Zuse Institute Berlin

Takustrasse 7
D-14195 Berlin-Dahlem
Germany

Anika Rettig[1], Tobias Haase[2], Alexandr Pletnyov[2], Benjamin Kohl[2], Wolfgang Ertel[2], Max von Kleist[1], and Vikram Sunkara[1,3]

[1] *Systems Pharmacology and Disease Control, Freie Universität Berlin, Berlin, Germany*

[2] *Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin, Humboldt-Universität zu Berlin, and Berlin Institute of Health, Department of Traumatology and Reconstructive Surgery, Campus Benjamin Franklin, Berlin, Germany*

[3] *Computational Medicine, Zuse Institute Berlin, Berlin, Germany*

SLCV–A Supervised Learning - Computer Vision

combined strategy for automated muscle fibre detection in cross sectional images

ZIB Report 19-10 (February 2019)

Zuse Institute Berlin
Takustrasse 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

SLCV–A Supervised Learning - Computer Vision combined strategy for automated muscle fibre detection in cross sectional images

Anika Rettig[1], Tobias Haase[2], Alexandr Pletnyov[2], Benjamin Kohl[2], Wolfgang Ertel[2], Max von Kleist[1], and Vikram Sunkara[1,3]

[1]Systems Pharmacology and Disease Control, Freie Universität Berlin, Berlin, Germany
[2]Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin, Humboldt-Universität zu Berlin, and Berlin Institute of Health, Department of Traumatology and Reconstructive Surgery, Campus Benjamin Franklin, Berlin, Germany
[3]Computational Medicine, Zuse Institute Berlin, Berlin, Germany
[3]Email: sunkara@mi.fu-berlin.de

February 13, 2019

Abstract

Muscle fibre cross sectional area (CSA) is an important biomedical measure used to determine the structural composition of skeletal muscle, and it is relevant for tackling research questions in many different fields of research. To date, time consuming and tedious manual delineation of muscle fibres is often used to determine the CSA. Few methods are able to automatically detect muscle fibres in muscle fibre cross sections to quantify CSA due to challenges posed by variation of brightness and noise in the staining images. In this paper, we introduce SLCV, a robust semi-automatic pipeline for muscle fibre detection, which combines supervised learning (SL) with computer vision (CV). SLCV is adaptable to different staining methods and is quickly and intuitively tunable by the user. We are the first to perform an error analysis with respect to cell count and area, based on which we compare SLCV to the best purely CV-based pipeline in order to identify the contribution of SL and CV steps to muscle fibre detection. Our results obtained on 27 fluorescence-stained cross sectional images of varying staining quality suggest that combining SL and CV performs significantly better than both SL based and CV based methods with regards to both the cell separation- and the area reconstruction error. Furthermore, applying SLCV to our test set images yielded fibre detection results of very high quality, with average sensitivity values of 0.93 or higher on different cluster sizes and an average Dice Similarity Coefficient (DSC) of 0.9778.

1 Introduction

Skeletal muscle and its adaptation to diverse stimuli plays a central role in various biological processes and disease states. Analysing the structural composition of skeletal muscle specimens is essential in many fields of research, ranging from basic developmental and physiological sciences to muscular and metabolic diseases like myopathies [18]. In preclinical models and studies in humans, one of the central elements in the characterization of muscle specimens is the analysis of the muscle fibre size (fibre cross sectional area, CSA) [18, 21, 24]. CSA allows for the assessment of muscle hypertrophy, atrophy and weakness. Despite its importance, the quantification of such muscle cell characteristics is still often done manually by multiple blinded observers–the muscle fibres are delineated by hand using software such as ImageJ [20]. This is a time consuming and labour intensive task, especially if multiple cross sections have to be analysed for a research task. Methods which aid in automating the process are available, but require large amounts of manual error correction and are often not free, for example the Zeiss AxioVison software. Automatic classification of muscle fibre cross sections is a hard problem due to variation with regards to staining quality and noise. There are three major factors which pose challenges to automated muscle fibre detection approaches: brightness, borders, and image stitching.

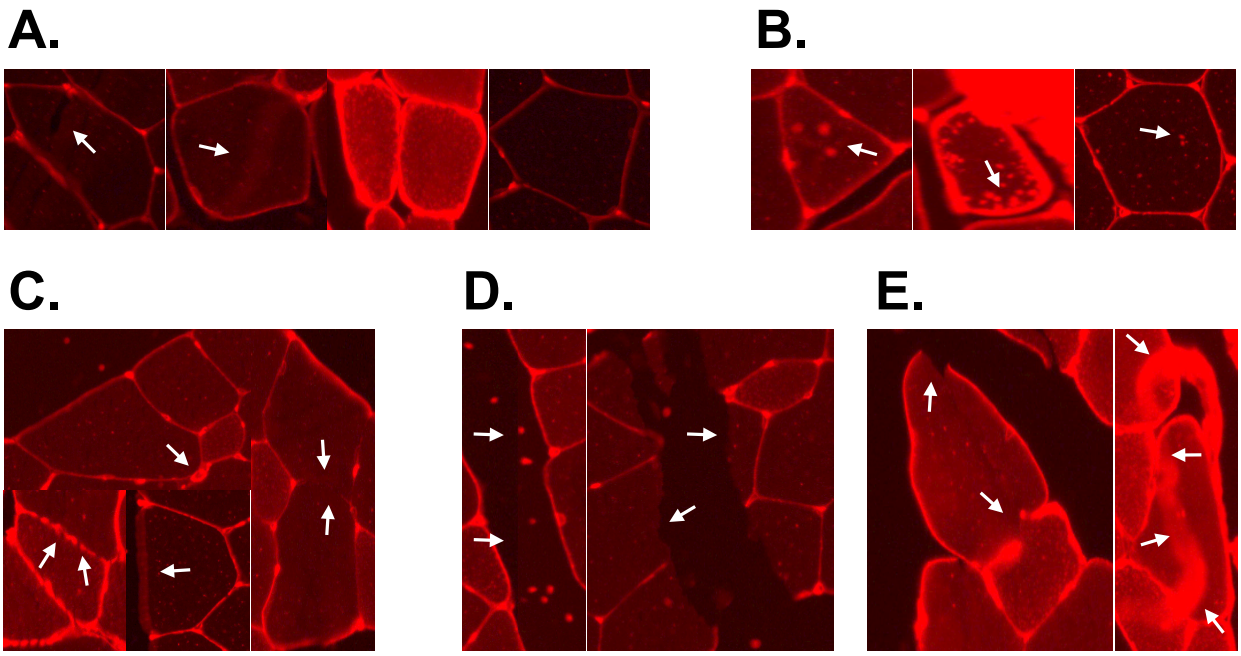

Figure 1. A: Examples for variation in cell tissue intensity. Arrows point at holes in cell tissue which disturb the cell detection process (left) and intensity variation within one cell (second from left). B: Different intensities and distributions of noise. C: Examples for varying border staining intensities. Arrows show irregularities, such as a border between two cells, which is so weakly stained, that it is hardly visible (right). D: Examples of noise in gap regions: Sudden intensity changes inside the cross section along a vertical line through the image, which is due to image stitching (left). Low contrast between gap region and cell tissue due to ripped tissue poses a challenge to cell detection (right). E: Other factors, such as non-continuous cell borders (left) and very bright staining artifacts (right).

Firstly, in fluorescence-stained pictures, brightness of cell tissues varies strongly between and even within pictures (see Figure 1A). As an example, we observed a general trend of small fibres appearing brighter than big fibres. Biological reasons for the variation of brightness within a picture might be the autofluorescence of tissue, or the wheat germ agglutinin (WGA) used for staining, which generally binds to glycoproteins of the cell membrane [7]. It can thus also stain membrane vesicles, which would show up as small bright spots in the cytosol of the fibres. This form of noise within the fibres is differently distributed and is thus hard to filter out (see Figure 1B). Secondly, cell borders can contain weakly stained areas, which appear as holes, and in extreme cases the whole border may be hard to spot on the image (Figure 1C). Additionally, there are interstitial spaces (gaps) between the muscle fibre bundles in most cross sections, which are expected to be devoid of the staining protein. However, these areas often contain noise (Figure 1D) and the resulting low intensity contrast between the gap and neighbouring cell tissue may hinder the correct detection of the cell borders. Another possible source of variation is the process of picture recording. When the whole image is reconstructed from multiple smaller pictures, these pictures are recorded separately using automatic brightness detection. The stitched cross section may thus contain sudden changes in intensity, where two small pictures recorded with different settings are assembled (Figure 1D). All of these aforementioned factors are part of the challenge that needs to be overcome to realise effective and practical automatic fibre cross sectional detection.

In the past years, methods for automatic or semi-automatic cell detection on different staining techniques have been introduced: Smith et al. used immunohistochemical staining [22], Mula et al. presented their method on examples of both immunohistochemical staining and WGA fluorescence-staining [17], and Liu et al. used haematoxylin and eosin (H&E) staining [14]. While these algorithms greatly facilitate cell detection in their respective settings, there is still no overall solution addressing all three aforementioned challenges. Furthermore, the trade-off between automation and quick appli-

cation on the one hand, and the adaptability and possibility of human intervention on the other hand is balanced differently between the methods, thus not each method might fit the respective needs of a user.

The pipeline proposed in this paper is called SLCV, and combines supervised learning (SL) with computer vision (CV). It does require human intervention, but is very intuitively and quickly adjustable with respect to different staining methods and staining qualities, while being robust to noise and variation. The novelty of this work is firstly to combine the adaptability and robustness of machine learning methods with the accuracy of computer vision methods on images and secondly, to perform a statistical error analysis with respect to cell separation and area reconstruction. We set up a comparison study between our SLCV pipeline and that of Mula et al. [17], since to our knowledge it is the best purely computer vision based automated fibre detection method to date. We present both pipelines on the example of fluorescence-stained images, which Mula et al. have also used in their paper.

We begin by introducing both the SLCV pipeline and the pipeline of Mula et al., highlighting their key characteristics and similarities. Then, we describe how the test images were created, and define the error measurements of cell separation and area reconstruction used for the statistical analysis. Finally, we compare results we obtained using both pipelines on our image test set, and discuss these results.

2 Methods

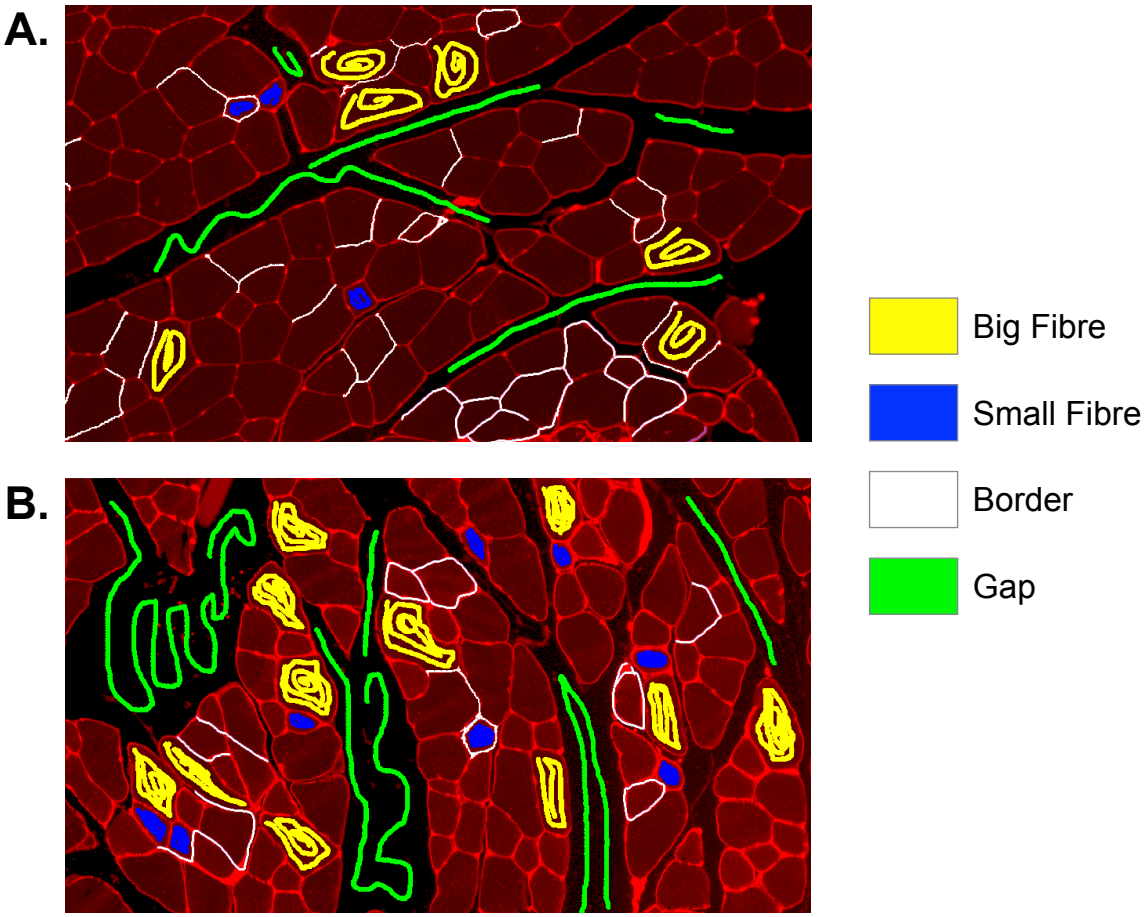

Figure 2. Training set of Ilastik classifier 1 out of 2, which was applied to 25 out of 27 of the test set images. Yellow: class "big fibre", blue: class "small fibre", white: class "border", green: class "gap". A: First training image. B: Second training image.

2.1 SLCV Pipeline

2.1.1 Step I: Supervised Learning

The first step to identifying cells in the muscle cross section picture is detecting the cell borders. To do this, a random forest supervised learning model for pixel classification is used. Pixel classification models are available as one of several workflows in Ilastik–an open-source image analysis, classification, and segmentation software [23]. The trained model assigns every pixel of an image to one of several previously defined classes, which describe the different textures within the cross sections to be analysed. In fluorescence-stained images, we defined the classes to be "border" (very bright and thin regions), "gap" (big dark areas), "big fibre" (bigger areas of low brightness) and "small fibre" (smaller bright regions). Only pixels of the class "border" are used for the following steps, but defining all four classes and training the model to distinguish them optimises the accuracy of the assignment of pixels to the "border" class. The classification model is based on two selected image features: the eigenvalues of the Hessian of Gaussian with $\sigma_{texture} = 1.6, 3.5, 5.0, 10.0$ px which are used to detect regions of intensity changes, and Gaussian image smoothing with $\sigma_{intensity} = 0.3, 0.7, 1.0, 1.6, 3.5$ px. The pixel (px) breadth values of σ are chosen from 7 different values available in the feature selection dialogue of Ilastik, and are selected to cover a range of pixel breadths for each feature. Based on these features, the training is conducted on only few selected pixels from a set of example pictures. Ilastik features a visual interface, where the training pixels can be drawn on the image, making the model easily adjustable to different kinds of staining and quick and intuitive to create (see §A.1 for how an Ilastik classifier is recommended to be created). All classification done in this work was performed using no more than small amounts of training on two images, as shown in Figure 2. This Figure shows the training of classifier 1, which was applied to 25 test set images. The two training images in the Figure are examples of muscle fibre cross sections, and were chosen such that they cover the common intensity and noise levels found in borders, gaps and fibres in the test set. The first image mostly contains noise-free gaps, fibres with low, equally distributed noise and thin or faint borders. In contrast, the second image contains noisy gaps, very bright borders and fibres with higher noise levels of different distribution. Classifier 2 was trained on two images in a similar fashion, and was applied to 2 test set images which showed much higher overall brightness and lower border quality than the other test images. Both images chosen for the training of classifier 2 also mainly contained high brightness fibres and thin and faint borders to match the test images and are shown in §A.2.

The output of step I are all pixels which Ilastik classified as "border". We refer to these as the initial borders or initial clusters, with both representations being equivalent. We define the clusters as follows: clusters are the smallest objects (with respect to set inclusion) in the picture, which are each completely encapsulated by a continuous sequence of touching border pixels. The clusters represent initial fibre detection results and can consist of more than one true fibre if the borders contain holes. An example of the relation between borders and clusters is shown in Figure 3, where Fig. 3B shows the borders resulting from the classification of Fig. 3A, and Fig. 3C shows the cluster obtained from the borders in Fig. 3B.

2.1.2 Step II: Watershed

The output of the previous step are the initial borders or clusters, where one cluster can contain one or multiple true fibres. The aim of this second step is to identify single fibres by refining or separating these initial clusters. This is achieved by filling holes in borders which were not completely detected by the supervised learning pixel classification. First, very small holes are filled by dilating the borders. The binary image is then subjected to the following distance transformation: Each non-border pixel within a cluster is given a value according to its minimal distance to a border pixel [4]. This transformation yields a "hill-like" structure, which has one local maximum if the cluster is round, or several local maxima, if the boundary is irregular and the cluster is thus likely to contain multiple single cells. A subsequent thresholding step removes small distance values and leaves connected components

representing the maxima. Let x and y be the coordinates of a pixel inside an image $\hat{I}$. Then,

$$I(x,y) := \begin{cases} \hat{I}(x,y) & \text{if } \hat{I}(x,y) \geq \tau \times max(\hat{I}), \\ 0 & \text{else} \end{cases}$$

where I is the thresholded image $\hat{I}$. The bigger the threshold $\tau \in [0,1]$ is selected, the more components result and the easier small irregularities within the cluster boundary lead to cluster separation. The connected components are subsequently input to the watershed algorithm [5]. The idea of the watershed algorithm is to "flood", that is, to steadily extend all connected components outwards. Flooding is stopped in regions, where either the component touches a border pixel or two different components touch each other. A visualization of input, distance transformation and output are shown in Figure 3 C–E, respectively. The result of this step are the final cell clusters.

2.1.3 Step III: GAC Snake

The previous two steps separated touching muscle cells in order to obtain single true cells. However, fibre area is lost in this process due to errors introduced by the approximations of previous steps and thus the aim of the last step is to accurately reconstruct the fibre area. To do this, the *geodesic active contours* (GAC) Snake model–an evolving 2-dimensional deformable curve–is used [2]. It is based on a partial differential equation (PDE), which is solved repeatedly until its overall energy is minimized. The PDE has the form:

$$\frac{\partial u}{\partial t} = \underbrace{g(I)|\nabla u| div\left(\frac{\nabla u}{|\nabla u|}\right)}_{\text{smoothing force}} + \overbrace{g(I)|\nabla u|\,\nu}^{\text{balloon force}} + \overbrace{\nabla g(I) \nabla u}^{\text{image attraction force}}$$

It consists of three parts, which modify the deformable 2D curve u: a smoothing function, a balloon force and an image attraction term. The balloon force is controlled by the parameter ν and is used to expand the snake u outward (or to contract it, if $\nu < 0$), while the image attraction term $g(I)$ draws the curve towards image features of interest and acts as a stopping criterion. We used Marquez-Neila e al.'s implementation of the algorithm [16]. One factor that distinguishes the GAC Snake from other Snake models is the usage of a morphological method for solving the PDE, which is quicker and numerically more stable than conventional numerical methods [2].

When edges in a picture are used as an image attraction force like in our case, then

$$g(I) = \frac{1}{\sqrt{1 + \alpha|\nabla G_\sigma \otimes I|}}, \tag{1}$$

which results in $g(I)$ having its minima near regions with high intensity changes. In the above equation, $\otimes$ represents the convolution operator. There are two parameters to be set, and we chose $\alpha = 2000$ and $\sigma = 2$ after testing different parameters on the image set. Additional parameter choices are given in §A.3. The input of the Snake algorithm is called *seed*, and its boundary represents the initial configuration of the curve u. Starting from the final clusters from step II as seeds, the cluster boundaries are expanded iteratively until they reach the cell borders (see Figure 3F).

In summary, the SLCV pipeline starts by applying the Ilastik random forest supervised learning (SL) method to an input image in order to obtain the initial clusters. The subsequent steps are computer vision (CV) based and correct and refine the SL output. The clusters are submitted to distance transformation and the watershed algorithm, with the aim of separating initial clusters into final single-cell clusters. The area which is lost in the separation process is then reconstructed by applying the GAC Snake model to each final cluster, where the resulting fibres are obtained by growing the cluster using information from the original image. An example of an input and output of the pipeline is shown in Figure 4.

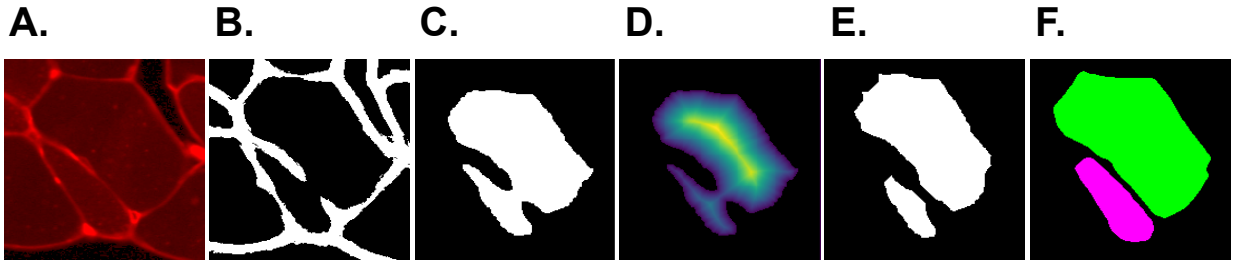

Figure 3. A: Sector of an original image after manual thresholding. B: White pixels were classified as "border" by the random forest segmentation model. C: Initial cluster. D: Distance transformation of the non-border pixels. The following thresholding was conducted with parameter $\tau = 0.3$. E: Results of applying the watershed algorithm to the thresholded image yields two final clusters. F: GAC Snake reconstruction of the two final clusters.

2.2 Pipeline of Mula et al.

Similarly to the SLCV pipeline, Mula et al.'s pipeline consists of three phases which follow the same objectives as the SLCV phases, respectively. In the first step, the algorithm detects ridges, which are regions of intensity changes in the picture. This is achieved by first convoluting the input picture with a Gaussian kernel and then calculating the eigenvalues of the Hessian matrix of this convoluted image. From the eigenvalues, likelihood measures are obtained and subjected to automatic *Otsu Thresholding* [19]. In the second step, the initial borders are morphologically closed to fill very small holes. Then the resulting clusters are subjected to iterative erosion until their size falls below a given threshold. This strategy tends to split all but very round clusters into multiple regions. Thus, touching cells with medium-sized or big holes in their borders are separated, and a set of final clusters of similar size is obtained. The third step applies the Snake algorithm to the final clusters to reconstruct the cell area. We implemented the pipeline of Mula et al. according to the description in the paper [17]. Within the procedure, there were several parameters to be set. For some of them, a recommendation was given, in which case we set them accordingly. All other parameters were set such that we reached the best segmentation result on our test pictures. The parameters are given in §A.3. We also performed some changes on the pipeline: We changed the described multi-scale ridge detection of step I to single-scale detection with $\sigma^* = 0.7$, because this parameter captured the borders best and minimized the noise in the resulting ridges. Furthermore, Mula et al. originally used the gradient vector flow (GVF) version of snake [25,26] in step III, which is similar to the GAC Snake, because both use a smoothing term and an image attraction force. However, GVF Snake lacks the balloon force term and does not always converge to the edges, if the initial seed is too small. Furthermore, if it is implemented according to the description in the original paper by Xu et al., it is less numerically stable than the GAC Snake, since it doesn't use morphological methods to solve the PDE. Therefore, and to simplify the comparison of the two pipelines, we used the morphological GAC Snake model instead of the GVF Snake in our implementation of Mula et al.'s pipeline.

2.3 Picture Test Set

Complete hindlimbs from male C57BL/6J mice aged 20 weeks were dissected and fixed in paraformaldehyde for 48 h at 4 $°C$ to keep the knee joint and muscles in their natural position. The specimens were decalcified for 10 days in 14% $EDTA$ at 4 $°C$ on a shaker. After dehydration, joints were embedded in paraffin and serial cross sections (5 μm) through the whole hindlimb musculature were done. Cross sections were mounted on slides, stained with fluorescent-labeled wheat germ agglutinin (WGA Alexa-Fluor 555, Thermo) and visualised using a slide scanner (Hamamatsu NanoZoomer).

The fluorescence-stained images created by this method have very similar properties to immunohistochemically stained images. Instead of fluorescence-labeled WGA, which binds to glycoproteins in the membrane, two types of antibodies are used to create immunohistochemically stained images. The

first antibody binds to a specific protein, for example dystrophin, which appears in the membranes of muscles. The second fluorescent protein binds to the first antibody to visualize the binding [9]. Hence, noise and variation are comparable between both methods, and SLCV and Mula et al.'s pipeline can be compared using only one of the two types of staining.

Our test set contains 27 images of different staining quality, including noisy and low-quality images. This is evidenced by the examples from Figure 1, which are all taken from the test set. The raw images are submitted to a manual thresholding step (see §A.4). The contrast between gaps and fibres is maximized in this step in order to assure the best possible performance of the CV methods. In each image, the maximum threshold was chosen such that no holes appeared in any muscle fibre. The resulting images serve as the test set. Furthermore, a corresponding groundtruth picture was created for each test set image by an experienced biologist who manually delineated all fibres. Any fibre which was not completely contained in the picture was omitted in both the test set image and the groundtruth image in order to obtain an unbiased fibre sample.

2.4 Error Measure: Cluster Separation

A simple image gradient analysis method was applied to each image to obtain the border pixels. We then defined the reference clusters to be the smallest objects in the image fully encapsulated by border pixels, equivalent to the definition in §2.1.2. The reference clusters were grouped by the number of groundtruth cells n which they contain to represent separation difficulty. Only reference clusters with $n > 1$ were kept in the considered test set. Furthermore, cluster sizes for which there were fewer than 5 samples in the test set were omitted from the statistical analysis. We chose ridge detection with one additional dilation as the gradient analysis method (see §A.3). To assess the separation quality of a particular pipeline, the reference clusters were compared to the fibres detected by the pipeline. The cluster separation error is a sensitivity measure, which is computed for each picture and each reference cluster size n using a contingency table as shown below:

	Positive (P)	**Negative (N)**
True (T)	a	$c = 2^n - 1 - a - b - d$
False (F)	b	$d = n - a$

The term a denotes the number of true positives, meaning the number of true cells in the reference cluster of size n, which were correctly detected by the algorithm. False negatives are denoted by $d = n - a$ and describe the number of true cells not detected by the algorithm, either because the cell was missing completely in the result or because it could not be separated correctly. The false positives b are the number of clusters found by the algorithm, which contain more than one groundtruth cell. Finally, c is based on the size of the result space including every possible way to separate the reference cluster, from which all existing results are subtracted. The sensitivity is calculated as follows: $sensitivity := \frac{a}{a+d}$. This measure captures how many cells within a reference cluster were separated correctly and is thus representative of the separation quality of a pipeline. In order to quantify the sensitivity difference between pipeline pl_1 and pipeline pl_2, we assume H_0: "The average sensitivity of pl_2 is $\geq$ the average sensitivity of pl_1." For each of the cluster sizes n, this hypothesis is tested by bootstrapping: 10^5 bootstrap iterations are conducted on all reference cluster samples N. The *p-value* of this test is defined to be the number of bootstrap iterations in which H_0 is true, divided by the total number of bootstrap resamples.

2.5 Error Measure: Area Reconstruction

The second type of error analysis in this work is based on the groundtruth of each test set image. The area of the groundtruth cell is compared to the calculated cross sectional area (CSA) for each cell that was correctly separated by all pipelines that are compared. This is done by calculating the Dice similarity coefficient (DSC), which is defined as follows:

$$DSC := \frac{2|X \bigcap Y|}{|X| + |Y|}, \tag{2}$$

where X is the area of the groundtruth cell and Y the area of the reconstructed cell [6]. The DSC is a measure for the similarity of two areas and punishes deviations of a reconstructed cell from the original cell with respect to both size as well as location in the picture. Differences in area reconstruction between two methods were quantified similar to sensitivity differences. We assumed H_0: "The average DSC of Mula et al.'s pipeline is $\geq$ the average DSC of SLCV", and tested it in each of the 10^5 bootstrapping iterations. The *p-value* is the fraction of bootstrapping resamples in which H_0 is true.

3 Results

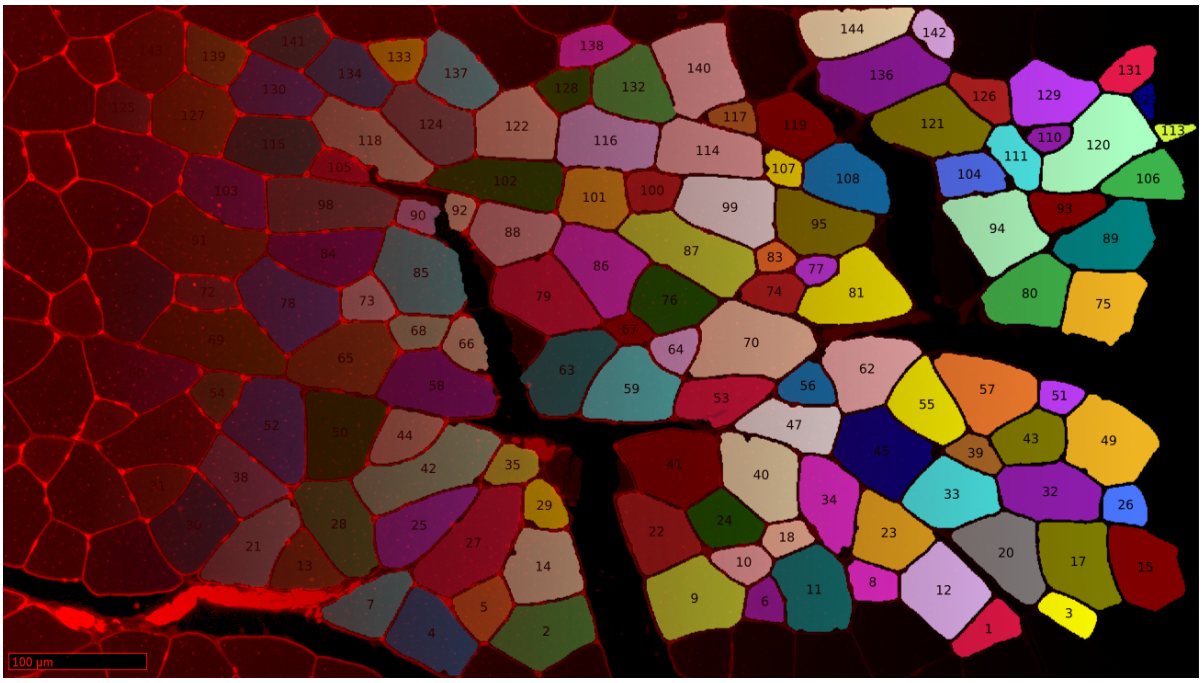

Figure 4. Example cross section after manual thresholding and the final processed image, blended into each other. Fibres touching the border are excluded from reconstruction.

3.1 Cluster Separation

The aim is to observe, how well SL, CV, and combined SL and CV perform with regards to separation of clustered fibres (reference clusters) into their respective individual fibres. To understand the individual contribution, we chose an Ilastik classifier as a representative for SL, the Mula et al. pipeline as the representative of CV, and the SLCV method as the representative of the combined workflow. The quality of the cluster separation of each of these methods is given in Table 1.

In the 27 test set pictures, 150 reference clusters of size 2, 58 clusters of size 3, 31 clusters of size 4, 8 clusters of size 5, and 15 cluster of size 6 were seen. Larger cluster sizes were also observed,

however, they were excluded from the analysis as there were less than 5 occurrences in the test set pictures. Firstly, it can be seen that as the cluster size increases, there is a decrease in the average sensitivity, that is, larger clusters are harder to separate. For small cluster sizes, such as 2 and 3, Mula et al. (only CV) has an average sensitivity of higher than 85%. Interestingly, Ilastik (only SL) has a similar average sensitivity as Mula et al.. Even though it appears that the sensitivity in Ilastik's cluster separation decays slower than the sensitivity of Mula et al.'s pipeline with growing cluster size, there was no significant difference between the two methods (see Figure 5A). Considering the SLCV (SL and CV) method, it can be seen that the average sensitivity is significantly higher than in both Mula et al. and Ilastik. Only $n = 5$ is an outlier in this respect. Due to the low sample size of 8, the bootstrapping could not detect a significant difference between SLCV and Ilastik. Furthermore, the average sensitivity of the SLCV pipeline only decreased by approximately 0.05 between cluster sizes two and six. In comparison, Mula et al. and Ilastik decreased by approximately 0.2 over the same range. This shows that the combination of both SL and CV (SLCV) is significantly better at cluster separation than either SL or CV alone, and that SLCV has a high chance of accurately separating even large fibre clusters. In contrast to incomplete cluster separation, we also observed oversegmentation in both pipelines causing separation errors (data not shown). That is, a cluster representing a true cell is sometimes erroneously separated into two or more clusters. In the watershed algorithm, oversegmentation is a known problem [10, 13, 14], while in Mula et al.'s pipeline, the cause are errors introduced by erosion.

Sensitivity Analysis

Cluster Size		2 ($N = 150$)	3 ($N = 58$)	4 ($N = 31$)	5 ($N = 8$)	6 ($N = 15$)
Mula et al.						
	Mean	0.92	0.85	0.73	0.78	0.69
	95% CI	0.88–0.96	0.78–0.92	0.60–0.83	0.60–0.95	0.49–0.88
Ilastik						
	Mean	0.93	0.86	0.84 ($)	0.80	0.76
	95% CI	0.89–0.96	0.78–0.92	0.73–0.93	0.60–1.0	0.61–0.89
SLCV						
	Mean	0.99 (∗†)	0.97 (∗†)	0.95 (∗†)	0.98 (∗)	0.93 (∗†)
	95% CI	0.97–1.0	0.93–0.99	0.90–0.99	0.93–1.0	0.82–1.0

(∗) Average sensitivity SLCV > Mula et al with *p-value* < 0.05.
(†) Average sensitivity SLCV > Ilastik with *p-value* < 0.05.
($) Average sensitivity Ilastik > Mula et al with *p-value* < 0.05.

Table 1. Mean sensitivity values and 95% confidence intervals as visualized in Figure 5A.

3.2 Areas

Now, the aim is to study if combining SL and CV results in a more accurate reconstruction of muscle fibre area than using CV only. In the analysis, only fibres which were correctly separated by all compared methods were considered in order to clearly separate the cell separation error from the area reconstruction error. Here, we only compare the Mula et al. reconstruction with the SLCV reconstruction. The Ilastik method which we included in the cluster separation analysis is omitted, since Ilastik yields less correctly separated single fibres than SLCV, but correctly separated true fibres share the same cluster shape and size as in SLCV and thus, both methods are equivalent in this comparison. It has to be noted that only correctly separated fibres are considered in the area error analysis. However, omitting non-separated cells from the analysis underestimates the consequences

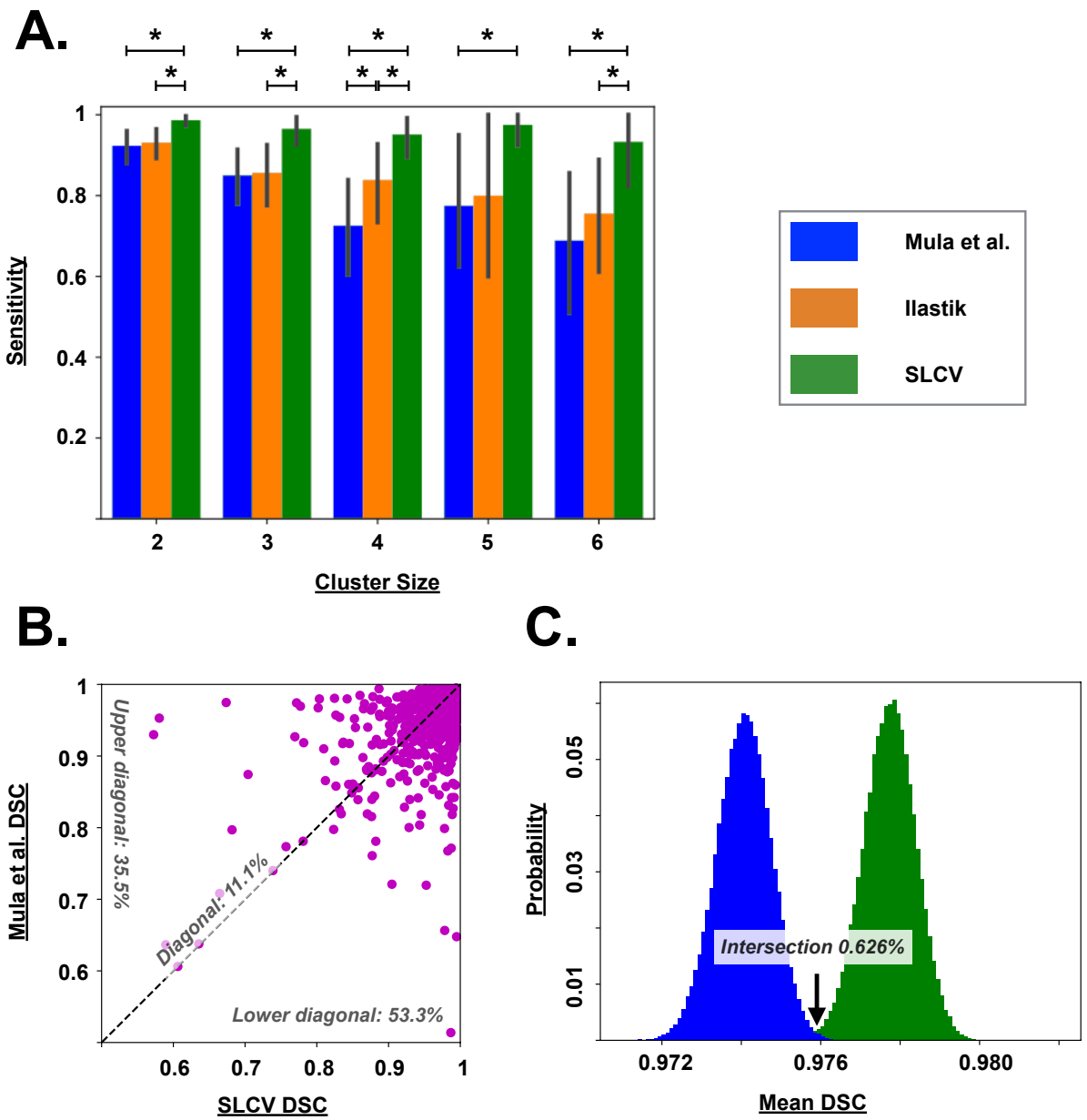

Figure 5. A: Average sensitivity values of all compared cell separation methods. Error bars are determined by bootstrapping. Raw *p-values* for significant difference are given in Table 3 in A.5, while raw sensitivity values are listed in Table 1. B: DSC of Mula et al. and SLCV reconstructions of all correctly separated samples of the test picture set. C: Mean DSC from 10^5 bootstrapping iterations on the cell reconstruction results for Mula et al. and SLCV.

that incorrect cell separation has on area reconstruction, that is, the combined effect of both errors. To address this phenomenon, an additional analysis step is presented in §A.6.

There were a total of 2603 fibres correctly separated by both Mula et al. and SLCV. For each fibre, the Dice Similarity Coefficient (DSC, §2.5) between the groundtruth and the GAC Snake-reconstructed fibre were calculated. The results are shown in Figure 5B. It was found that approximately 11.1% of the fibres had a similar DSC in both the Mula et al. and the SLCV pipeline. Furthermore, approximately 35.5% of fibres had a higher DSC if they were reconstructed by the Mula et al. pipeline. The remaining fibres (approximately 53.3%) had a higher DSC in the SLCV pipeline results.

Regarding the average reconstruction quality of fibres, the average DSC over the 2603 fibres was approximately 0.9741 for Mula et al. and 0.9778 for SLCV (see Table 2). The two averages are very close, however, when the samples are bootstrapped, it becomes clear that the average DSC of SLCV reconstructions is significantly better (see Figure 5C and Table 2). The bootstrapping procedure

draws–with replacement–a new set of fibres from the original set in each iteration and calculates the average DSC every time. A small *p-value* obtained by this procedure (as explained in §2.5) confirms that the difference between two methods is not due to chance. Hence, combining SL and CV produces a better reconstruction of a muscle fibre than a purely CV based method. Considering that Mula et al. and the SLCV method both used the same GAC Snake implementation, the gain observed in the reconstruction quality must be attributed to the size and shape of the cluster produced in the cluster separation step of the respective method.

Area Reconstruction Analysis

	Mula et al.	SLCV
Average DSC	0.9741	0.9778 (∗)
95% CI	0.9727–0.9754	0.9764–0.9790
Original Sample Size	2628	2735
Adapted Sample Size	2603	2603

(∗) Average DSC SLCV > Mula et al. with *p-value* < 0.05.

Table 2. Comparison of the area reconstruction quality of SLCV and Mula et al. Including only the reconstruction results of cells that were separated correctly by both pipelines.

4 Discussion

In this work, we introduced a semi-automatic muscle cell segmentation pipeline, which is robust against variation of image features. The usage of the random forest classifier as a supervised learning technique is critical, as it can cope with imaging variation and noise much better than CV strategies, which solely use intensity changes for border detection. A beneficial effect of this is that SLCV does not require any preprocessing of the raw images, which a computer vision only pipeline as Mula et al. does, since otherwise fibres positioned on the outside of muscle fibre bundles can be lost due to insufficient contrast. An example is shown in §A.7, Figure 9B. However, preprocessing using an intensity threshold is still recommended to prevent or reduce the issue of gaps being detected as muscle fibres.

We experimented with other supervised learning methods that could be used in the first phase to replace the random forest model. We trained a Convolutional Neural network (CNN) on multiple training images and found that no additional gains in the quality of the borders could be observed (§A.8). The pipeline is thus adaptable to different supervised learning methods, but the random forest performs best among the methods tested.

The SLCV pipeline is quickly and intuitively adaptable due to the training and parameter tuning process: Training requires no more than drawing a few lines onto a training image. Tuning of the pipeline is very user-friendly, because apart from the training process, there is only one parameter which needs to be set in order to obtain a good segmentation quality, which is the distance transformation cutoff within the watershed algorithm. In contrast, Mula et al.'s pipeline needs several parameters to be tuned in the first two phases. The GAC Snake algorithm used for both pipelines contains additional parameters, which were relatively easy to set for our test images, since the best setting with respect to the image test set was close to the parameters given in the implementation of [16]. This makes the SLCV pipeline not only adaptable to different data sets, but also to different staining methods: The classes of the segmentation classifier can be changed to represent the different typical textures in images created by the staining method. Furthermore, the usage of SLCV is not restricted to muscle segmentation problems, but can be applied to other problems, for example heart- or nerve cell segmentation or the detection of any other tubular or rounded structure in stained images.

A positive characteristic is that the cell separation quality of the pipeline can be as accurate as the user wishes. That is, little training on few pictures is sufficient to obtain a good result, but if a perfect separation is required, the pipeline can be tuned to achieve this result by more training or by splitting the picture set into multiple subsets of similar pictures, with one classifier for each subset. This splitting could be automized by characteristics like average picture intensity or intensity distribution, but it can also be done manually.

One limitation is that it is not possible to use the pipeline without human input. However, once the respective classifiers have been created for the user's different classes of imaging data and an appropriate parameter for watershed is chosen, the algorithm can automatically be applied to new data. Another limitation are the artifacts that are introduced because of the oversegmentation of the watershed algorithm. In Mula et al.'s pipeline, where the oversegmentation originates from erosion, these false segmentation results can not be removed easily due to the high level of automation of the pipeline. In SLCV, oversegmentation can be lowered or even completely avoided by adding more training lines to Ilastik or by splitting the image set as described above. However, a way to improve the algorithm in the future would be to find a strategy that can circumvent the oversegmentation problem which is inherent to the watershed algorithm (cf. [1,3,11]), and which is suitable for the type of images used in the SLCV pipeline.

We found that the size and shape of the cluster input to the Snake algorithm has an impact on the area reconstruction quality. In the fluorescence-stained images processed here, as well as in the immunohistochemically stained images with similar characteristics, this is very likely due to the noise present in the fibres. This noise can disturb the image attraction force used in the GAC Snake algorithm, such that the area can not expand to the true cell borders. Since the fibre separation strategy used in the SLCV pipeline does not involve shrinking of the cell-cluster, the resulting final clusters are bigger than in shrinkage-based separation methods such as erosion (step II in Mula et al.'s pipeline). These bigger clusters are already close to the original area and thus yield a better DSC, if the cell can not be reconstructed correctly. Another factor which contributes to the bigger size of final clusters in the SLCV pipeline is that the SL step without further CV correction already provides reasonably good cluster separation. Thus, many clusters already represent single fibres and are unchanged after the watershed algorithm. An example is shown in §A.7, Figure 9A.

With regards to cell separation quality, we showed that supervised learning (step I of the SLCV pipeline) performs as well as two CV methods combined (step I and II of two Mula et al.'s pipeline) and thus outperforms pure CV. However, when SL was combined with a CV correction step (the second step of SLCV), a significant improvement in the separation quality could be seen with regards to both CV and SL. Hence, learning alone is a powerful method, but to reach optimal performance, it has to be combined with computer vision.

With regards to area reconstruction, the combination of SL and CV also leads to a significant improvement compared to CV methods only, which is due to the more favourable characteristics of the final clusters created by SL and CV as described above, and leads to an improved robustness to noise. As a concluding remark, combining SL and CV creates a significant improvement with respect to all muscle fibre detection quality criteria used in this work and is thus a superior method to both purely SL- and CV-based methods.

References

1. P. P. Acharjya and D. Ghoshal. An approach to reduce oversegmentation in watershed ridge line observation. *International Journal of Advancements in Research & Technology*, 2(6), 2013.
2. L. Álvarez, L. Baumela, P. Henríquez, and P. Márquez-Neila. Morphological snakes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

3. W. Bieniecki. Oversegmentation avoidance in watershed-based algorithms for color images. In *Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2004. Proceedings of the International Conference*, pages 169–172, 2004.

4. G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, jun 1986.

5. J. Chanussot and P. Lambert. Watershed Approaches for Color Image Segmentation. In *NSIP'99*, pages 129–133, 1999.

6. L. R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, jul 1945.

7. B. Emde, A. Heinen, A. Gödecke, and K. Bottermann. Wheat germ agglutinin staining as a suitable method for detection and quantification of fibrosis in cardiac tissue after myocardial infarction. *European Journal of Histochemistry*, 58(4), dec 2014.

8. A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *Medical Image Analysis*, volume 9, pages 130–137. jun 1998.

9. Q. Q. Gao and E. M. McNally. The Dystrophin Complex: Structure, Function, and Implications for Therapy. In *Comprehensive Physiology*, pages 1223–1239. John Wiley & Sons, Inc., Hoboken, NJ, USA, jun 2015.

10. J. Gauch. Image segmentation and analysis via multiscale gradient watershed hierarchies. *IEEE Transactions on Image Processing*, 8(1):69–79, 1999.

11. M. A. Gonzalez, G. J. Meschino, and V. L. Ballarin. Solving the over segmentation problem in applications of Watershed Transform. *Journal of Biomedical Graphics and Computing*, 3(3), apr 2013.

12. S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, mar 1951.

13. P. Li and X. Xiao. An unsupervised marker image generation method for watershed segmentation of multispectral imagery. *Geosciences Journal*, 8(3):325–331, sep 2004.

14. F. Liu, A. Mackey, R. Srikuea, K. Esser, and L. Yang. Automated image segmentation of haematoxylin and eosin stained skeletal muscle cross-sections. *Journal of Microscopy*, 252(3):275–285, dec 2013.

15. D. J. C. MacKay. *Information Theory and Learning Algorithms*. Cambridge University Press, fourth pri edition, 2003.

16. P. Marquez-Neila, L. Baumela, and L. Alvarez. A Morphological Approach to Curvature-Based Evolution of Curves and Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):2–17, jan 2014.

17. J. Mula, J. D. Lee, F. Liu, L. Yang, and C. A. Peterson. Automated image analysis of skeletal muscle fiber cross-sectional area. *Journal of Applied Physiology*, 114(1):148–155, jan 2013.

18. V. M. Niemeijer, T. Snijders, L. B. Verdijk, J. van Kranenburg, B. B. L. Groen, A. M. Holwerda, R. F. Spee, P. F. F. Wijn, L. J. C. van Loon, and H. M. C. Kemps. Skeletal muscle fiber characteristics in patients with chronic heart failure: impact of disease severity and relation with muscle oxygenation during exercise. *Journal of Applied Physiology*, 125(4):1266–1276, oct 2018.

19. N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, jan 1979.

20. F. Papadopulos, M. Spinelli, S. Valente, L. Foroni, C. Orrico, F. Alviano, and G. Pasquinelli. Common Tasks in Microscopic and Ultrastructural Image Analysis Using ImageJ. *Ultrastructural Pathology*, 31(6):401–407, jan 2007.

21. M. Salanova, C. Gelfi, M. Moriggi, M. Vasso, A. Viganò, L. Minafra, G. Bonifacio, G. Schiffl, M. Gutsmann, D. Felsenberg, P. Cerretelli, and D. Blottner. Disuse deterioration of human skeletal muscle challenged by resistive exercise superimposed with vibration: evidence from structural and proteomic analysis. *The FASEB Journal*, 28(11):4748–4763, nov 2014.

22. L. R. Smith and E. R. Barton. SMASH – semi-automatic muscle analysis using segmentation of histology: a MATLAB application. *Skeletal Muscle*, 4(1):21, 2014.

23. C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 230–233. IEEE, mar 2011.

24. C. N. Wurtzel, J. P. Gumucio, J. A. Grekin, R. K. Khouri, A. J. Russell, A. Bedi, and C. L. Mendias. Pharmacological inhibition of myostatin protects against skeletal muscle atrophy and weakness after anterior cruciate ligament tear. *Journal of Orthopaedic Research*, 35(11):2499–2505, nov 2017.

25. C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, mar 1998.

26. C. Xu and J. L. Prince. Gradient vector flow: a new external force for snakes. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 66–71, 1997.

A Appendix

A.1 Recommendations for the training of the Ilastik classifier

We recommend the training of a segmentation classifier on two cross sectional pictures, which taken together contain the lower and upper end of the range of border intensity, cell plasma intensity, cell plasma structure, noise amount and noise distribution of the pictures that the classifier will be applied to. However, if the range of intensities is very big, the classifier will lose accuracy and two classifiers should be trained on the lower and upper part of the intensity range, respectively. Training pixels for "border" should be drawn in with a relatively small brush, and a border should be drawn in one continuous stroke. It is especially recommended to add borders which contain areas of visibly lower staining intensity to the training pixels, see Figure 1C. Additionally, borders which are not continuous due to torn tissue or staining artifacts like in Figure 1E should imperatively be added to the training set, drawn as a continuous object. In other words, borders which are flawed in the raw picture should be drawn in the way they would optimally look like. The training dialogue features a live update, which can be used to check if holes remain in the training picture borders. If so, these borders should be added to the training set. Big fibre and small fibre training should include cells with different structure (different intensity and noise distribution). The whole cell is recommended to be covered with training pixels, and border pixels have to be excluded. The live update should be used to check if border pixels are found within cell or gap regions. If so, it is recommended to add a few more cell training pixels, drawing one fibre at a time. Training on gap regions can be sparse, but should include intensity changes and artifacts that may be found in the gap regions, such as very bright objects or background noise. In general, artifacts should be labeled as the object that they are supposed to represent. When the classifier is applied to the training images and the result contains little to no holes in the borders, and no border is predicted in gap or fibre regions, the classifier is fit to be applied to test set images. If this state is not reached or can only be reached by drawing a lot of training lines, the training images might be too different from each other and at least one image should be replaced.

A.2 Training set of Ilastik Classifier 2

The full training set for classifier 2 is given in Figure 6. Fig. 6B shows a section that was cut out from a cross sectional image, since no training lines were added to the rest of the image. The size of the section remains the same as in the original image.

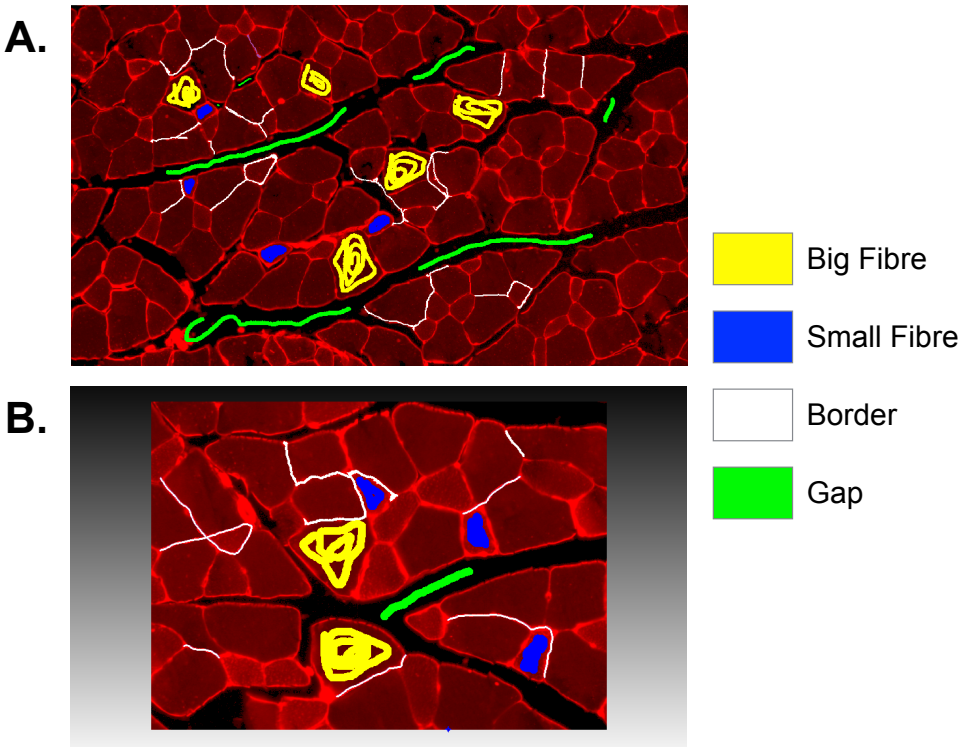

Figure 6. Training set of Ilastik classifier 2 out of 2, which was applied to 2 out of 27 of the test set images which were of higher brightness and contained lower quality borders than the other test set images. Yellow: class "big fibre", blue: class "small fibre", white: class "border", green: class "gap". A: First training image. B: Second training image, which is a section from a full image.

A.3 Additional parameter choices in the implementation

Ridge Detection (Step I in Mula et al.) In contrast to the description in Mula et al., we performed single-scale ridge detection, where the scale parameter is denoted by σ^*. We chose $\sigma^* = 0.7$. We first constructed the Hessian matrix H:

$$H(x, y, \sigma^*) = \begin{bmatrix} \frac{\partial^2 I_{\sigma^*}(x,y)}{\partial x^2} & \frac{\partial^2 I_{\sigma^*}(x,y)}{\partial x \partial y} \\ \frac{\partial^2 I_{\sigma^*}(x,y)}{\partial x \partial y} & \frac{\partial^2 I_{\sigma^*}(x,y)}{\partial y^2} \end{bmatrix}, \tag{3}$$

where $I_{\sigma^*}(x, y) = I(x, y) \otimes G(x, y, \sigma^*)$ and $\otimes$ represents the convolution operator. $G(x, y, \sigma^*)$ is the two-dimensional Gaussian kernel:

$$G(x, y, \sigma^*) = \frac{1}{2\pi(\sigma^*)^2} \, e^{-\frac{x^2+y^2}{2(\sigma^*)^2}} \tag{4}$$

Then, the two eigenvalues λ_1, λ_2 of H with $|\lambda_1| > |\lambda_2|$ were computed and each ridge obtained a likelihood measure:

$$r_{\sigma^*} = \begin{cases} 0 & \text{if } \lambda_1 > 0 \\ e^{-\frac{R_B}{\alpha^2}} \left(1 - e^{\frac{-S^2}{\beta^2}}\right) & \text{else} \end{cases}$$

where $R_B = |\frac{\lambda_2}{\lambda_1}|$, $S = \lambda_1^2 + \lambda_2^2$ and we chose $\alpha = 0.5$ following a recommendation from [8] and $\beta = 0.03$.

Erosion (Step II in Mula et al.) Ridges were submitted to a closing operation, where we chose the kernel to be of rectangular shape with size 11×11 pixel (px). After that, Mula et al. state that the inverse of the resulting edge map was used to detect the clusters. We implemented this as a dilation step carried out in two iterations with a rectangular 3×3 px kernel followed by the inversion of the edge map and a contour detection algorithm. Each detected cluster was then iteratively eroded with an elliptical kernel as specified in Mula et al., of size 8×8 px, until it was smaller than a certain threshold. We used the exemplary threshold of 5000 px given in Mula et al. Additionally to the procedure described in the paper, we filtered out clusters which were smaller than 500 px to remove artifacts.

GAC Snake (Step III in SLCV and Mula et al.) Additional parameters for the Snake algorithm are ν, μ and θ. ν determines the strength of the balloon term and is used as $\nu = -1$ for a deflating balloon and as $\nu = 1$ for an inflating balloon in the implementation of the GAC Snake by Marquez-Neile et al. [16]. We used $\nu = 1$. μ controls the number of repetitions of the smoothing step in every iteration of Snake. We chose $\mu = 3$ to obtain a smooth curve. θ is related to the discretization of $g(I)$ and used to control the smoothing operation strength at different points of the curve. We set $\theta = 0.3$ analogous to the parameter setting in all test cases given in the implementation by Marquez-Neila et al.. For the algorithm to work correctly, the final cluster has to be completely enclosed within the boundaries of the original cell, since we chose the balloon force to have an inflating effect. Furthermore, the bigger the cluster is, the more robust the area detection is for muscle cells which contain high concentrations of the staining protein. As mentioned in the Discussion, this form of noise, especially close to the muscle bundle border, disturbs the image attraction term and leads to an early stop of the snake contour and thus to an incomplete area reconstruction (see Figure 9A).

A.4 Pre/ Postprocessing

Muscle fibre bundles inside a picture are usually separated from each other by an interstitial area which does not contain stained cells and which we refer to as gap region. In many pictures, this area contains a certain concentration of the stained protein. The more protein the region contains, the brighter it appears in the picture and the lower the contrast to the staining intensity of muscle cells. Consequently, it is harder to properly detect the fibre bundle boundaries. While the training step in our supervised learning strategy can be adapted to correct for the latter phenomenon, it is problematic for computer vision (CV) based pipelines such as that of Mula et al. and often results in gaps in the muscle fibre bundle boundary, such that outer muscle cells of the bundle may not be identified correctly (see Figure 9B). Furthermore, staining noise in gap regions makes it impossible to automatically distinguish gaps from muscle cells, such that spurious muscle cells detected in gap regions would have to be removed in a post-processing step. Hence, a suggested strategy is to manually choose an intensity threshold $\tau \in [0, 255]$ (for 8-bit pictures) with

$$I(x,y) := \begin{cases} \hat{I}(x,y) & \text{if } \hat{I}(x,y) \geq \tau \\ 0 & \text{else} \end{cases}$$

and to exclude regions of value zero from the cell detection.

A.5 Bootstrapping of cluster separation

Table 3 contains the *p-values* of sensitivity differences between the Ilastik random forest model, the SLCV pipeline and Mula et al.'s pipeline (shown in Figure 5A in §3.1). These *p-values* were obtained by bootstrapping, as described in section 2.4. A *p-value* smaller than 0.05 means that the sensitivity of pl_1 (left term) is significantly higher than the sensitivity of pl_2 (right term).
Figure 7 shows the bootstrapping distribution over 10^5 resamples with respect to cluster separation (similar to Figure 5 C in section 3.2, which show the same distributions with respect to DSC). The

n	*p-value* SLCV vs. Mula	*p-value* SLCV vs. Ilastik	*p-value* Ilastik vs. Mula
2	1.5×10^{-4}	7×10^{-5}	0.383
3	5×10^{-5}	7×10^{-5}	0.466
4	$\ll 0.05$	9.2×10^{-4}	0.0152
5	0.021	0.058	0.462
6	$\ll 0.05$	8×10^{-5}	0.275

Table 3. Raw *p-values* of the cell separation sensitivity differences in Figure 5 A, section 3.1. The *p-values* are obtained by bootstrapping, H_0: "The average sensitivity of pipeline pl_2 (right term) is $\geq$ the average sensitivity of pipeline pl_1 (left term)".

Figures contain the bootstrapping results of the three pipelines with respect to cluster separation, and they are included to visualize the small *p-values* given in Table 3 above.

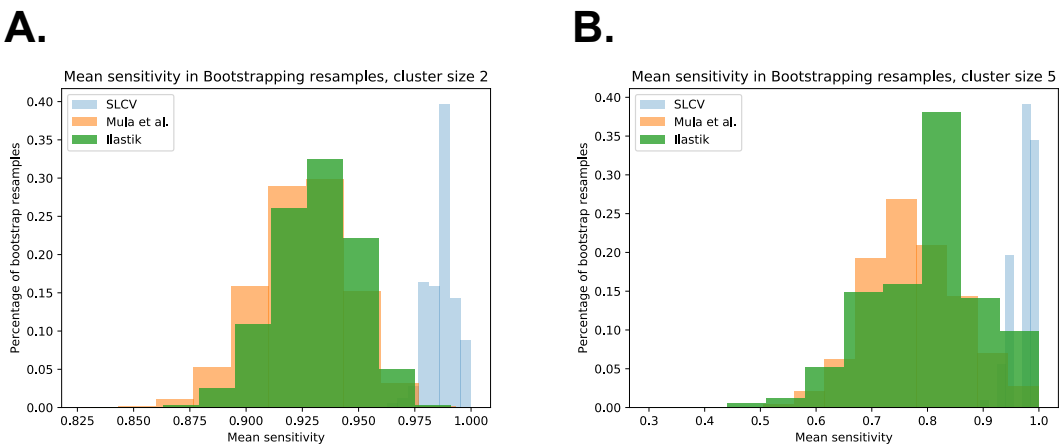

Figure 7. A: Bootstrap distribution of average sensitivity values on cluster size 2. B: Bootstrap distribution of average sensitivity values on cluster size 5.

A.6 Analysis of the combined cluster separation-area reconstruction error

The Kullback-Leibler (KL) divergence, first introduced by Kullback et al., is a measure for the distance between two probability distributions [12]. If the distributions $P(x)$ and $Q(x)$ are discrete and they are both defined on the same space, the KL divergence is defined as follows [15]:

$$D_{KL}(P||Q) = \sum_x P(x)\ log\frac{P(x)}{Q(x)}. \tag{5}$$

The distribution of groundtruth cell areas $P(x)$ was compared to the distributions of areas $Q(x)$ reconstructed by SLCV and Mula et al.'s pipeline, respectively. The histograms of the distributions were calculated using interval bin sizes $h_b = \{50, 100, 200, 500, 1 \times 10^3, 2 \times 10^3, 5 \times 10^3, 1 \times 10^4, 2 \times 10^4\}$.

Figure 8A shows the mean KL divergence and the standard deviation calculated over all test set images. The KL divergence shrinks exponentially with growing bin interval size, but the SLCV reconstruction shows a consistently smaller deviation from the groundtruth cell area than Mula et al.'s reconstruction, while the standard deviations are similar. The difference between the SLCV deviation- and the Mula et al. deviation from the groundtruth grows with growing bin interval size. Figure 8B shows the same calculation, only for the cell area distribution of all images taken together. The KL divergence is smaller than in Figure 8A, where the mean value over the single images was considered. However, the shrinking behaviour with growing bin interval size is similar in both Figures, while the difference between the curves fluctuates more in Fig. 8 B. In conclusion, SLCV consistently reconstructs cell areas better than Mula et al. for all histogram bin intervals considered and in both single images and the overall test set; as it is always closer to the groundtruth cell area distribution.

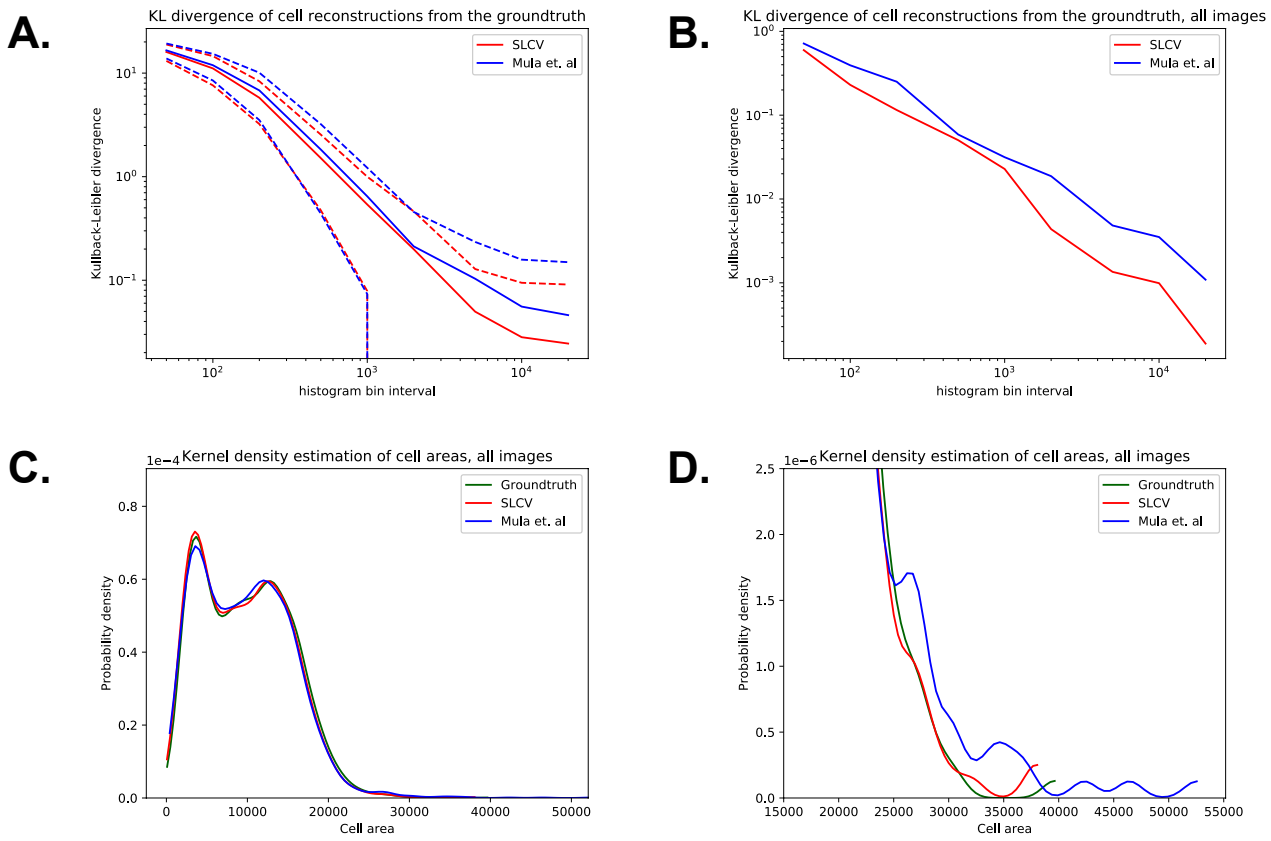

Figure 8. A: Kullback-Leibler divergence of SLCV and Mula et al. reconstructed cell area distributions from groundtruth cell area distribution. Solid line represents the mean KL divergence over all images, dotted lines represent the standard deviation. Both axes are plotted logarithmically. B: Kullback-Leibler divergence of SLCV and Mula et al. reconstructed cell area distributions from groundtruth cell area distribution over all images. C: Kernel density estimation of cell areas over all images. D: Zoomed in tail region from C.

Figures 8C and 8D show a kernel density estimation of the cell area distributions over all test set images. A kernel of bandwidth $bw_k = 0.2$ was used to estimate the densities of the histograms, which were constructed with bin interval $h_b = 800$. As seen in Figure 8C, both SLCV and Mula et al. are close to the original cell area distribution, while SLCV shows less deviation from the original distribution in the peaks. In Figure 8D, the tail of the distributions is zoomed in. Again, SLCV is closer to the groundtruth, while Mula et al. shows a longer and more pronounced tail, resulting from the bigger cell separation error.
If individual test set images are considered, the deviation from the groundtruth distribution of SLCV and Mula et al. varies more (data not shown): Mula et al.'s distribution shows a shift to the left in a few images, which represents a tendency to incomplete area reconstruction in these images. In most cross sections, both reconstructed area distributions have a tail on the right side, while the tail of Mula et al.'s curve is more pronounced. Overall, SLCV is mostly close to the original distribution, while Mula et al.'s area distribution tends to deviate from the original shape more due to the longer tail.
In summary, a higher cell separation error leads to a bigger area reconstruction error, hence a low cell separation error is a necessary condition for an automated cell separation pipeline to create cell reconstructions of high quality and to yield good CSA estimates.

A.7 Examples for challenges in muscle fibre detection

Two representative examples for challenges we observed in automated muscle fibre detection are shown below in Figure 9. Row A shows incompletely reconstructed cells, which appear in cases where the

image attraction force used in the GAC Snake algorithm is disturbed by large amounts of noise. The final clusters created by the SLCV pipeline are bigger and are thus not affected as much as the seeds created by Mula et al.'s pipeline. Row B shows an example of the effect that a noisy gap region can have on cell separation. The border between fibre and gap is not bright enough to be fully detected, such that the cell is eventually lost in the purely CV based pipeline. The SL step of SLCV was able to classify the border, because its training included noisy gap regions. In most images, this phenomenon can be avoided by preprocessing with a thresholding step.

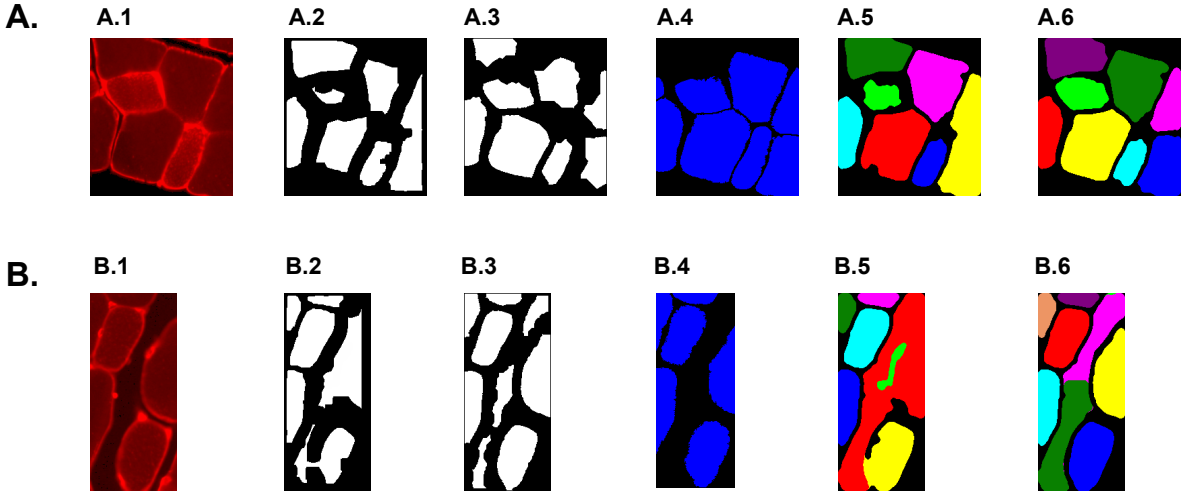

Figure 9. Examples of challenging cell reconstructions. 1: thresholded original picture, 2: final clusters produced by Mula et al.'s pipeline, 3: final clusters produced by SLCV pipeline, 4: groundtruth, 5: Mula et al.'s segmentation result, 6: SLCV segmentation result. A: missing cell area in reconstruction of noisy cells due to small seeds, B: merging of cells and gap regions due to low contrast.

A.8 Alternatives for Step I of the SLCV Pipeline

Alternative supervised learning techniques can also be used for the detection of the initial borders. We trained a Convolutional Neural Network (CNN) on multiple images and could not detect any quality gains in the detected borders. An example of the output of the CNN and that of the Ilastik Random Forest classifier can be seen in Figure 10.

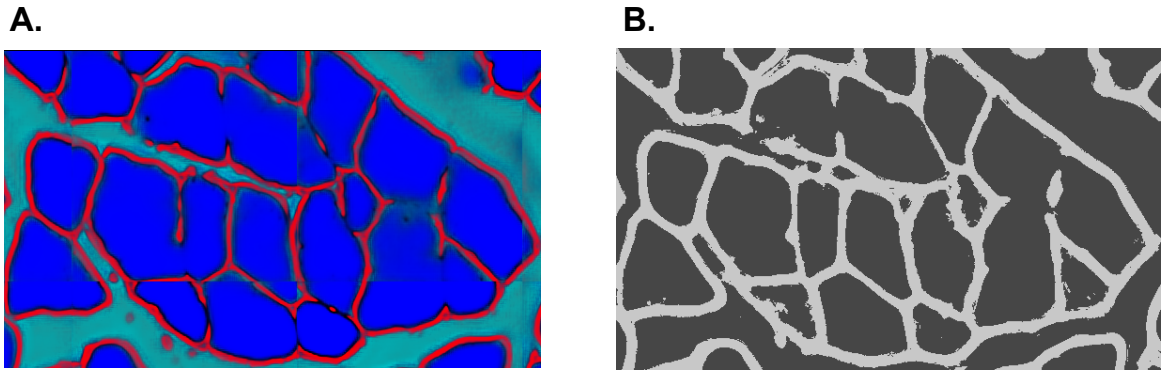

Figure 10. Comparison of supervised learning methods to create the initial borders. A: Output of the CNN. Red: borders, green: gap, blue: muscle fibre. B: Output of Ilastik. Only class "border" is shown in light gray.