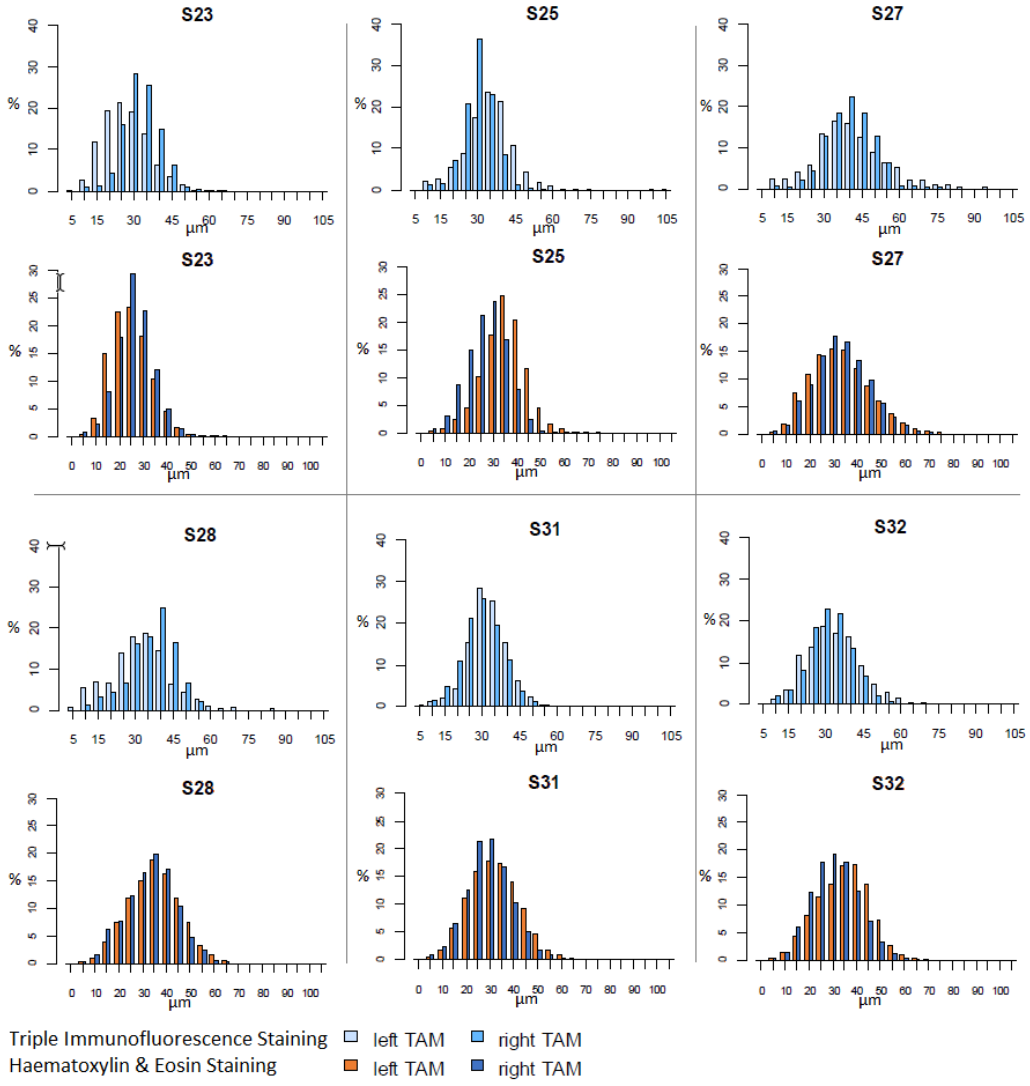
We measure F_{min} for each identified fiber and cluster them by size (see Minimum Feret Diameter (2.3)). We qualitatively compare the size shift from the standard method with our approach.

3.2.3.2 Results

We grouped the resulting measured fibers in the same manner as with the provided statistical distribution. The 270 histological sections were processed which included the morphometric information was extracted for each slide and then grouped by the corresponding sheep identifier. If a stimulation was applied, we expect that the trained muscle has, in average, larger F_{min}. This is visible in the graphics by a shift of the differently colored pins. Lack of such a shift indicates that the training was not done or insufficient.



(a) Treatment Group, HE vs triple immunofluorescence staining. First and third row from HE-staining, second and fourth from triple immunoflurescence staining. The right muscle is treated and compared to the left. A shift between the different colored pins within a figure indicates a training effect.



(b) Control Group, HE vs triple immunofluorescence staining. First and third row from HE-staining, second and fourth from triple immunoflurescence staining. No treatment differences between left and right muscle.

Figure 3.2: Dataset B, Thyroarytenoid Muscle (TAM) sections, comparison from reference results of triple immunofluorescence staining with our process relying on Haematoxylin Eosin (HE) staining. A total of 12 sheep were used for evaluation, six sheep from the treatment group and six sheep from the sham-group where the title S identifies a sheep (with numerical ID). The different pins identify fiber counts grouped by the F_{min} (5 μm bins, y-axis in %). Odd rows show triple immunofluorescence staining, the row below each shows the results from *HE* staining of the same sheep.

In the treatment group (see Figure 3.3a where *FES* was applied), we see an intensity shift that corresponds to muscle stimulation in sheep 34, 39, 44 correlating in both staining methods. Sheep 34 and 39 show a training effect that result in larger fiber sizes, whereas sheep 44 displays a reverse trend. Sheep 38 and 40 indicate that no training effect was

achieved which may be due to physique of the selected sheep (i.e. a young sheep that has a well-trained muscle) or location of the electrodes. For sheep 46 no conclusive statement can be said. In the sham-group (Sheep 23, 25, 27, 28, 31 and 32, Figure 3.2b) we see a correlation between the two morphometry extraction methods in most images. Individual outliers like in S25 may be due to the selected region for histological section.

3.2.3.3 Conclusion

Triple immunofluorescence and *HE* stained slices cannot be compared directly, since once a section is stained it cannot be undone. Furthermore, the different staining methods would result in varying F_{min} (if theoretically practiced on the same muscle fiber section) per fiber as there is a different shrinking in diameter due to the diverse applied staining method. We compare the different statistical results, since enough specimen were taken from the particular muscle and prepared for both staining methods. The statistical distribution of F_{min} from triple immunofluorescence staining reveal a qualitative correlation to our HE staining method (see Figure 3.2).

3.2.4 Segmentation: Cross Validation

We want to know the influence of our introduced modifications of the image preprocessor of the *NN* (see Enhancements to the neural toolbox). Initially, we did not have any ground truth for our data to compare to. We provided a review set of segmentation predictions (15 prediction images from dataset A, with the corresponding histological section) from Segmentation Bootstrapping to our medical partner for manual review in conjunction with our annotation *GUI* (see appendix B.2).

After correction by our medical partner, we evaluated the U-Net segmentation model in a three-fold cross validation setup, splitting the accurately annotated images into 10 training and 5 test images respectively.

3.2.4.1 Metrics

The binary neural network output and the reference data are compared by evaluating the hypothesis (prediction) with the ground truth (segmentation). We extend the metrics to include specificity and apply them in a segmentation context:

- True positive (TP): Hypothesis class-foreground (pixel) is the same as ground truth
- False positive (FP): Hypothesis class-foreground does not match ground truth
- False negative (FN): Hypothesis class-background does not match ground truth
- True negative (TN): Hypothesis class-background matches the ground truth.

Iterations	D	I	F1	PREC	REC	SPEC
50000	0,5	30	**0,88316**	0,89037	0,88102	0,87055
45000	0,5	60	0,88315	0,89036	0,88098	0,87070
45000	1	60	0,88266	0,87791	0,89008	0,84960
25000	0	30	0,88254	0,89316	0,87670	0,87450
50000	0,5	60	0,88240	0,89164	0,87880	0,87349
30000	0,5	30	0,88223	0,88542	0,88335	0,86283
40000	0,5	30	0,88207	0,88668	0,88353	0,86554
50000	0	30	0,88193	0,90314	0,86715	0,88988
20000	0,5	30	0,88188	0,89050	0,87736	0,87075
45000	0	60	0,88169	0,89854	0,86983	0,88306

Table 3.3: Three-fold cross validation of U-Net training with 15 corrected binary segmentation images: Top ten parameters over all three sets for intensity shifts $I \in \{0\ 30\ 60\}$ and affine deformations $D \in \{0\ 0,5\ 1\}$ ordered by F1-Score.

- Recall (sensitivity):
$$REC = \frac{TP}{TP + FN}$$
- Precision:
$$PRC = \frac{TP}{TP + FP}$$
- F1 score (harmonic mean of precision and sensitivity):
$$F1 = \frac{2 * PRC * REC}{PRC + REC}$$
- Specificity (true negative rate)
$$SPEC = \frac{TN}{TN + FP} \tag{3.5}$$

3.2.4.2 Experiment

The randomly chosen images for cross-validation were split in three sets. We vary iterations, deformation D and intensity-shifts I. After every 5000 iterations we take snapshots of the calculated *NN* weights. These are used to periodically create predictions. We evaluate which variation and combination of parameters, of our introduced extensions to the neural tool, yield the best results. Best result can be seen in table 3.3, the progress of F1-score for the individual sets can be seen in Figure 3.3;

The best segmentation results show an average F1-score of $88,32\%$ (corresponding average recall and precision are $88,1\%$ and $89,04\%$). The fluctuations in our set 1 (see Figure 3.3a) are likely due to the random combination of training data out of our limited

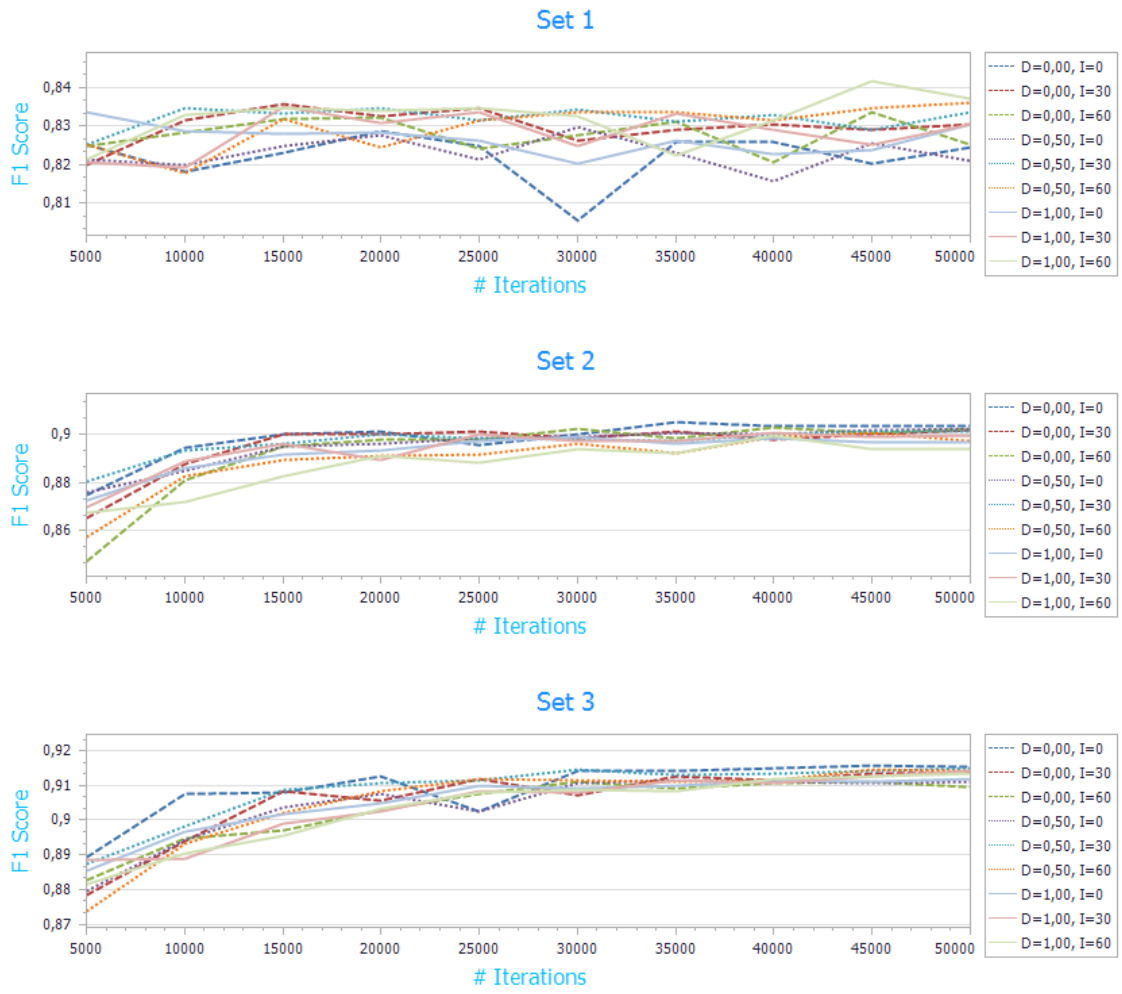


Figure 3.3: The diagram shows the progress of F1 score over several iterations (Dashed lines indicate 0 deformation, dotted lines 0.5 and solid lines have maximum).

ground truth set and therefore lack of information on similar fiber areas as in the test set.

3.2.4.3 Conclusion

As we can see in table 3.3 our modifications to the neural toolbox yield the best results (deformation and intensity shifts). The progress of the F1-score in regards to the training iterations show a fast convergence after 15000 iterations for set 2 and 3, whereas the score result for set 1 remains fluctuating. This is likely due to the split of training and test images, as specific training information that is required to converge towards a top-score, seems to be missing.

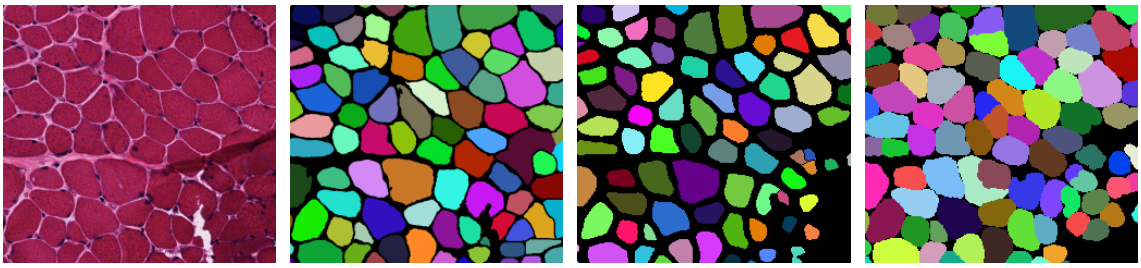
A drawback of this method is that the baseline for the annotation images was created from results which are based on our processing pipeline. Any bias our method includes

therefore is present in the prediction images when they are sent for review to the medical expert and they may propagate them in the reviewed reference segmentation images. Even though the segmentation result allows distinct fiber morphometry evaluation, it cannot be said that a segmentation from scratch would result in the same.

Nevertheless the results, in regards to the used training and test data, demonstrate that with our parameters better results were achieved than without.

3.2.5 Segmentation: Comparison to Expert Annotation

All previous experiments lead up to this validation experiment. We reuse the same evaluation procedures and extend them to present the performance of our image processing pipeline. In contrast to Segmentation: Cross Validation (3.2.4) we want to compare fiber segmentations that were done independent from our processing work flow (in this case manually from scratch), to our automatic approach. We evaluate binary segmentation, fiber distribution and individual fiber detection performance. Our medical partner prepared 10 images for comparison from dataset *B*, which were never used in our training of our segmentation pipeline. The annotation required time consuming manual labor which depended on the concentration and endurance, and thus required several hours per image.



(a) Raw ROI of *S40_TA-R1_1*

(b) Manual Segmentation

(c) Initial Automated Segmentation (U-Net)

(d) Tuned Automated Segmentation (U-Net), including dilation operation

Figure 3.4: Different segmentation results: Figure (b) illustrates the colored result after the manual fiber border notation. In (c) and (d) we see the result from the same raw image using our processing pipeline (2.3 Direct Fiber Segmentation). The gap between each fiber is clearly displayed in (c) but can be reduced by applying dilation for each detected fiber as visible in (d). The bottom right of each image displays the different artifact treatment.

Even though the basis for the segmentation were the same *HE* stained slices (Figure 3.4a), the results differ visually (Figure 3.4b - Figure 3.4d). The dilation process to increase the fiber size (Figure 3.4d) is applied consecutively for each fiber so that for possible overlapping fiber areas, the last one wins (i.e. is the visible one).

3.2.5.1 Metrics

In the first step, we compared the binary segmentation result (segmented regions are white) and the resulting F_{min} distribution. For this two-class segmentation we used the metrics from Segmentation: Cross Validation (3.2.4). The quantitative morphometric information (fiber distribution) was compared in the same manner as in Segmentation: Quantitative Evaluation of Morphometric Information Extraction (3.2.3).

To get more in depth performance feedback, we compared the individual fiber segmentation efficiency. As a precondition we detected the fiber-center position (mass center) in hypothesis and ground-truth and consequently classified each fiber as TP, FP or FN, similar to Dot Annotation Detection (3.2.1):

- T = 20 pixel: Distance threshold to identify TP/FP/FN
- True positive (TP): Euclidean distance (Eq. 3.1) $d_{min} < \mathrm{T}$
- False positive (FP): Euclidean distance $\forall d(d < T \wedge d \neq d_{min})$
- False negative (FN): Rest

With the information about each matching fiber pair from hypothesis and ground-truth, we use the areas of each to calculated the Sørensen–Dice Coefficient (DSC) [22, 70] that evaluates the segmentation of matching fibers (overlapping fiber bodies of identified TP):

- Given two areas, $A_{F,GT}$ and $A_{F,H}$, it is defined as

$$DSC = \frac{2 * |A_{F,H} \cap A_{F,GT}|}{A_{F,H} + A_{F,GT}} \tag{3.6}$$

where:

$A_{F,H}$ ... Area of the fiber from hypothesis (prediction)
$A_{F,GT}$ ... Area of the fiber from groundtruth

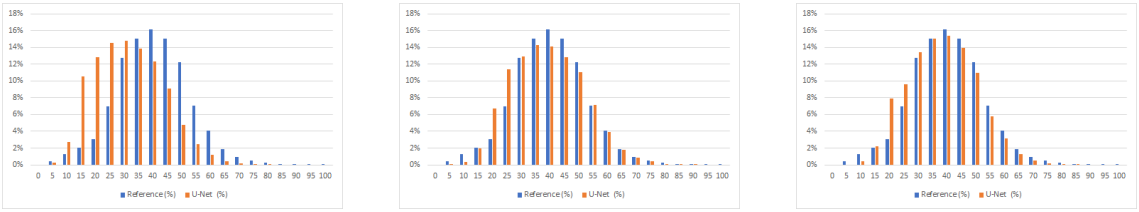
3.2.5.2 Experiments

We choose 10 images that were not used in training the neural network and asked our medical partner to delineate each fiber. In a postprocessing step each fiber was identified (see Figure 3.4b). We used this reference data and compared the result from our neural network prediction output (using the best performing network model from 3.3). We evaluate the correlation from hypothesis to ground-truth at the pixel level in a first step (see Metrics (3.2.4.1)) and at fiber level in a second step.

Name	F1	PREC	REC	SPEC
S34_TA-L1_2	0,87026	0,92082	0,82496	0,95183
S34_TA-L2_2	0,91138	0,92983	0,89365	0,92301
S38_TA-L1_3	0,90315	0,98279	0,83545	0,96630
S38_TA-R2_1	0,93266	0,97186	0,89649	0,94157
S40_TA-L2_2	0,92870	0,97174	0,88932	0,95091
S40_TA-R1_1	0,93952	0,97531	0,90627	0,95065
S44_TA-L1_1	0,93006	0,99144	0,87583	0,97620
S44_TA-R3_3	0,91601	0,96164	0,87453	0,95795
S46_TA-L3_2	**0,94491**	0,96989	0,92118	0,94284
S46_TA-R3_1	0,90839	0,98218	0,84491	0,97136
Average	0,92141	0,96854	0,87866	0,95109

Table 3.4: Direct comparison of U-Net output prediction images (Threshold $T = 200$) (binary) with 10 manually annotated segmentation images at the pixel level.

The good performance of the binary segmentation comparison (Table 3.4) hides the fact of differences in the measured fiber diameters (see Figure 3.5). The average F_{min} in the reference segmentation is 37,86 pixel whereas it is 28,91 pixel in our processing pipeline (a difference of 23,65%).



(a) Feret Diameter Distribution when directly comparing neural network output prediction with reference data

(b) Feret Diameter Distribution after adding a dilation step for each fiber

(c) Feret Diameter Distribution with post processing step that includes the dilation step and parameter optimization (prediction threshold, blurring)

Figure 3.5: Clustered F_{min} distribution. In (a) we see the differences of the distribution without consideration of the different border treatments. (b) displays the distribution after performing a dilation for each fiber (diameter of 11 pixel) and (c) shows further improved results due to optimized post processing parameters

Therefore and in contrast to the previous comparison (Table 3.4), we want to show a more detailed evaluation that takes individual fiber correlation, from ground-truth to hypothesis into account (Table 3.5). For the in depth analysis we compare individual fibers

that we identified as matches from prediction and ground truth. With the calculated fiber mass center and in combination with a defined distance threshold ($T = 20$ pixel), we are able to identify correlating fibers and thus can point out TP, FP and FN detections. For each TP we compute the average matching score from hypothesis and ground truth (Sørensen–Dice Coefficient, DSC, Eq. 3.6) and average Euclidean distance (Eq. 3.1, $\overline{d}(TP)$). Furthermore, we calculate the F1-Score (Eq. 3.4). We ran three experiments:

I. Default neural network output processing parameters, with the threshold set to 200 pixel and the blur kernel size set to 5 pixel (closes smaller holes within fibers).

II. Added a dilation operation to the output generated in I that is applied to every fiber (dilation kernel diameter of 11 pixel).

III. Optimized neural network output processing parameters. Threshold is set to 25 pixel, blur kernel diameter to 11 pixel and dilation kernel diameter set to 11 pixel.

	I.			II.			III.		
Image	DSC	F1	$\overline{d}(TP)$	DSC	F1	$\overline{d}(TP)$	DSC	F1	$\overline{d}(TP)$
`S34_TA-L1_2`	0,8364	0,8362	2,9769	0,8392	0,8332	3,0926	0,8911	0,8672	2,7477
`S34_TA-L2_2`	0,8049	0,8542	2,9651	0,8693	0,8527	2,9985	0,8942	0,8914	2,7339
`S38_TA-L1_3`	0,6761	0,7008	5,0986	0,7664	0,7003	5,2577	0,8085	0,7999	4,5801
`S38_TA-R2_1`	0,7860	0,8477	3,0967	0,8486	0,8414	3,2571	0,8897	0,9053	2,6527
`S40_TA-L2_2`	0,7981	0,8667	2,5854	0,8698	0,8651	2,6437	0,9061	0,8988	2,3646
`S40_TA-R1_1`	0,8221	0,9247	1,9383	0,8787	0,9180	1,9537	0,9178	**0,9509**	**1,7083**
`S44_TA-L1_1`	0,7436	0,7561	3,5824	0,8423	0,7502	3,5611	0,8814	0,8400	3,2548
`S44_TA-R3_3`	0,8423	0,8022	3,1251	0,8956	0,8011	3,1645	0,9019	0,8339	2,9357
`S46_TA-L3_2`	0,9127	0,9326	1,7449	0,8716	0,9248	1,7510	**0,9260**	0,9279	1,7924
`S46_TA-R3_1`	0,8360	0,8157	3,1279	0,8663	0,8471	2,8971	0,8947	0,8632	2,9633
`Average`	0,8058	0,8337	3,0241	0,8548	0,8334	3,0577	0,8911	0,8779	2,7733

Table 3.5: Comparison of U-Net prediction images with 10 corrected segmentation image, on a fiber matching level. For the U-Net model we used the best performing combination from Segmentation: Cross Validation (3.2.4). Between experiment I and II we see an improvement in the Sørensen–Dice Coefficient, but slightly worse results for F1-Score and average Euclidean Distance. On average, the parameter optimization in Experiment III outperforms the other two experiments.

When comparing the fiber identification results we see a large discrepancy in the Sørensen–Dice Coefficient (`S46_TA-L3_2` III: 0,9260 vs `S38_TA-L1_3` I: 0,6761) and F1-Score (`S40_TA-R1_1` III: 95,09% and `S38_TA-L1_3` II: 70,03%). A continuous improvement in the Sørensen–Dice Coefficient is visible from experiments I - III.

An inspection of poor quality segmentation region results in Figure 3.6c- Figure 3.6e indicates that the fiber artifacts (likely a variation of cryo-artifacts see 1.5d) in Figure 3.6a are detected as fiber borders and therefore fibers are inaccurately segmented.

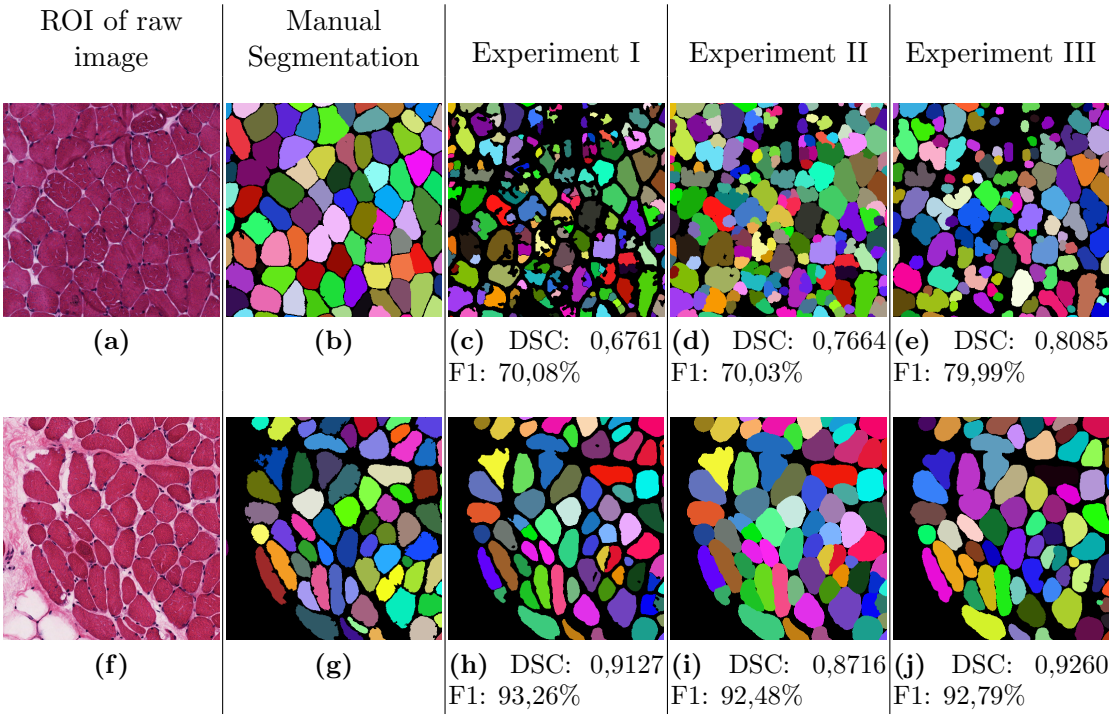


Figure 3.6: (a) - (e): ROI of a bad performing region, taken from *S38_TA-L1_3* in contrast to well performing region taken from *S46_TA-L3_2* in (f) - (j). Even though the individual segmentation outcomes fluctuate (i.e. F1-Score especially for the bottom row, or see Table 3.5), the postprocessing yields an average improvement of 4,42% for F1-Score and 8,53% for DSC (I vs III).

3.2.5.3 Conclusion

A major point for the automated approach is the reduced time from image to segmentation result. For all ten images, the manual preparation required a motivated trained laborer, to continuously draw the contours of each fiber. This required more than a week, whereas the automated segmentation and morphometric extraction is, not optimized, a task of minutes for all ten images and can be applied to an arbitrary number of images.

When analyzing segmentation output, there is a visual difference when comparing the different segmentation examples in Figure 3.4. The most prominent dissimilarities, the gap between each fiber and not detected fibers, have different causes. The gap is a combination of the applied erosion (Postprocessing (2.2.3), Figure 2.16d) and of the parameters in the postprocessing step of the watershed transformation (see the effect of threshold variation in Annotation GUI parameter influence (3.2.6)). To reduce the impact, we used a dilation operation in the postprocessing for each fiber (Figure 3.6d-3.6e and 3.6i-3.6j). As for the not detected fibers, this is due to our focus on eliminating fibers that overlap or are in contact with what we identified as artifacts.

With independent reference segmentation data available (10 manually annotated images), we are able to optimize neural network prediction post-processing parameters to

improve the results (Table 3.5, Experiments I-III). Yet, there are limitations when the post-processing input (neural network prediction) does not provide a good baseline to handle fiber identification as the artifact-regions are too dissimilar to our training data (Figure 3.6a-3.6e).

Another difficult issue is the different expectations from domain experts on how to handle borderline cases. Similar to the results in Segmentation: Quantitative Evaluation of Morphometric Information Extraction (3.2.3), for a valid interpretation the reference and new data should be processed with the same expectations on ambiguous fiber areas segmentation (i.e. artifact-treatment), as comparison becomes difficult otherwise. This is a challenging topic as even experts in the domain of histological fiber annotations create different segmentations, especially in uncertain fiber regions.

We know now that in processing of such slices, the resulting segmentation can be influenced from any point, from cutting the specimen (angle thickness), staining (intensities), freezing (or other artifacts) and respectively the expert or tool. Our Sørensen–Dice score varies from $0,8085$ to $0,8911$ (on average per experiment), where an even higher score may be achieved by further adjusting post processing parameters (see Annotation GUI parameter influence (3.2.6) or appendix B.2.2) whereas low score in individual images requires review of the specific areas that failed and new training data that learns the neural model how to predict yet unknown regions as expected by the domain expert (see Figure 3.6a).

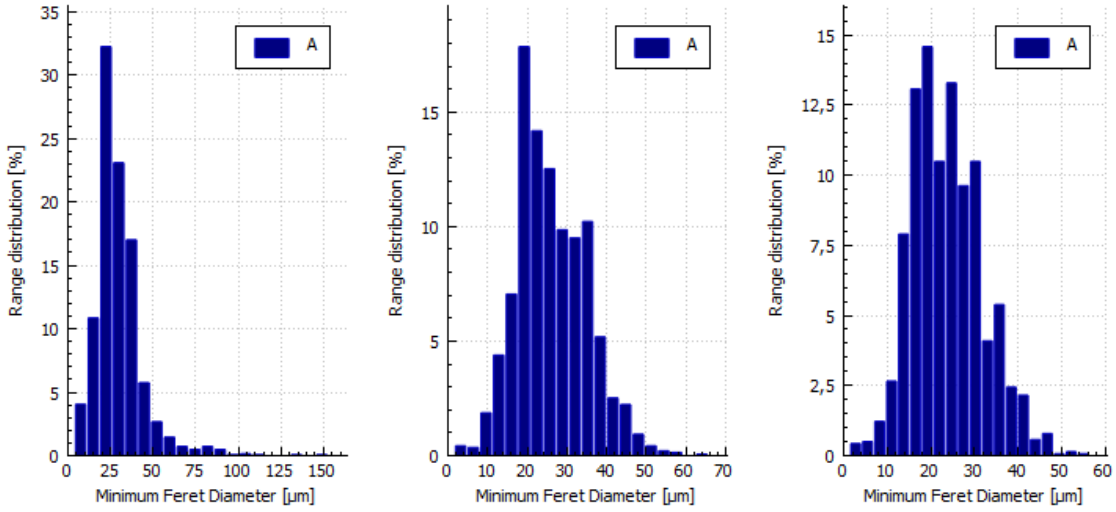
In conclusion, we see that even though we have good results on most images, the next steps for further improvement in regards to robustness are clear. In general, a well performing, time-saving, constant, reproducible (since manual annotation varies from expert to expert) approach which enables comparison should be the goal.

3.2.6 Annotation GUI parameter influence

The *GUI* is used to process the *NN* predictions and extract morphological information. As the preparation for the watershed algorithm allows variations of the parameters, we want to briefly introduce them and their influence.

3.2.6.1 Experiment

We initialize the annotation *GUI* with an image (*S23_TA-L3_3*) and a corresponding prediction of the neural network to display the impact of these parameters.

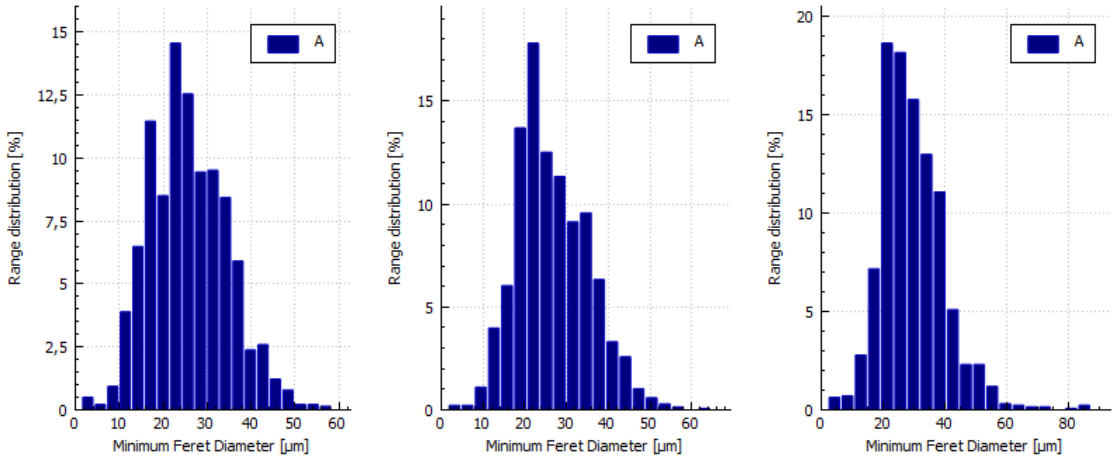


(a) T = 25 **(b)** T = 125 **(c)** T = 225

Figure 3.7: S23_TA-L3_3 (Blur-Kernel=1,Hole-Fill Kernel=1): Influence of the threshold parameter. Large variations of distribution of measured F_{min} are visible. Depending on the quality of the data, a good value for threshold is 200. Image specific parameters are stored for each within a configuration file. Visual control of the resulting segmentation is essential for parameter identification (Figure B.2a).

The largest impact on the measured morphometric information is seen when changing the threshold (Figure 3.7) that is applied on the prediction image. It can be seen as changing the altitude to which the water in the watershed algorithm rises (see Figure 2.4).

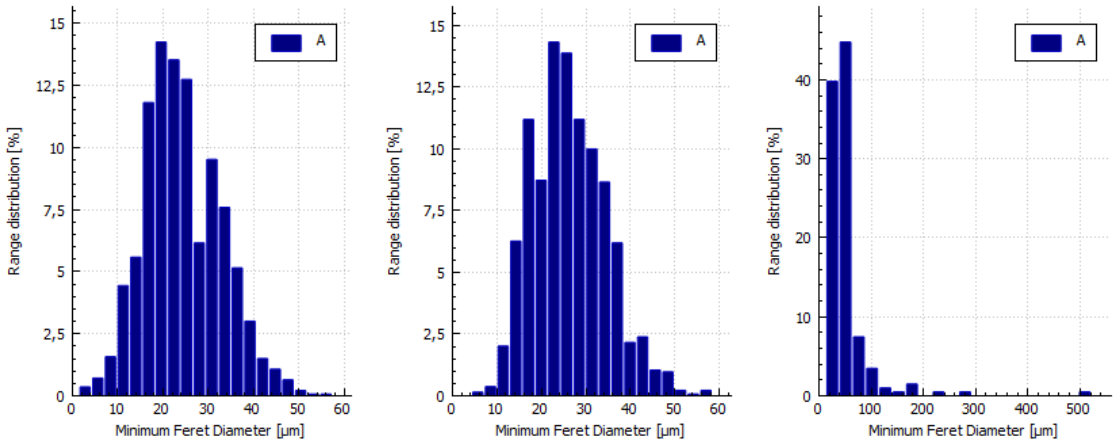
The `Blur-Kernel` is, how the name suggests, used to blur the *NN*-prediction to create smoother edges and close small holes within fibers (i.e. freezing artifacts 1.5d). For prediction images with sharp corners, the blurring is counterproductive. Through the



(a) BK = 1 **(b)** BK = 9 **(c)** BK = 15

Figure 3.8: S23_TA-L3_3 (Threshold=130,Hole-Fill Kernel=1): Influence of the blur-kernel parameter (gaussian blur). The influence of the blurring in conjunction of the threshold is visible, as a shift of the statistics distribution to the left is visible (Figure 3.8a to Figure 3.8b). Further increase in the blur kernel results in merging of fibers (Figure 3.8c, where fibers with a diameter $> 50\ \mu m$ are counted).

applied threshold, proportionally to the `Blur-Kernel`-value, fiber area may be labeled as non-fiber-area (Figure 3.8).



(a) $K_{Dilation} = 1$ **(b)** $K_{Dilation} = 9$ **(c)** $K_{Dilation} = 15$

Figure 3.9: S23_TA-L3_3 (Threshold=200,Blur-Kernel=1): Influence of the Hole-Fill-kernel parameter (dilation of watershed seeds). As soon as HFK reaches a value, where the gaps between fibers vanish, the watershed transformation fails to distinguish fibers (Figure 3.9c).

By changing the value for the `Hole-Fill Kernel` the annotation reviewer can decide the size of the kernel, which is used to dilate the distance transformation, and therefore influences the seeds for the watershed initialization (Figure 2.18c). While this provides a good mechanism to eradicate too close seeds, it is likely to merge fibers.

3.2.6.2 Conclusion

These experiments of the parameters from the annotation update *GUI* target to create an understanding of modifying them. Furthermore, they reflect the influence of parameters when running the watershed transformation. Even though our current choice of parameters yielded decent results, investigation to optimize them may return even better results, but this is a difficult task to achieve, since we lack available test data.

3.3 Conclusion and Outlook

Reflecting on our goal to create a streamlined process to reduce complexity, expenses, and required time while still producing accurate results we proposed a solution that tackles each problem. Even though our results can be used for automated segmentation and extraction of morphometry from *HE*-stained sections, there is room for improvements.

3.3.1 Summary

In this work we started with nothing but *HE*-stained histological sections with the aim to obtain morphometric information. Therefore it was a necessity to identify individual fibers. We selected state of the art machine-learning as segmentation engine that proved to be successful in the field of medical image analysis. We extended the chosen neural network machine learning approach to cope with *HE*-specific features, like intensity shift or boosting training performance by artificially enhancing our training data through image deformation.

To handle our lack of (annotated) training data, we introduced several simple annotation classes that were partly automatically generated and, for the other part, only required a layman to do the vastly reduced manual labor in contrast to manually annotate each fiber.

We started from a first quantitative feasibility check on fiber center detection where we evaluated the performance of our deep learning predictions and deduced that our setup is capable to distinctively identify fibers.

Building on that we proposed a bootstrapping approach to generate a binary segmentation baseline, which relied on the simple (multi class) annotation (that was performed by a layman/automatically). We were able to combine the neural network output (predictions) to create a binary annotation baseline that already took artifacts into consideration. We identified that by reducing the complexity, a baseline segmentation that was created by

a nonprofessional (or automated after training) is a feasible approach that reduces costly expert-time.

The annotation baseline provided us with the necessities to train for our direct binary segmentation, thus enabled us to provide our segmentation result to our medical partner, which can iteratively be used in conjunction with the supportive annotation *GUI* to continuously improve the segmentation and therefore the extracted morphometric information.

All the previous steps combined lead us to an automated *HE*-image processing pipeline that extracted morphometric information. Since the performance in this process is primarily influenced by the segmentation result and that itself depends on the available training data we investigated options for improvement.

To achieve continuous segmentation performance improvements, we incorporated feedback from domain experts (our medical partner) with our simple active learning scheme. Therefore and to reduce the technical entry hurdle, we provided a streamlined graphical user interface where the domain expert applies corrections on annotations for our deep learning setup.

3.3.2 Conclusion

In conclusion, we showed that the combination of layman annotation to create a baseline segmentation and domain expert knowledge to improve it is a feasible way for creating annotation data in a deep learning setup. We proposed an image processing pipeline that extracts morphometric information and targets, with focused feedback, to iteratively improve the result.

We tackled the lack of annotation data, by dividing the problem into smaller ones. We identified what can automatically be detected (background), what can easily be manually refined (additional background, fiber centers, artifacts) and how to combine that into a segmentation of individual fibers. Even though this yields decent results, in border cases it is not yet clear what can be declared a valid fiber or not. Furthermore, (regions of) images which are not close to our training data, may yield unwanted results.

The feedback loop we propose, is not yet automated. We provide a *GUI* that has a focus to improve the segmentation task, nevertheless the annotation updates require to be manually transfered back to the training setup. For the training setup it is not tested, if a training from scratch with the updated data or a continued training would be better.

3.3.3 Outlook

Several further evaluations and applications are worthy of consideration. First, our bootstrapping approach that is used to generate learning data as baseline. We expect our process to be applicable to similar problems. Next is the incorporation of domain expert to create an active learning setup. Since our approach consists of interchangeable modules, improvements can be performed independently. Depending on the targeted segmentation,

the preprocessing for bootstrapping may vary. Improvement or changing of the machine learning setup may yield better results. Since the input for the annotation *GUI* can be arbitrary, as long as prediction and raw image are provided, a modification or change of the prediction generator can be applied and may provide better results. A further aspect of our work we want to work on in the future will be to extend our bootstrapping approach to a full active learning interaction loop with minimal intervention of a domain expert, while still providing high performance in segmentation of the thus refined deep learning model. Next to improving the quality of segmentation result, increasing the user experience by offering the computation of segmentation results through a web service should be thought of.

A

List of Acronyms

CNN	Convolutional Neural Networks
CPU	Central Processing Unit
DNN	Deep Neural Networks
DoF	Degree of Freedom
F	Feret Diameter
F_{min}	Minimum Feret Diameter
FES	Functional Electrical Stimulation
GAC	Geodesic Active Contours
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HE	Haematoxylin Eosin
HSV	Hue Saturation Value
MLS	Moving Least Square
NN	Neural Network
RGB	Red Green Blue
ROI	Region Of Interest
TAM	Thyroarytenoid Muscle
TSV	Tab Separated Value

B

Implementation Detail

For our image processing pipeline (see Figure 2.1), we identified two modules where we can contribute through our implementations. First we enhanced the used *CNN*, to gain robustness and then created a portable annotation update *GUI* for the continuous segmentation improvement. Both extensions can run independently, but the *GUI* requires raw input image and a corresponding ground-truth (prediction), which can be provided by the binary pixelwise prediction from the *CNN*, as used Direct Fiber Segmentation (2.3). We will explain the setup regarding our modification in detail:

- Neural Toolbox Modifications: Enhancements for robustness

  We use the existing neural-toolbox provided by [76] built upon Caffe [39] which offers a huge degree of parallelism by training and evaluation on *GPU* as well as extended imaging support. Our extensions[1] to the existing code[2] were written in C++.

- Annotation *GUI*: User friendly editing of annotations

  The *GUI* is written in C++ and relies on OpenCV[3], Qt 5[4] and runs on Linux and Windows based operating systems.

B.1 Neural Toolbox Modifications

The neural toolbox draws random patches out of the provided image set. Each patch can be modified by a preprocessor, which we enhanced, before passing through the different layers (Figure 2.11) of the *NN*. To configure our implemented modifications, one adds sections to the respective (training and/or processing) configuration filed, identified by the **.prototxt* file extension. Since our annotation augmentation modifications aim to improve training results, applying them while testing is nonsensical.

[1] https://bitbucket.org/derKlaus/, 25.10.2017
[2] https://github.com/naibaf7/caffe_neural_tool, 25.10.2017
[3] https://opencv.org/, 25.10.2017
[4] http://doc.qt.io/, 25.10.2017

Library	Version
CUDA	8.0
OpenBLAS	0.2
OpenCL	1.2
OpenCV	3.1

Table B.1: Neural toolbox library requirements

The required additional libraries, next to Caffe, to run the neural toolbox can be seen in Table B.1. Setup and implementation was done on Ubuntu 16.04 LTS.

B.1.1 Setup

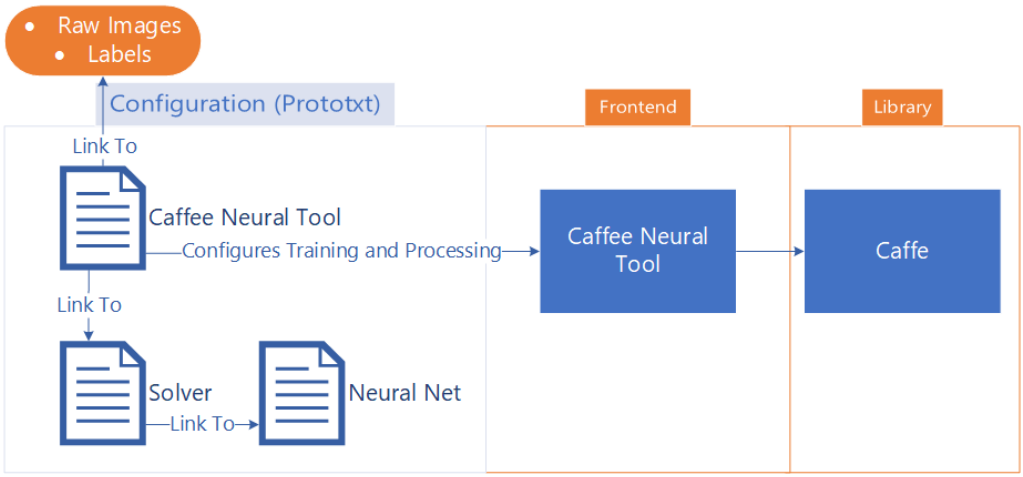


Figure B.1: The Caffe Neural Tool requires a configuration file 'Caffe Neural Tool Configuration' which itself links to the data (raw images and label images), 'Solver Configuration' and indirectly to the 'Net Configuration' (e.g. U-Net model see Figure 2.11). All document types are saved in the Google's prototxt network and learning configuration format.

Setup of the neural network pipeline to process training data requires several configuration steps. The structural setup can be seen in Figure B.1.

Caffe Neural Tool Configuration The neural tool configuration file contains definitions for training and process runs. Our modifications can be configured in the image processing section.

Solver Configuration Links to neural network configuration (model), defines learning rate, momentum and neural network weight snapshot parameters (enables pause and

continue for training).

Neural Net Configuration Description of the used neural net (layers), e.g. the U-Net model.

B.1.2 Modifications

Our extensions to the neural toolbox target the image pre-processor. To augment training data, we added intensity-shift, to gain image intensity invariance, and random geometric transformation, to artificially increase our training data. Furthermore, we added support for varying image sizes (limited by network configuration and available memory).

Image Intensity Invariance To apply the changed intensity, the drawn patch is converted to Hue Saturation Value (HSV) color model and apply equation 2.5. To configure the usage and setup up parameters edit the `train.prototxt`'s parameter within the `input`'s `preprocessor` and add a section called `intshift` like:

```
intshift {
use_hsv: false
range: 30
}
```

use_hsv: if *true* stays in *HSV* color model, otherwise the patch is converted back to Red Green Blue (RGB) color model (default)

range: defines a random variation of 0-*range* of hue modification which is applied to the drawn patch (see Figure 2.12)

Image Deformation Rotations (by a multiple of 90°) and mirroring operation to the drawn patches are already available. In case of few annotated images (i.e. only a small set from the iterative feedback) to further supply image data for training, image deformation is of great assistance (see Figure 2.13). Configuration is again done within the `train.prototxt`'s, `input`'s `preprocessor` parameter. Add a section called `elasticDeformation` like:

```
elasticDeformation {
mode: "affine"
cols: 10
rows: 5
max_variation: 0.5
}
```

mode: supported deformation modes are affine, rigid and similarity

rows, cols: defines in combination with rows/cols the grid points to be used to randomly move

max_variation: maximum random variation $\in [0, 1]$, where 1 can theoretically result in overlapping grid points and 0 equals no deformation. $P_{Grid,moved} = P_{Grid} + (random * d - \frac{d}{2}) * variation_{max}$ where d is the distance, defined through image dimension and chosen cols and rows, from one grid point to the next.

Image Dimension Invariance It is not necessary to apply any configuration. For training it is required that ground truth and raw image size match.

B.2 Annotation GUI

The annotation *GUI* targets to streamline and simplify the annotation update process, by focusing on the required image modification tools while working with the tuple of raw image and predicted annotation.

Setup and implementation were done on Ubuntu 16.04 and Windows 10. As integrated development environment (IDE) Qt Creator[5] was used.

Library	Version
Qt	5
OpenCV	3.1
Boost	1.64

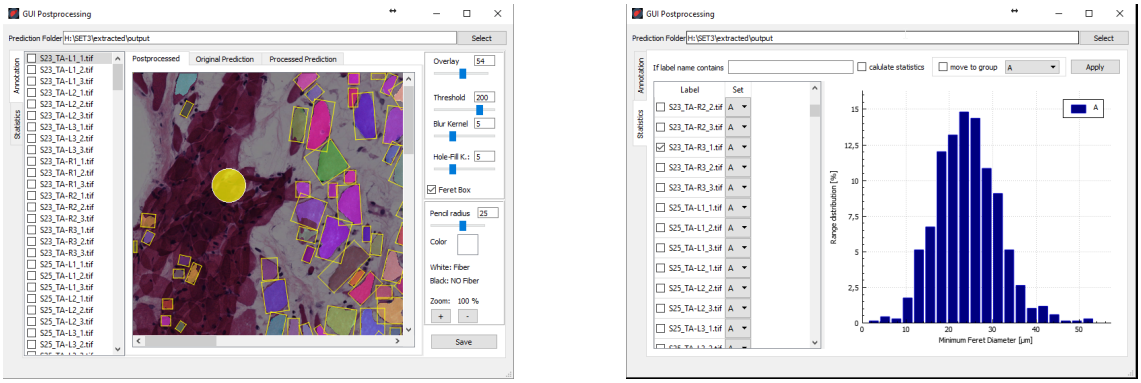
Table B.2: *GUI* libary requirements

The additional required libraries to compile the annotation *GUI* can be seen in Table B.2.

B.2.1 Overview

In the main view of the *GUI*, the annotation tab is initially displayed (Figure B.2a). At the top, the prediction folder which shows the path to the selection data, is shown. Below, the working space is split into three main areas (left to right): a list of images and their save-state, the current selected editing view and parameters which either influence the preprocessing (top) or the actual manual editing (bottom).

[5] https://www.qt.io/, 25.10.2017



(a) Main View (Editing) **(b)** Statistics View

Figure B.2: Overview of annotation and statistics *GUI*

The working area `Annotation`[6] supports three different views: `Postprocessed` where each distinct identified fiber if colored, `Original Prediction` which shows the gray scale prediction image of the $NN \in [0, 255]$ and `Processed Prediction` which is the result of the `Threshold` (right top box) applied to the `Original Prediction`. Any of those three views supports overlaying of the raw input image (`Overlay` $\in [0, 100]$ at right top box).

The left side shows the list of images and supports navigation, whereas at the right side of the *GUI* in the top box parameters for `Overlay`, `Threshold`, `Blur Kernel`, `Hole-Fill Kernel` can be set. `Feret Box` indicates the minimum bounding rectangle (yellow squares in B.2a) used for F_{min} calculation.

At the bottom box, parameters for editing within the working area can be set. The `Pencil radius`, and `Color` (white: fiber area, black: no fiber, B.2a).

The second tab `Statistics` is used to view and export statistics (Figure B.2b). To support display of different groups, and ease use when there are hundreds of images grouping and calculation can be done by a simple text search. Supported export formats are '.png' for images and '.tsv' which is a raw, un-grouped format but can easily be imported by other programs. The scaling factor for images can be defined by adding within the configuration file `CONFIG.ini` in the section `DefaultParameters`, a key `StatisticsScale` with an fitting value. Since the scale is the same for current image sets, the used factor, if not defined otherwise is 0.98 ($pixel * 0.98 = \mu m$).

B.2.2 Prediction Processing

In the context of this application, prediction processing implies preparation of the prediction image for the watershed transformation. It has proven to be useful to apply changes to `Threshold`, `Blur Kernel`, `Hole-Fill-Kernel` to fine tune the segmentation result. We provide a user-friendly parameter modification which is stored, if different from default

[6] Any text in the `typewriter font` can be found as label in the application

values, for each image in a configuration file (in the application path, `CONFIG.ini`).

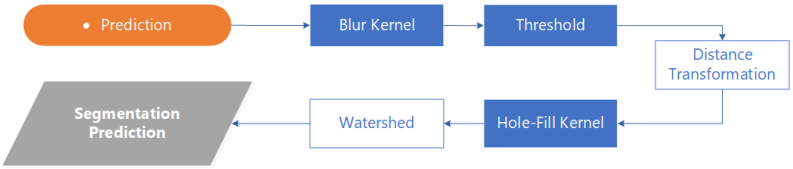


Figure B.3: Orange: Input images, Blue (filled): Modifyable parameters, Blue (empty): automated step, Gray: Visible segmentation result in the application. The main parameters which influence the watershed segmentation result can be edited with the *GUI*.

Since the input is a raw image and ground truth pair, and the ground truth's origin is from the *CNN* prediction, we added on the fly parameter modification, which influence the segmentation result. The prediction processing workflow can be seen in Figure B.3.

B.2.3 Editing

Changing of the annotation should be done after the parameters for the segmentation prediction are set. Features include zooming (scroll-wheel, or B.2a right bottom box), changing of editing circle `Pencil Radius` (right bottom box) as well as `Color`. To ease verification if an area should or should not be a fiber, `overlay` of the original raw image is available, and can be switched by hitting the space-bar. Undo and redo of manual painted annotations can be done by hitting CTRL+Z or CTRL+Y respectively. The pencil itself is a transparent yellow circle, where the border color (black or white) indicates what kind of change will be applied to the current ground-truth. In the working area editing is possible in all three tabs `Postprocessed` , `Original Prediction` and `Processed Prediction`.

B.2.4 Configuration

The program requires image pairs as input. Raw images are used for `Overlay` within the respective view and cut to the *ROI* when saving an updated annotation area, and ground truth prediction images, which correspond by name to the raw images.

The program uses following folder structure:

`/Output/` [7]: Contains the network predictions

`/Input/` [7]: Contains raw images

`/Processed/` [8]: Contains prediction images

[7] Mandatory
[8] This, and any folder below, will be created at runtime

`/Processed/edited/` : Folder which mirrors common Caffe [39] training input structure

`/Processed/edited/raw/` : *ROI* of raw image

`/Processed/edited/labels/` : *ROI* of modified ground truth

The base folder for `/Output/` can be set in the configuration file `CONFIG.ini` (in the application folder) by setting the `RootFolder` parameter:

```
[Session]
RootFolder=./data/output
[DefaultParameters]
Overlay=80
Thresh=200
Blur=5
HoleFillKernel=5
PencilRadius=25
CalculateStatistics=false
```

Changing these parameters within the configuration is optional, since they are set automatically when using the *GUI*. Default parameters, which are initially used for each image are individually saved for each image. As best practice, an application user should be able to identify if any changes are necessary and apply them.

C

Dataset A

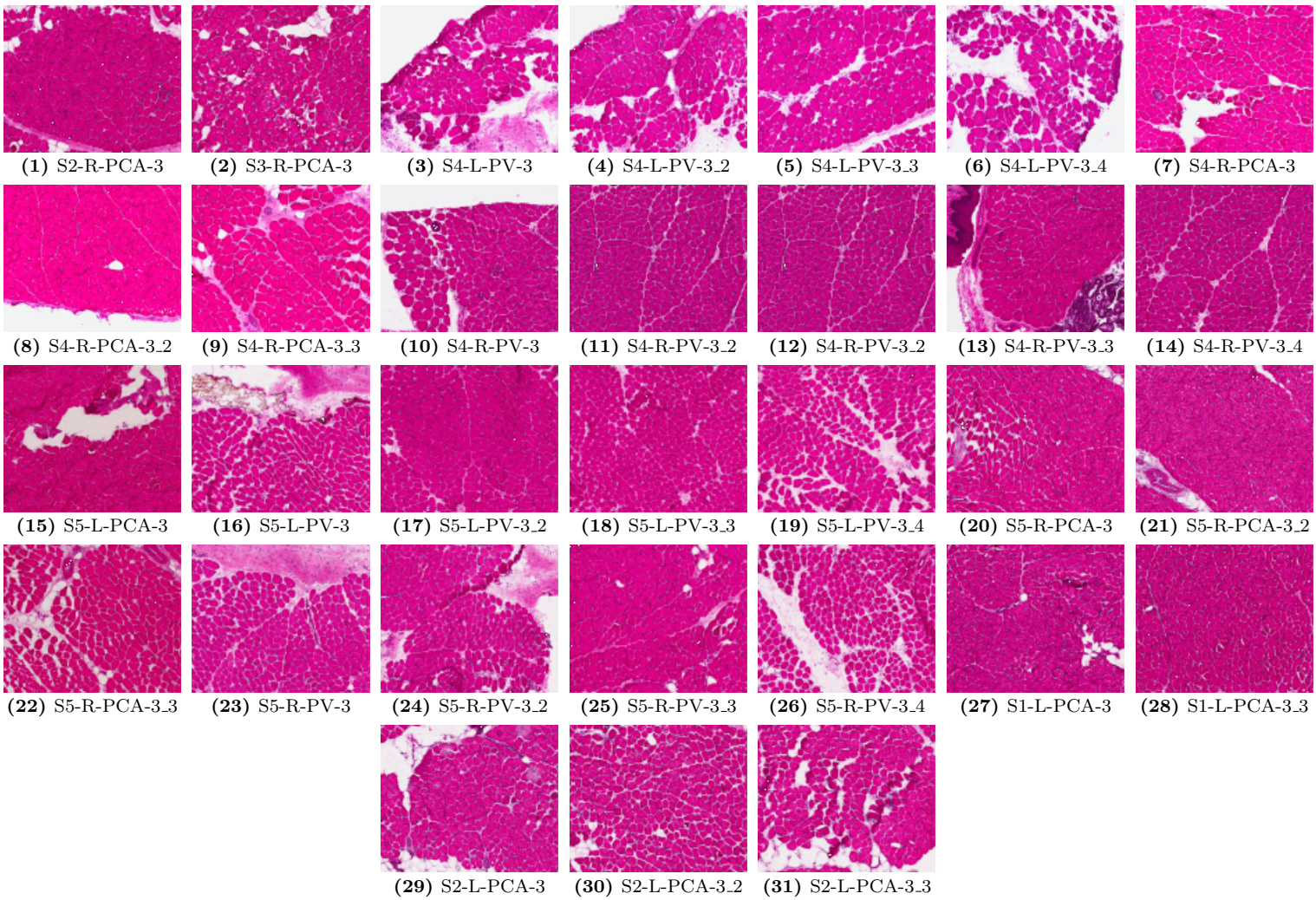
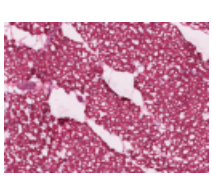
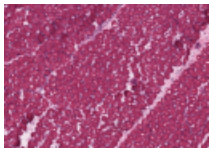
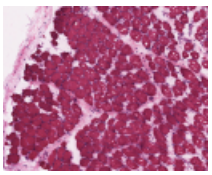
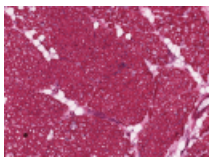


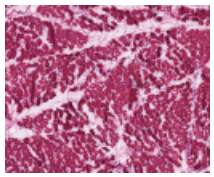
Figure C.1: Batch I: Overview of all available histological sections. Image dimension and staining have no extensive variation.

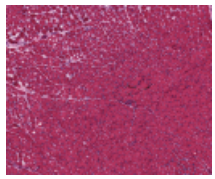

(1) S1-L-PV-3

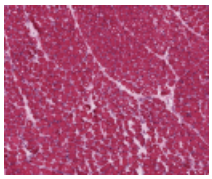

(2) S1-R-PV-2

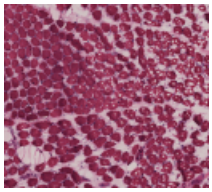

(3) S1-R-PV-2_2

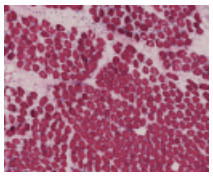

(4) S2-L-PV-2

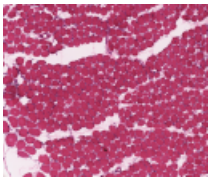

(5) S2-L-PV-2_2

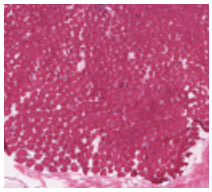

(6) S2-R-PV-2

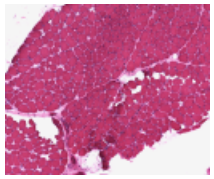

(7) S2-R-PV-2_2

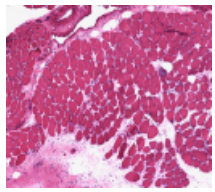

(8) S3-L-PV-2

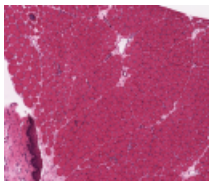

(9) S3-L-PV-2_2

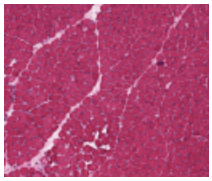

(10) S3-R-PV-2

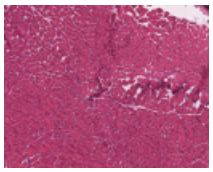

(11) S3-R-PV-2_2

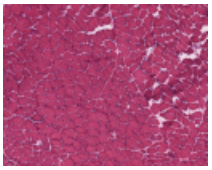

(12) S4-L-PV-2

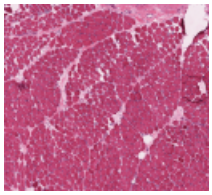

(13) S4-L-PV-2_2

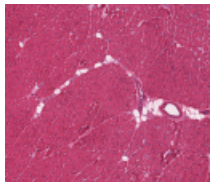

(14) S4-R-PV+2

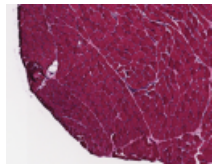

(15) S4-R-PV+2_2

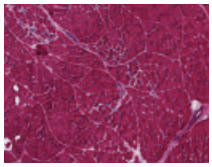

(16) S5-L-PV-2

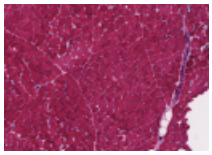

(17) S5-L-PV-2_2

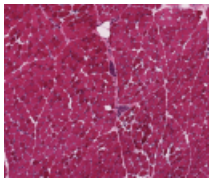

(18) S5-R-PV-2

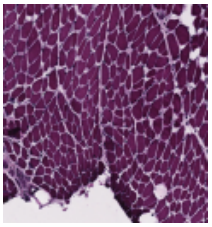

(19) S5-R-PV-2_2

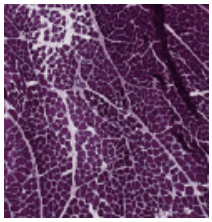

(20) Y1 PCA-L L 1-1_001

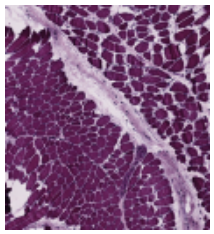

(21) Y1 PCA-L L 1-1_002

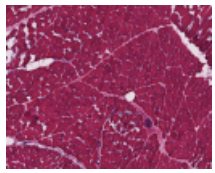

(22) Y1 PCA-L R 1-1_001

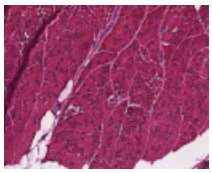

(23) Y1 PCA-L R 1-1_002

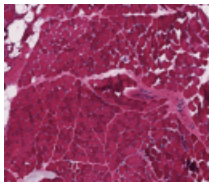

(24) Y1 VOC L 1-1_001

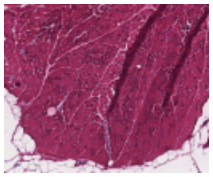

(25) Y1 VOC R 1-1_001

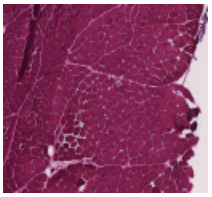

(26) Y1 VOC R 1-1_002

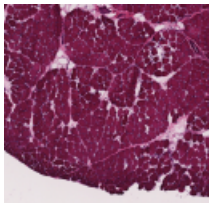

(27) Y2 PCA-L L 1-1_001

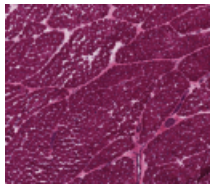

(28) Y2 PCA-L L 1-1_002

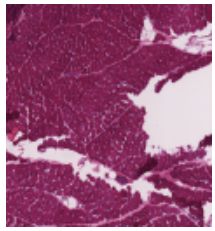

(29) Y2 PCA-L R 1-1_001

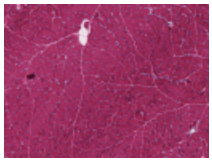

(30) Y2 PCA-L R 1-1_002

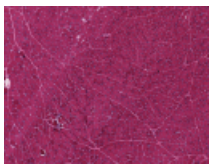

(31) Y2 VOC-L 1_001

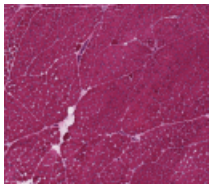

(32) Y2 VOC-L 1_002

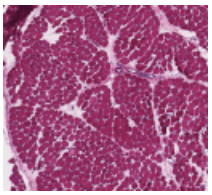

(33) Y2 VOC-R 1_001

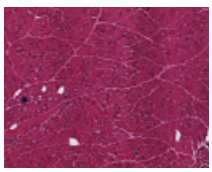

(34) Y2 VOC-R 1_002

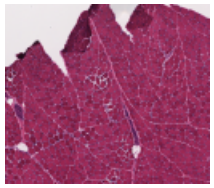

(35) Y3 L PCA 1-1_001


(36) Y3 L PCA 1-1_002


(37) Y3 L VOC 1-1_001


(38) Y3 L VOC 1-1_002


(39) Y3 R PCA 1-1_001


(40) Y3 R PCA 1-1_002

(41) Y3 R VOC 1-1_001

(42) Y3 R VOC 1-1_002

(43) Y4 L PCA 1-1_001

(44) Y4 L PCA 1-1_002

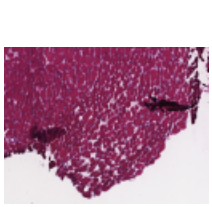
(45) Y4 L VOC 1-1_001

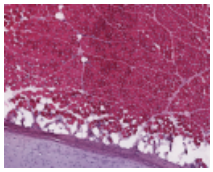
(46) Y4 L VOC 1-1_002

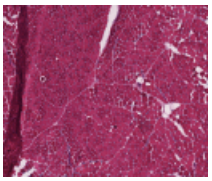
(47) Y4 R PCA 1-1_001

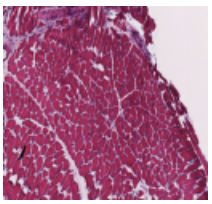
(48) Y4 R PCA 1-1_002

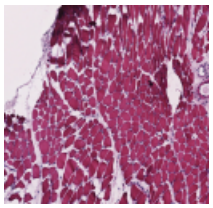
(49) Y4 R VOC 1-1_001

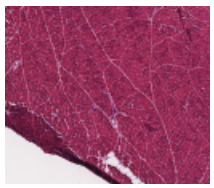

(50) Y4 R VOC 1-1_002

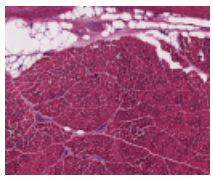

(51) Y5 L PCA 1-1_001

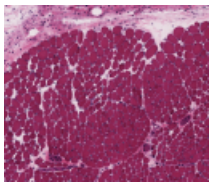

(52) Y5 L PCA 1-1_002

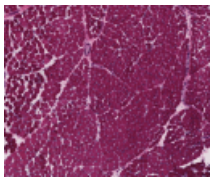

(53) Y5 L VOC 1-1_001

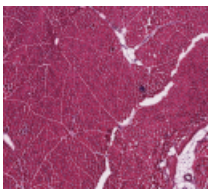

(54) Y5 L VOC 1-1_002

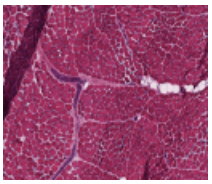

(55) Y5 R PCA 1-1_001

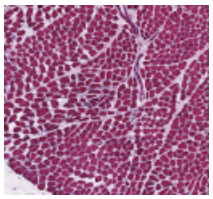

(56) Y5 R PCA 1-1_002

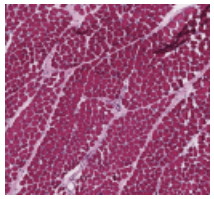

(57) Y5 R VOC 1-3_001

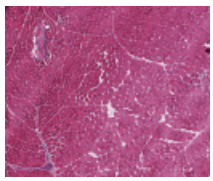

(58) Y5 R VOC 1-3_002

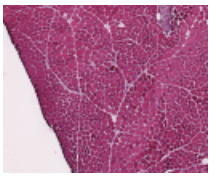

(59) Y6 L PCA 1-1_001

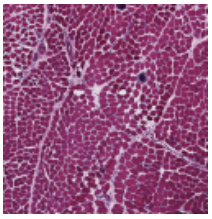

(60) Y6 L PCA 1-1_002


(61) Y6 L VOC 1-1_001


(62) Y6 L VOC 1-1_002


(63) Y6 R PCA 1-1_001


(64) Y6 R PCA 1-1_002


(65) Y6 R VOC 1-1_001

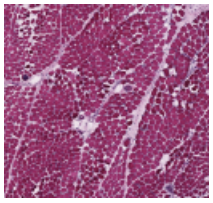

(66) Y6 R VOC 1-1_002

Figure C.2: Batch II: Overview of all available histological sections. A variation in dimension and staining intensity is discernible.

Bibliography

[1] Aeffner, F., Wilson, K., Martin, N. T., Black, J. C., Hendriks, C. L. L., Bolon, B., Rudmann, D. G., Gianani, R., Koegler, S. R., Krueger, J., et al. (2017). The gold standard paradox in digital image analysis: manual versus automated scoring as ground truth. *Archives of Pathology & Laboratory Medicine*, 141(9):1267–1275. (page 10, 24)

[2] Arganda-Carreras, I., Turaga, S. C., Berger, D. R., Cireşan, D., Giusti, A., Gambardella, L. M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J. M., Liu, T., Seyedhosseini, M., Tasdizen, T., Kamentsky, L., Burget, R., Uher, V., Tan, X., Sun, C., Pham, T. D., Bas, E., Uzunbas, M. G., Cardona, A., Schindelin, J., and Seung, H. S. (2015). Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, 9:142. (page 11, 15)

[3] Barth, R., IJsselmuiden, J., Hemming, J., and Van Henten, E. (2017). Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Computers and Electronics in Agriculture*. (page 11)

[4] Belafsky, P. C. and Postma, G. N. (2004). Vocal fold augmentation with calcium hydroxylapatite. *American Academy of Otolaryngology-Head and Neck Surgery*, 131(4):351–354. (page 3)

[5] Bennie, S. D., Petrofsky, J. S., Nisperos, J., Tsurudome, M., and Laymon, M. (2002). Toward the optimal waveform for electrical stimulation of human muscle. *European Journal of Applied Physiology*, 88(1-2):13–19. (page 3)

[6] Beucher, S. (1982). Watersheds of functions and picture segmentation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*, volume 7, pages 1928–1931. IEEE. (page 10)

[7] Bezerra, P., Zhou, S., Crowley, Z., Brooks, L., and Hooper, A. (2009). Effects of unilateral electromyostimulation superimposed on voluntary training on strength and cross-sectional area. *Muscle & Nerve*, 40(3):430–437. (page 3)

[8] Boykov, Y. Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112. IEEE. (page 10)

[9] Bradley, J. P., Hapner, E., and Johns, M. M. (2014). What is the optimal treatment for presbyphonia? *The Laryngoscope*, 124(11):2439–2440. (page 3)

[10] Carding, P. N., Horsley, I. A., and Docherty, G. J. (1999). A study of the effectiveness of voice therapy in the treatment of 45 patients with nonorganic dysphonia. *Journal of Voice*, 13(1):72–104. (page 3)

[11] Castleman, K. R., Chui, L. A., Martin, T. P., and Edgerton, V. R. (1984). Quantitative muscle biopsy analysis1. *Monographs in Clinical Cytology*, 9:101–116. (page 10)

[12] Chatterjee, S. (2014). Artefacts in histopathology. *Journal of Oral and Maxillofacial Pathology (JOMFP)*, 18(Suppl 1):S111–S116. (page 8, 27)

[13] Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc. (page 15)

[14] Ciresan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 411–418. (page 11)

[15] Collobert, R., Bengio, S., and Mariéthoz, J. (2002). Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, IDIAP. (page 8)

[16] Collumbien, R., Zukowski, F., Claeys, A., and Roels, F. (1990). Automated analysis of muscle fibre images. *European Society for Analytical Cellular Pathology*, 2(6):373–387. (page 10)

[17] Colson, S. S., Martin, A., and Van Hoecke, J. (2009). Effects of electromyostimulation versus voluntary isometric training on elbow flexor muscle strength. *Journal of Electromyography and Kinesiology*, 19(5):e311–e319. (page 3)

[18] Cordier, N., Delingette, H., Lê, M., and Ayache, N. (2016). Extended modality propagation: image synthesis of pathological cases. *IEEE transactions on medical imaging*, 35(12):2598–2608. (page 11)

[19] Costantini, M., Sciallero, S., Giannini, A., Gatteschi, B., Rinaldi, P., Lanzanova, G., Bonelli, L., Casetti, T., Bertinelli, E., Giuliani, O., et al. (2003). Interobserver agreement in the histologic diagnosis of colorectal polyps: the experience of the multicenter adenoma colorectal study (SMAC). *Journal of Clinical Epidemiology*, 56(3):209–214. (page 10, 24)

[20] de Araújo Pernambuco, L., Espelt, A., Balata, P. M. M., and de Lima, K. C. (2015). Prevalence of voice disorders in the elderly: a systematic review of population-based studies. *European archives of Oto-Rhino-Laryngology*, 272(10):2601–2609. (page 1)

[21] Dhawan, A. P. (2011). *Medical Image Analysis*, volume 31. John Wiley & Sons. (page 6, 7, 10, 21)

[22] Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302. (page 58)

[23] Dudley, A. W., Spittal, R. M., Dayhoff, R. E., and Ledley, R. S. (1984). Computed image analysis techniques of skeletal muscle. *Methods and Achievements in Experimental Pathology*, 11:34–57. (page 10)

[24] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874. (page 45)

[25] Feierabend, R. H. and Shahram, M. N. (2009). Hoarseness in adults. *American Family Physician*, 80(4):363–370. (page 2, 3)

[26] Felzenszwalb, P. F. and Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of Computing*, 8(1):415–428. (page 38)

[27] Foley, J. D., Van Dam, A., Feiner, S. K., and Hughes, J. F. (1996). *Computer Graphics: Principles and Practices. Second Edition in C.* Addison-Wesley Publishing Company. (page 30)

[28] Ford, C. N. and Bless, D. M. (1986). Clinical experience with injectable collagen for vocal fold augmentation. *The Laryngoscope*, 96(8):863–869. (page 3)

[29] Gaur, U., Kourakis, M., Newman-Smith, E., Smith, W., and Manjunath, B. S. (2016). Membrane segmentation via active learning with deep networks. In *International Conferencce on Image Processing (ICIP)*, pages 1943–1947. IEEE. (page 12)

[30] Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., 2nd edition. (page 7, 10)

[31] Gurcan, M. N., Boucheron, L. E., Can, A., Madabhushi, A., Rajpoot, N. M., and Yener, B. (2009). Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 2:147–171. (page 10)

[32] Gutman, D., Codella, N. C. F., Celebi, M. E., Helba, B., Marchetti, M. A., Mishra, N. K., and Halpern, A. (2016). Skin lesion analysis toward melanoma detection: A challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC). *CoRR*, abs/1605.01397. (page 11)

[33] Habibzadeh, M., Krzyżak, A., and Fevens, T. (2013). White blood cell differential counts using convolutional neural networks for low resolution images. In *International Conference on Artificial Intelligence and Soft Computing*, pages 263–274. Springer Berlin Heidelberg. (page 11)

[34] Hagen, P., Lyons, G. D., and Nuss, D. W. (1996). Dysphonia in the elderly: diagnosis and management of age-related voice changes. *Southern Medical Journal*, 89(2):204–207. (page 1)

[35] Hammernik, K., Ebner, T., Stern, D., Urschler, M., and Pock, T. (2015). Vertebrae segmentation in 3d ct images based on a variational framework. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pages 227–233. Springer. (page 18)

[36] Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196. (page 11)

[37] Holloszy, J. O. (2000). The biology of aging. *Mayo Clinic proceedings*, 75:3–8. (page 1)

[38] Isshiki, N., Shoji, K., Kojima, H., and Hirano, S. (1996). Vocal fold atrophy and its surgical treatment. *The Annals of Otology, Rhinology, and Laryngology*, 105(3):182–188. (page 3)

[39] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM. (page 8, 12, 22, 23, 28, 31, 46, 71, 77)

[40] Johns, M. M., Arviso, L. C., and Ramadan, F. (2011). Challenges and opportunities in the management of the aging voice. *American Academy of Otolaryngology-Head and Neck Surgery*, 145(1):1–6. (page 1)

[41] Kainz, P., Pfeiffer, M., and Urschler, M. (2017). Segmentation and classification of colon glands with deep convolutional neural networks and total variation regularization. *PeerJ*, 5:e3874. (page 11)

[42] Karbiener, M., Jarvis, J. C., Perkins, J. D., Lanmüller, H., Schmoll, M., Rode, H. S., Gerstenberger, C., and Gugatschka, M. (2016). Reversing age related changes of the laryngeal muscles by chronic electrostimulation of the recurrent laryngeal nerve. *PloS one*, 11(11):e0167367. (page 4, 5, 41)

[43] Kim, Y.-J., Brox, T., Feiden, W., and Weickert, J. (2007). Fully automated segmentation and morphometrical analysis of muscle fiber images. *Cytometry Part A*, 71(1):8–15. (page 10)

[44] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. (page 15)

[45] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. (page 7)

[46] LeCun, Y., Kavukeuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *International Symposium on Circuits and Systems (ISCAS)*, pages 253–256. IEEE. (page 11)

[47] Lein, D. H., Myers, C., and Bickel, C. S. (2015). Impact of varying the parameters of stimulation of 2 commonly used waveforms on muscle force production and fatigue. *Journal of Orthopaedic and Sports Physical Therapy*, 45(8):634–641. (page 3)

[48] Lillie, R. D. (1947). *Histopathologic technic and practical histochemistry*. Blakiston; New York. (page 5, 6, 10)

[49] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440. (page 11, 15)

[50] Malatesta, D., Cattaneo, F., Dugnani, S., and Maffiuletti, N. A. (2003). Effects of electromyostimulation training and volleyball practice on jumping ability. *Journal of Strength and Conditioning Research*, 17(3):573–579. (page 3)

[51] Malon, C. D. and Cosatto, E. (2013). Classification of mitotic figures with convolutional neural networks and seeded blob features. *Journal of Pathology Informatics*, 4. (page 11)

[52] Melton, L. J., Khosla, S., Crowson, C. S., O'Connor, M. K., O'Fallon, W. M., and Riggs, B. L. (2000). Epidemiology of sarcopenia. *Journal of the American Geriatrics Society*, 48(6):625–630. (page 1)

[53] Meng, H., Janssen, P. M. L., Grange, R. W., Yang, L., Beggs, A. H., Swanson, L. C., Cossette, S. A., Frase, A., Childers, M. K., Granzier, H., Gussoni, E., and Lawlor, M. W. (2014). Tissue triage and freezing for models of skeletal muscle disease. *Journal of Visualized Experiments*, 89:e51586. (page 8, 27)

[54] Meyer, F. (1992). Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303–306. IET. (page 19, 20)

[55] Mula, J., Lee, J. D., Liu, F., Yang, L., and Peterson, C. A. (2012). Automated image analysis of skeletal muscle fiber cross-sectional area. *Journal of Applied Physiology*, 114(1):148–155. (page 10)

[56] Pham, D. L., Xu, C., and Prince, J. L. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2(1):315–337. (page 7, 11)

[57] Prince, J. L. and Links, J. M. (2006). *Medical imaging signals and systems*. Pearson Prentice Hall Upper Saddle River. (page 7)

[58] Prince, S. J. (2012). *Computer vision: models, learning, and inference*. Cambridge University Press. (page 11)

[59] Remacle, M. and Eckel, H. E. (2010). *Surgery of larynx and trachea*. Springer. (page 3)

[60] Roerdink, J. and Meijster, A. (2000). The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*, 41(1-2):187–228. (page 20)

[61] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer. (page 11, 12, 28)

[62] Schaefer, S., McPhail, T., and Warren, J. (2006). Image deformation using moving least squares. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 533–540. ACM. (page 31, 32)

[63] Schenk, F., Aichinger, P., Roesner, I., and Urschler, M. (2015). Automatic high-speed video glottis segmentation using salient regions and 3D geodesic active contours. *Annals of the British Machine Vision Association and Society for Pattern Recognition (BMVA)*, 2015(3):1–15. (page 10)

[64] Schoenfeld, B. J. (2013). Is there a minimum intensity threshold for resistance training-induced hypertrophic adaptations? *Sports Medicine*, 43(12):1279–1288. (page 4)

[65] Sertel, O., Kong, J., Catalyurek, U. V., Lozanski, G., Saltz, J. H., and Gurcan, M. N. (2009). Histopathological image analysis using model-based intermediate representations and color texture: Follicular lymphoma grading. *Journal of Signal Processing Systems*, 55(1-3):169. (page 11)

[66] Sertel, O., Kong, J., Lozanski, G., Çatalyürek, Ü. V., Saltz, J. H., and Gurcan, M. N. (2008). Computerized microscopic image analysis of follicular lymphoma. In *Medical Imaging 2008: Computer-Aided Diagnosis*, page 691535. International Society for Optics and Photonics. (page 11)

[67] Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. (page 12)

[68] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556. (page 45)

[69] Sonka, M., Hlavac, V., and Boyle, R. (2014). *Image processing, analysis, and machine vision*. Cengage Learning, Fourth edition. (page 7, 21)

[70] Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter, Kongelige Danske Videnskabernes Selskab*, 5:1–34. (page 58)

[71] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. (page 45)

[72] Thomas, G., Dixon, M., Smeeton, N., and Williams, N. (1983). Observer variation in the histological grading of rectal carcinoma. *Journal of Clinical Pathology*, 36(4):385–391. (page 10, 24)

[73] Titze, I. R. (1994). *Principles of Voice Production*. Prentice Hall. (page 3)

[74] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE. (page 34)

[75] Tosta, T. A. A., Neves, L. A., and do Nascimento, M. Z. (2017). Segmentation methods of h&e-stained histological images of lymphoma: A review. *Informatics in Medicine Unlocked*, 9:35–43. (page 10)

[76] Tschopp, F. (2015). Efficient convolutional neural networks for pixelwise classification on heterogeneous hardware systems. *CoRR*, abs/1509.03371. (page 12, 22, 23, 25, 28, 29, 31, 43, 46, 71)

[77] Tulloch, L., Perkins, J., and Piercy, R. (2011). Multiple immunofluorescence labelling enables simultaneous identification of all mature fibre types in a single equine skeletal muscle cryosection. *Equine Veterinary Journal*, 43(4):500–503. (page 5, 10)

[78] Unger, M., Pock, T., Trobin, W., Cremers, D., and Bischof, H. (2008). Tvseg - interactive total variation based image segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 31, pages 44–46. Citeseer, BMVA Press. (page 11, 18)

[79] Vachier, C. and Meyer, F. (2005). The viscous watershed transform. *Journal of Mathematical Imaging and Vision*, 22(2-3):251–267. (page 19)

[80] Van Putten, P. G., Hol, L., Van Dekken, H., Han van Krieken, J., Van Ballegooijen, M., Kuipers, E. J., and Van Leerdam, M. E. (2011). Inter-observer variation in the histological diagnosis of polyps in colorectal cancer screening. *Histopathology*, 58(6):974–981. (page 10, 24)

[81] Verdonck-de Leeuw, I. M. and Mahieu, H. F. (2004). Vocal aging and the impact on daily life: a longitudinal study. *Journal of Voice*, 18(2):193–202. (page 1)

[82] Volpi, E., Nazemi, R., and Fujita, S. (2004). Muscle tissue changes with aging. *Current Opinion in Clinical Nutrition and Metabolic Care*, 7(4):405–410. (page 1)

[83] Wahl, P., Hein, M., Achtzehn, S., Bloch, W., and Mester, J. (2015). Acute effects of superimposed electromyostimulation during cycling on myokines and markers of muscle damage. *Journal of Musculoskeletal & Neuronal Interactions*, 15(1):53–59. (page 3)

[84] Xie, W., Noble, J. A., and Zisserman, A. (2018). Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 6(3):283–292. (page 11)

[85] Xu, S., Liu, H., and Song, E. (2011). Marker-controlled watershed for lesion segmentation in mammograms. *Journal of Digital Imaging*, 24(5):754–763. (page 10, 19)

[86] Yamasaki, T., Honma, T., and Aizawa, K. (2017). Efficient optimization of convolutional neural networks using particle swarm optimization. In *Multimedia Big Data (BigMM), 2017 IEEE Third International Conference on*, pages 70–73. IEEE. (page 45)

[87] Yu, D., Eversole, A., Seltzer, M., Yao, K., Guenter, B., Kuchaiev, O., Seide, F., Wang, H., Droppo, J., Huang, Z., Zweig, G., Rossbach, C., and Currey, J. (2014). An introduction to computational networks and the computational network toolkit. *Microsoft Technical Report MSR-TR-2014–112*. (page 8)

[88] Ziegler, A., Verdolini Abbott, K., Johns, M., Klein, A., and Hapner, E. R. (2014). Preliminary data on two voice therapy interventions in the treatment of presbyphonia. *The Laryngoscope*, 124(8):1869–1876. (page 3)