Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Radioelectronics

Automatic segmentation of cell structures from electron microscope images

Automatická segmentace buněčných struktur ze snímků elektronového mikroskopu

Bc. Markéta Kvašová

Supervisor: Ing. Karel Fliegel, Ph.D.
Co-supervisor: Ing. Jakub Pospíšil, Ph.D.
Study program: Electronics and Communications
Specialisation: Audiovisual Technology and Signal Processing
May 2024

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kvašová Markéta** Personal ID number: **483774**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Radioelectronics**

Study program: **Electronics and Communications**

Specialisation: **Audiovisual and Signal Processing**

II. Master’s thesis details

Master’s thesis title in English:

Automatic Segmentation of Cell Structures from Electron Microscope Images

Master’s thesis title in Czech:

Automatická segmentace buněčných struktur ze snímků elektronového mikroskopu

Guidelines:

The work will be related to the particular type of liver cell LSEC (liver sinusoidal endothelial cell) and its essential filtering function. The segmentation of LSEC cells and their fenestrations have been semi-automatic, which is very time-consuming. Provide a state-of-the-art overview of this field. Implement tools for substituting the manual segmentation of cell structures from an electron microscope (EM) image into a fully automatic one. Verify the performance of the developed tools on the available image datasets.

Bibliography / sources:

[1] Szafranska, K. et al. Quantitative analysis methods for studying fenestrations in liver sinusoidal endothelial cells, 2021.
[2] Ouyang, W. et al. BioImage Model Zoo: A Community-Driven Resource for Accessible Deep Learning in BioImage Analysis, 2022.
[3] BioImage Model Zoo, Advanced AI models in one-click (https://bioimage.io/).

Name and workplace of master’s thesis supervisor:

Ing. Karel Fliegel, Ph.D. Department of Radioelectronics FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **03.02.2024** Deadline for master's thesis submission: ___________

Assignment valid until: **21.09.2025**

Ing. Karel Fliegel, Ph.D. Supervisor’s signature	doc. Ing. Stanislav Vítek, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Acknowledgements

I would like to thank Ing. Karel Fliegel, Ph.D., for supervising this thesis. I am also grateful to Ing. Jakub Pospíšil, Ph.D., thesis co-supervisor, for his help with the ground truth mask editing and for sharing his knowledge relating to the thesis topic. Additionally, I would like to thank Bartłomiej Zapotoczny, Ph.D and Karolina Szafranska, Ph.D. for sharing their image data with us. Without this data, the training and evaluation would have been done on a much smaller scale.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 23. May 2024

Abstract

This thesis presents a method of segmenting electron microscope images of *in vitro* Liver Sinusoidal Endothelial Cells (LSECs). The segmented objects are LSEC fenestrations, which cover the cells' bodies. Segmentation masks of LSEC fenestrations are commonly used in research on the influences on fenestration properties. Neural networks combined with classical image processing techniques are used in the solution of this task. In this thesis, 14 images were used for training, and 22 images were used for evaluation. The image modality is Scanning electron microscopy (SEM), and the individual image sizes are 6 144 × 4 608 pixels, which corresponds to 57 × 43 µm. The proposed method was compared to a semiautomatic method, which is widely used for LSEC segmentation. The performance of these two methods was compared using masks, which we consider to be the ground truth. A pixel-based evaluation was conducted by computing the Dice score metric. The mean Dice score of the 22 test masks is 0.92 ± 0.06 for the semiautomatic and 0.94 ± 0.02 for the automatic method. Statistical evaluation of the segmented fenestrations was also performed. The ground truth fenestration properties were compared to those found automatically and semiautomatically by computing the correlation of the main fenestration properties in each image. The advantages and disadvantages of both methods are discussed.

Keywords: binary image segmentation, scanning electron microscopy, liver sinusoidal endothelial cells

Supervisor: Ing. Karel Fliegel, Ph.D.

Abstrakt

Tato práce představuje metodu segmentace snímků *in vitro* jaterních sinusoidálních endoteliálních buněk (LSECs) pořízených elektronovým mikroskopem. Segmentované objekty jsou fenestrace, které pokrývají LSEC buňky. Segmentační masky fenestrací LSEC buněk se běžně používají ke zkoumání vlivů na vlastnosti fenestrací. Při řešení této úlohy byly použity neuronové sítě v kombinaci s klasickými metodami zpracování obrazu. V této práci bylo použito 14 snímků pro trénování a 22 snímků pro vyhodnocení. Obrazová modalita je skenovací elektronová mikroskopie (SEM) a jednotlivé snímky mají velikost 6 144 × 4 608 pixelů, což odpovídá 57 × 43 µm. Navržená metoda byla porovnána s poloautomatickou metodou, která se běžně používá pro segmentaci LSEC buněk. Srovnání těchto dvou metod bylo provedeno pomocí masek, které považujeme za ground truth. Hodnocení kvality automatických a semiautomatických masek na základě hodnot pixelů bylo provedeno výpočtem metriky Dice score. Průměrná Dice score hodnota pro 22 testovacích masek je 0,92 ± 0,06 pro poloautomatickou a 0,94 ± 0,02 pro automatickou metodu. Bylo rovněž provedeno statistické vyhodnocení segmentovaných fenestrací. Fenestrace v ground truth obrázcích byly srovnány s fenestracemi získanými oběma metodami. Byla vypočtena korelace nejdůležitějších vlastností fenestrací v každém snímku mezi ground truth a oběma metodami. Výhody i nevýhody obou metod jsou diskutovány.

Klíčová slova: binární segmentace obrazu, skenovací elektronová mikroskopie, jaterní sinusoidální endoteliální buňky

Contents

Figures

Tables

Chapter 1

Introduction

Electron microscopy has the capacity to capture small objects that cannot be resolved with an optical microscope. There are many areas where electron microscopy can be used, one of which is the study of cell morphology [1].

One type of cell with distinctive structures is the liver sinusoidal endothelial cell (LSEC). LSECs line the smallest blood vessels in the liver. Their whole cell body is dotted with small pores called fenestrations between 50 and 350 nm in size. These fenestrations ensure bidirectional passive transport of lipoproteins, drugs, and solutes between blood and hepatocytes [2]. The properties of fenestrations, such as their size and quantity, determine how well these cells fulfill their function [3].

A decreased number of fenestration in LSECs is linked to various diseases and health problems [4]. This makes the influences on LSEC health an interesting topic to study.

Detecting LSEC fenestrations can be time-consuming when done manually or semiautomatically, depending on the number of studied images and individual image properties. A fully automatic method for detecting these structures could decrease the time it takes to examine a large amount of LSEC images, which is needed for the research of diverse influences on fenestration properties.

1.1 Task definition

This thesis aims to create a method of segmenting scanning electron microscopy images of *in vitro* LSECs. The segmented objects are LSEC fenestrations, which cover the cell's bodies. Part of the solution should be the computation of relevant fenestration properties from the segmentation masks. Two key properties are fenestration frequency, which is the total number of fenestrations per cell area, and LSEC porosity, measured as the total fenestration area per cell area. Other properties such as fenestrations roundness and equivalent circle diameter can also be determined.

The performance of the proposed method should be quantified by the computation of segmentation quality metrics and statistical evaluation of fenestration properties.

1.2 Thesis structure

- In Chapter 2, the function and properties of LSECs and their fenestrations are described. Technical background about electron microscopy is given, mainly focusing on scanning electron microscopy.
- In Chapter 3, a literature review concerning the thesis topic is given. Previous studies on automatic LSEC segmentation methods are described. The history and properties of Neural networks are also discussed.
- In Chapter 4, the proposed solution is described. The description includes data preprocessing, and the training and inference process. This chapter also includes a description of the evaluation steps.
- In Chapter 5, results of the methods' comparison are presented.
- In Chapter 6, the presented results are discussed.

Chapter 2
Background

This chapter contains a description of liver sinusoidal endothelial cells, a technical background relating to electron microscopy, and information about the scanning electron microscope used to produce the images used in this thesis.

2.1 Liver sinusoidal endothelial cells (LSECs)

Liver sinusoidal endothelial cells (LSECs) line the blood vessels of the liver, and they ensure bidirectional passive transport of nanoparticles between hepatocytes and the blood flowing through the liver. This passive transport is done through fenestrations, small pores in the membranes of the cells. Examples of particles that pass through LSEC's fenestrations from blood are lipids, proteins, and viruses [2].

The properties of LSEC fenestrations, like their total number and surface area, can change with time. The main influences on these changes are age and compounds like ethanol or nicotine. These influences decrease the total number of fenestrations, reducing their filtration ability. This change is connected to diseases like liver cirrhosis, diabetes mellitus, and atherosclerosis [5, 3].

Some *in vitro* studies show the possibility of increasing the number of fenestrations. Treatments that have shown a positive effect include caffeine and theobromine [6].

LSEC fenestrations are usually arranged into groups called sieve plates (see Figure 2.1) [1]. The size of individual fenestrations is in the range of 50 to 350 nm, which is below the resolution limit of conventional light microscopy [7].

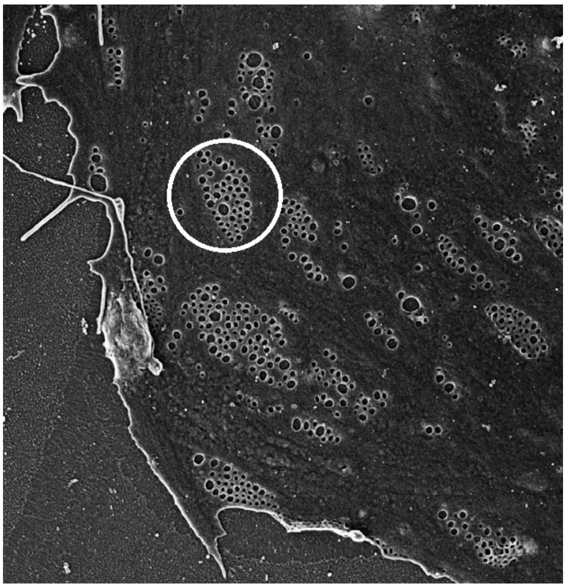


Figure 2.1: Part of an LSEC (*in vitro*) captured with a scanning electron microscope. Sieve plates, which are formed by fenestration clusters, can be seen on the cell surface. One sieve plate is marked by a white circle. Courtesy of the Arctic University of Norway.

2.2 Electron microscopy

The optical resolution of the human eye is approximately 0.1 mm. This resolution can be enhanced by using an optical lens, which is the approach of optical microscopy. The resolution of an optical microscope is limited by the wavelength of the light that is used to create the resulting image. Optical microscopy, which uses light from the visible spectrum, has a resolution limit of ~200 nm [8].

Electron microscopy substitutes the light source with an electron beam and switches the lens for electromagnetic coils. Substituting visible light with electrons of a shorter wavelength results in better resolution. Current electron microscope technology can achieve resolution even below 1 nm [9]. There are two main types of electron microscopes: transmission electron microscopes (TEM) and scanning electron microscopes (SEM) [8]. Since segmenting SEM images is the topic of this thesis, the following section focuses on SEM.

2.2.1 Scanning electron microscopy (SEM).

Scanning electron microscopy (see Figure 2.2) utilizes a focused electron beam to scan across the surface of the specimen, producing large numbers of signals. The signals used to form an image in SEM are backscattered electrons (BSEs) and secondary electrons (SEs). Additional signals produced when an electron beam strikes a sample are characteristic X-rays, Auger electrons, and

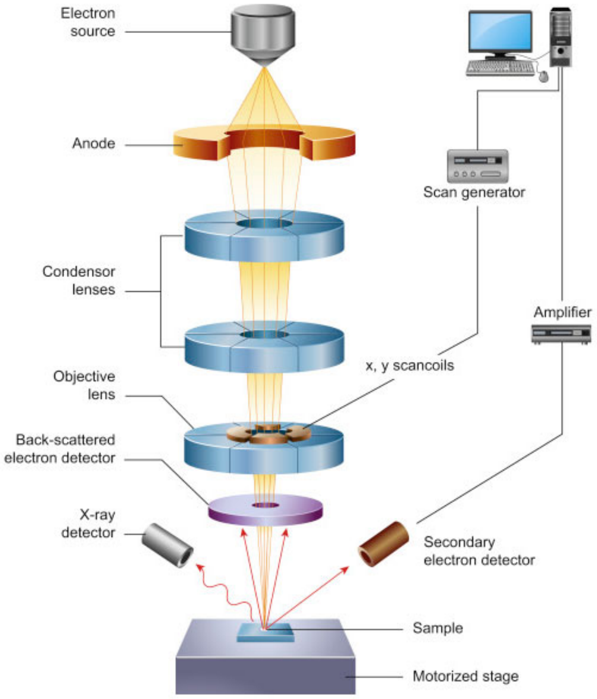


Figure 2.2: Schematic of a scanning electron microscope. Electrons are produced by the electron source and accelerated towards the anode, creating the electron beam. The electron beam is collimated into a relatively parallel stream by the condensor lenses. Objective lenses focus the electron beam into a probe point, which scans the specimen surface. Generated signals are detected and processed [8]. Image taken from [10].

cathodoluminescence [8].

Backscattered electrons are beam electrons that are elastically scattered through an angle of more than 90°. The scattering results from the deflection of the beam electron by the specimen atomic nucleus or by outer shell electrons of similar energy (see Figure 2.3) [8].

Interactions between the electrons that form the beam and the electrons of the sample result in energy transfer to the sample atoms. Secondary electrons are generated as a result of the excitation of the specimen electrons during the ionization of specimen atoms (see Figure 2.3).

Scintillator detectors, which convert electron energy into photons, are used to form images from BSEs and SEs. These photons are then moved to a photomultiplier, whose output voltage is amplified and forms the output image. BSEs have much greater energy than SEs, which makes detecting these two electron groups on separate detectors possible by applying different electric charges to the detectors [8].

BSE's higher energy prevents electrons from being absorbed by the sample, where they can undergo multiple scattering events. This results in worse resolution of BSE images than SE, but using BSEs can be beneficial because

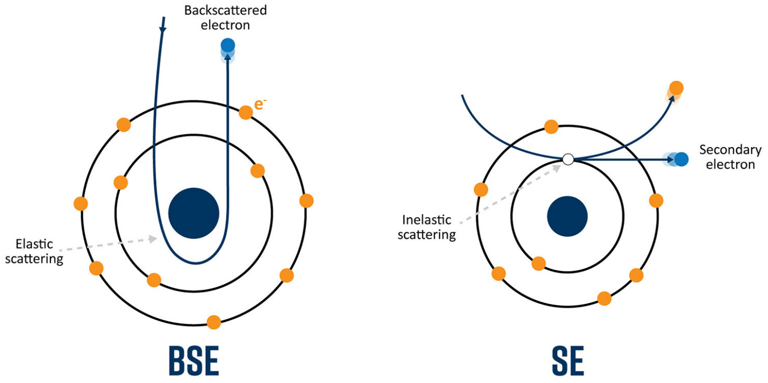


Figure 2.3: A backscattered electron (left) and a secondary electron (right). Taken from [10].

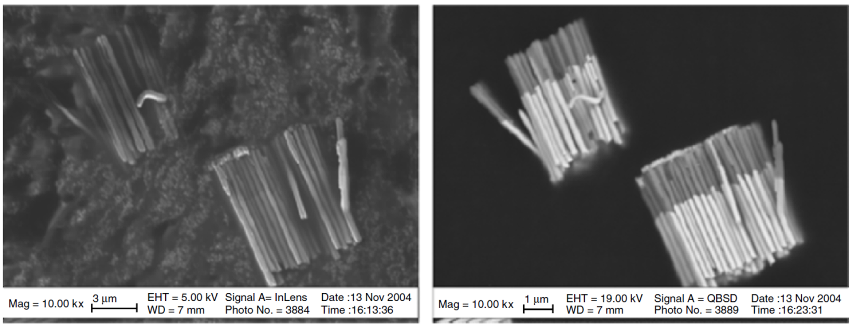


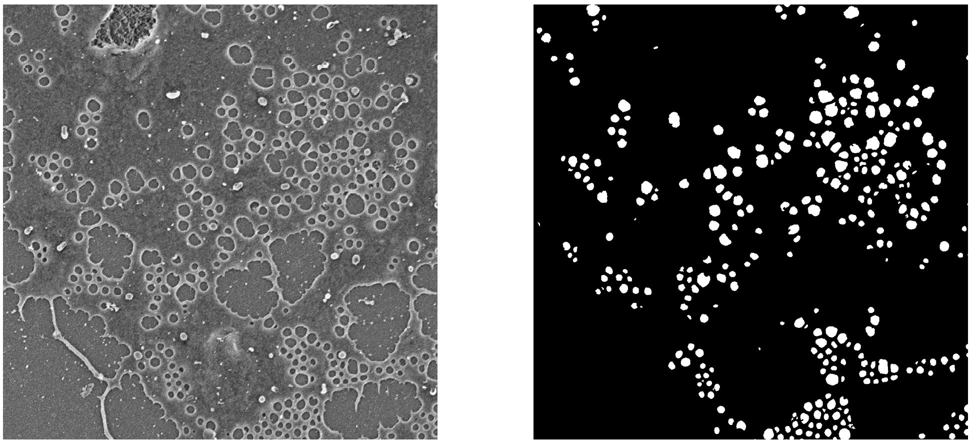
Figure 2.4: Images formed by secondary electron signal (left) and backscattered electron signal (right). Taken from [8].

they carry information about features that are deeper beneath the specimen surface [8]. The difference between images created from BSE signal and SE signal is shown in Figure 2.4.

2.3 Available data

LSEC images, which were used to develop and evaluate the proposed method, were provided to us by Bartłomiej Zapotoczny[1]. This data has been used for quantitative analysis in the research of changes in LSEC morphology [7]. The data, which was given to us, include SEM images of LSECs studied *in vitro* and their corresponding binary segmentation masks. Two classes are present in the binary masks, source image pixels showing fenestrations are assigned white in the masks, and everything else in the source image is assigned black (See Figure 2.5).

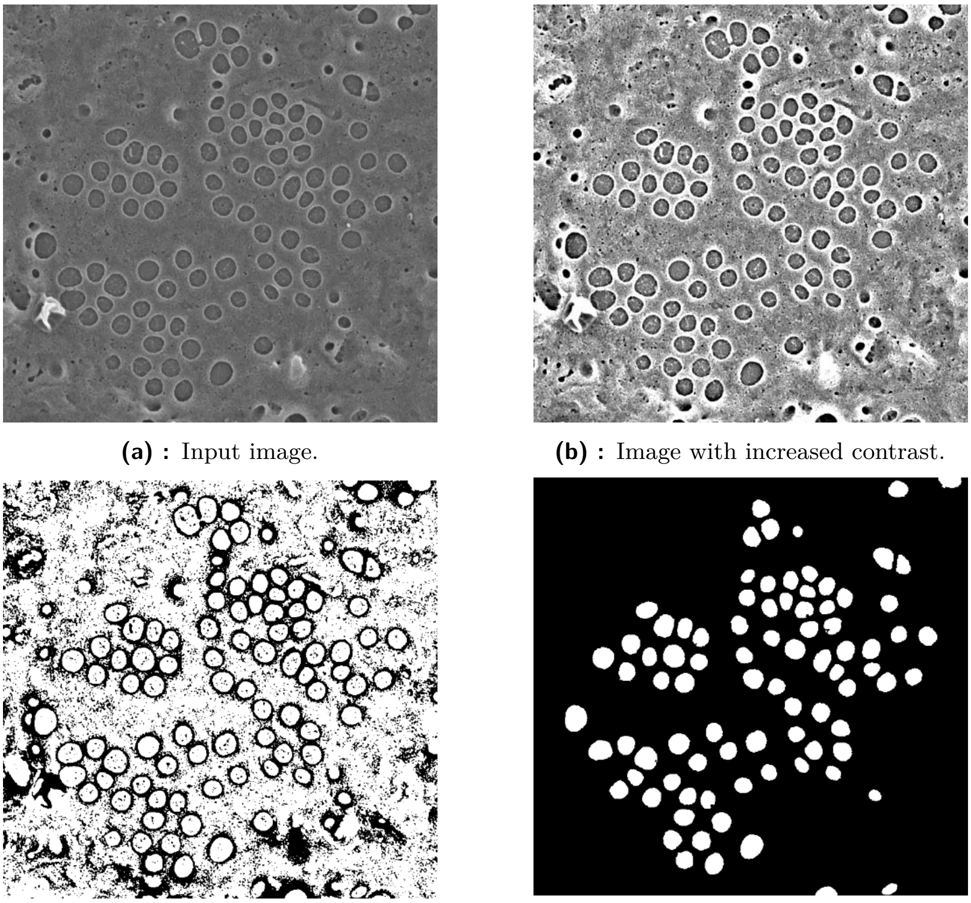
[1] https://www.ifj.edu.pl/dept/no5/nz55/department/team



(a) : Example image patch

(b) : Corresponding mask

Figure 2.5: An example image patch of 1 200 × 1 200 pixels (11 × 11 µm) and its corresponding mask.



(a) : Input image.

(b) : Image with increased contrast.

(c) : Binary threshold applied.

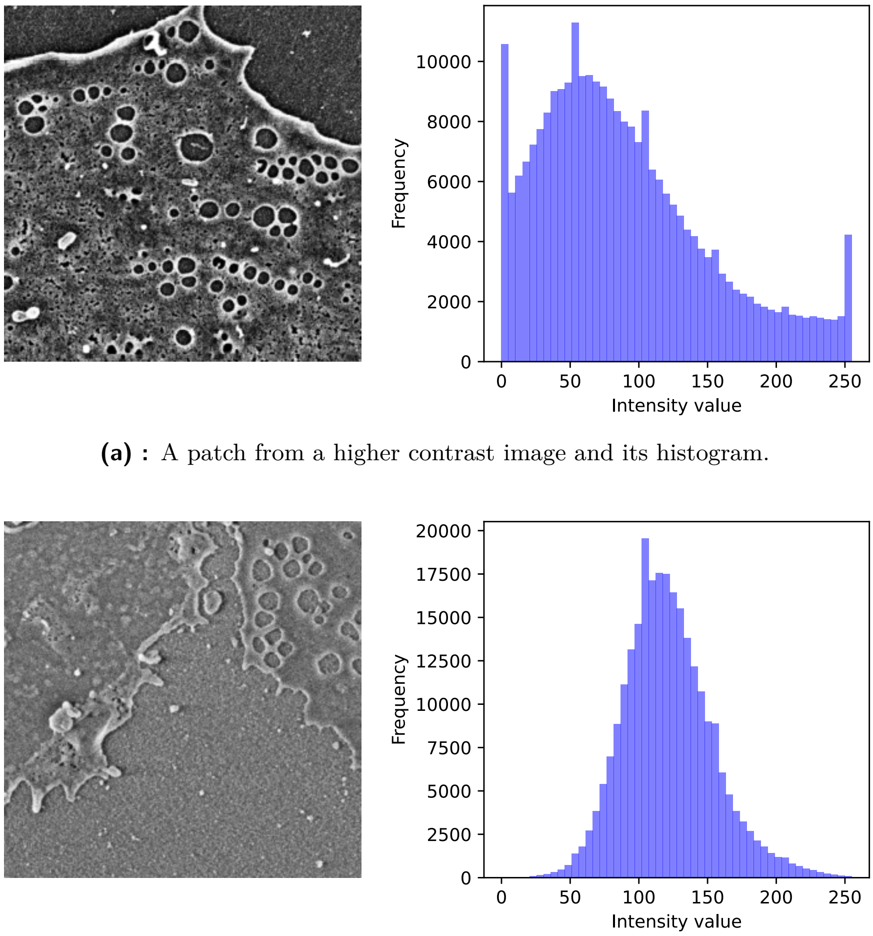
(d) : Binary mask made by removing objects from the binary threshold based on the object size and circularity.

Figure 2.6: The individual steps of the semiautomatic method in the order (a), (b), (c), and (d). Images taken from [11].

The masks were created by a semiautomatic method, which is based on the difference in contrast between the inside of fenestration and the LSEC membrane. The steps of this method include contrast enhancement followed by thresholding. The threshold value must be set manually so that only fenestrations and not their edges are marked. The last step of the semiautomatic method is the exclusion of objects based on their size and circularity [11]. The principle of this method is illustrated in Figure 2.6.

The main disadvantage of the semiautomatic method is having to set the binary threshold value because image intensity and contrast can vary in individual SEM images. This variation is mostly caused by the chemical properties of the staining materials used in specimen preparation [12].

The available dataset contains images with higher contrast and lower contrast (See Figure 2.7).



(a) : A patch from a higher contrast image and its histogram.

(b) : A patch from a lower contrast image and its histogram.

Figure 2.7: An example patch taken from a higher contrast image (a) and a lower contrast image (b).

From observing the semiautomatic masks, it can be seen that the semiautomatic method has certain limitations. Binary masks can contain falsely detected fenestrations (False positives). This is mainly caused by the darker parts of the cell membrane being detected as fenestrations (See Figure 2.8). Some of these areas are the same size as fenestrations, meaning the false positives cannot be filtered out based on their size. On the other hand, some fenestrations are not detected at all (False negatives). This happens when there is not enough contrast between the fenestration edge and the middle of the fenestration (See Figure 2.8).

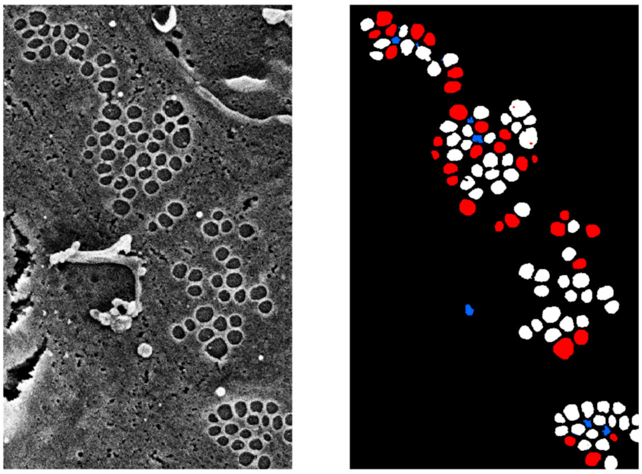


Figure 2.8: An example of a part of an image and its corresponding mask created by the semiautomatic method. False negative detections are indicated in red, false positives in blue, and correct detections are indicated in white.

2.4 A comment on ground truth masks

The masks, which are referred to as the ground truth in this thesis, were created by the author of this thesis, and Jakub Pospíšil, the thesis co-supervisor. They were made by editing the semiautomatically segmented masks, as that was the fastest and easiest way to acquire masks with marked fenestrations. Creating enough ground truth masks from scratch for both training and statistical evaluation would be extremely time-consuming, as there are approximately 3 000 fenestrations in each image (~100 000 fenestrations in the whole dataset). Even with our simplified approach, each mask took approximately over an hour of editing to reach a satisfactory state. The changes made to the semiautomatic masks include false detection removal and the addition of undetected fenestrations. We created these ground truth masks to the best of our abilities, but we do not claim that they are perfect, as they can still contain some overlooked false detections or missing fenestration. Both the training and testing ground truth masks are published online and can be found at https://drive.google.com/drive/folders/18O8pFbqFLx34X1dliWbPf9EkqeFO0ASK.

2.5 Fenestration properties

LSEC segmentation masks can be used for the computation of fenestration properties, as described in [11]. Examples of relevant fenestration properties are the number of fenestrations per cell area (frequency) and the total area of fenestrations per cell area (porosity). Fenestration frequency is computed as

$$f = \frac{\text{number of fenestrations}}{\text{cell area}}. \tag{2.1}$$

Fenestration porosity is computed as

$$p = \frac{\text{total area of fenestrations}}{\text{cell area}} \cdot 100. \tag{2.2}$$

Additional fenestration properties are fenestration size, fenestration roundness, and equivalent circle diameter.

When computing these properties, fenestrations are assumed to be elliptical in shape. Fenestration properties are computed by fitting an ellipse to each found fenestration in a segmentation mask. The major and minor axes of each ellipse can then be used to find the fenestration roundness, which is computed as:

$$R = \frac{\text{minor axis}}{\text{major axis}}. \tag{2.3}$$

The equivalent circle diameter of an ellipse is defined as the diameter of a circle of the same area as the ellipse. It is computed as

$$d = \sqrt{\text{minor axis} \cdot \text{major axis}}. \tag{2.4}$$

Chapter 3

Segmentation of LSEC fenestrations

An automatic method for LSEC segmentation was proposed in [13], using a fully convolutional neural network (FCNN). The authors use atomic force microscopy (AFM), an imaging modality different from the one used in this thesis. The authors present the achieved mean Dice score of 0.773 by comparing 14 automatically segmented images of size 256 × 256 pixels to manually segmented ones [13]. Unfortunately, the proposed model weights and code cannot be found online.

In [11], Ilastik software was used to segment LSEC fenestrations in SEM images automatically. The segmentation results were compared to manually segmented images and a semiautomatic method, giving satisfactory results. The authors list the simplicity of the used software and no need to work with code as advantages of using Ilastik. As a disadvantage, the authors mention the need to manually adjust the contrast and brightness of segmented images. A disadvantage of Ilastik is that it does not offer much flexibility in tuning the segmentation model compared to using a programming language like Python and its machine learning libraries. Another disadvantage of using Ilastik is that one needs to have a GPU, so the training is not too time-consuming [11].

Groups like ZeroCostDLForMic [14] have published various neural network implementations for general medical image segmentation and classification on their GitHub page. These implementations are in the form of Python notebooks intended to be run in Google Colab, and the networks can be trained on Google's GPUs. This approach enables anyone to train their model with their own data without the need to own a GPU for model training.

To the best knowledge of the author of this thesis, there is no publicly available model trained specifically for the segmentation of LSEC fenestrations in SEM images.

3.1 CNN architectures for computer vision tasks

Convolutional neural networks (CNNs) are a popular choice for solving computer vision tasks both in medical and real-world images [15, 16]. They can be used in several areas, including image classification, semantic segmentation, or object detection [15].

As with any method, CNNs also have their drawbacks, some of which are the need for good training data, large processing power requirements, and their complexity compared to more traditional methods. Despite these limitations, they have been used in many computer vision tasks because of the state-of-the-art results they provide [15].

A large breakthrough in convolutional networks was made in 2012 by Alex Krizhevsky [17] by training a large network on the ImageNet [18] dataset. Krizhevsky won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with his proposed solution, AlexNet. The challenge tasks were the classification and localization of common objects. Over a million labeled images were provided for the challenge participants for training [19].

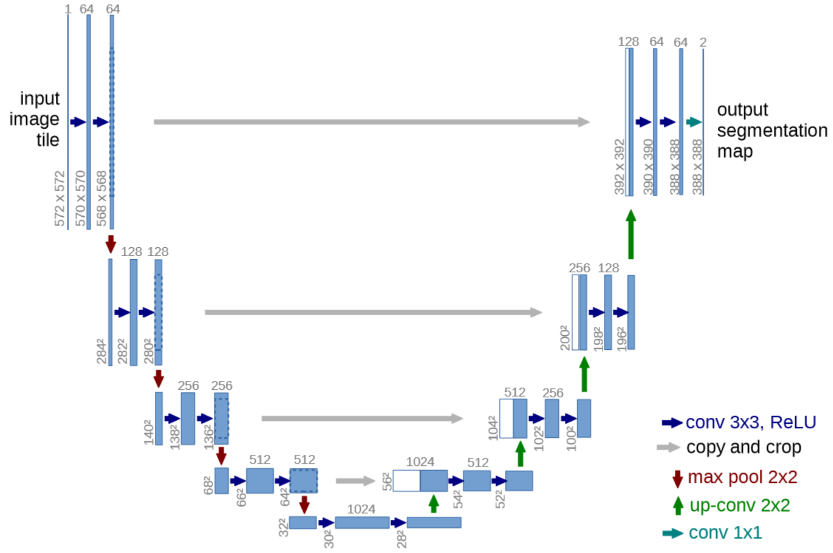


Figure 3.1: The U-net architecture, as presented in [16]. Blue boxes represent multi-channel feature maps. White boxes represent copied feature maps. The number of channels is denoted on top of the boxes. The feature map sizes are provided at their lower left edge. The arrows denote the different operations. Taken from [16].

In 2014, the Visual Geometry Group (VGG) participated in ILSVRC. They placed first in the classification task and second in the localization task, and VGG is a popular architecture for computer vision tasks to this day [20]. In this architecture, the size of the convolution kernels was lowered while increasing the number of network layers to 16–19 [15, 21].

In 2015, an architecture called U-Net [16] was proposed for the task of medical image segmentation. It is an encoder-decoder network. The encoder comprises a series of layers that progressively downscale the image, and the decoder then upscales the encoder output. The outputs of some encoder layers are connected to inputs of decoder layers with skip-connections, which enables precise localization. The U-Net architecture is illustrated in Figure 3.1.

Another architecture proposed in 2015 is the Residual Network (ResNet) [22]. ResNet uses shortcut connections to solve the vanishing gradient problem of deep networks. The shortcut connections should ease their training. A ResNet building block is shown in Figure 3.2.

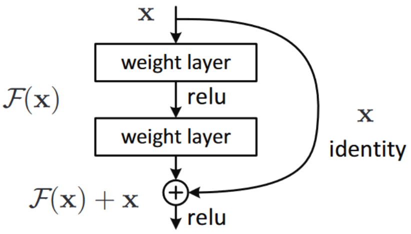


Figure 3.2: A building block of a ResNet, showing a shortcut connection with identity mapping of its input x. $F(x)$ is the original non-linear mapping and $F(x) + x$ is called the residual mapping. relu is the Rectified linear unit. Taken from [22].

A ResNet, even with 152 layers, 8× deeper than a VGG net, has lower complexity thanks to using shortcut connections. Compared to VGG, ResNets tends to show higher performance and error rates [15].

The number of layers of a specific network implementation can differ depending on the specific task. This is denoted in the network name by appending the number of layers to the network name. For example, ResNet18 means a ResNet architecture with 18 layers. The choice of architecture and its depth for a given task depends on the computation and size requirements and the specific task properties. The comparison of common architecture sizes is shown in Figure 3.3.

Recently, attempts have been made to create networks, which try to solve the task of segmenting any object in a natural scene. Meta AI published its Segment Anything Model (SAM)[1] in 2023 [24]. SAM uses Vision Transformer (ViT)-based image encoder [25]. It was trained on 11 million images with 1 billion masks. Even though microscopy images were included in the training data, most of the training data include common objects. SAM performs best when the segmented object is marked by the user, either by a bounding box around the object or individual points [24]. A disadvantage of such a general model is the huge model size (the weights file size for the default SAM is 2.4 GB).

Inspired by SAM, in 2023, MicroSAM [26] was introduced, which was developed to segment microscopy data. MicroSAM builds on top of SAM by finetuning it with images from light and electron microscopy. To segment an image, MicroSAM needs the user to mark the segmented objects by a bounding box or individual points. It is mainly intended for the segmentation

[1] https://github.com/facebookresearch/segment-anything

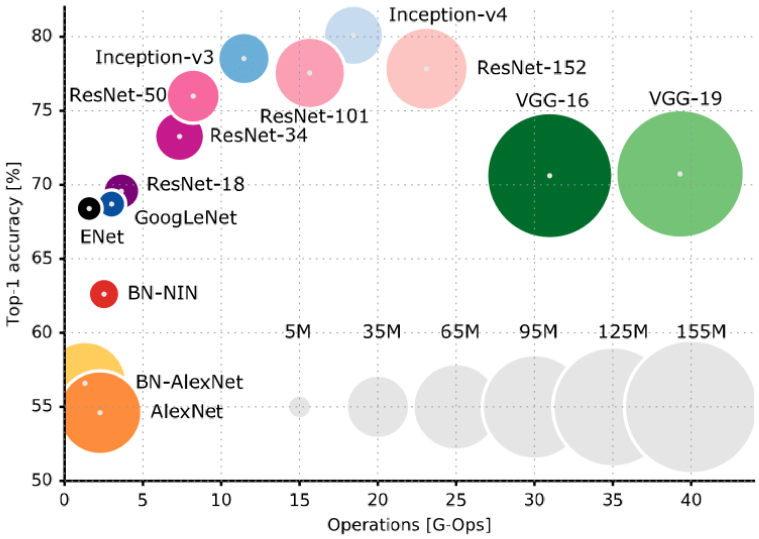


Figure 3.3: Comparison of popular CNN architectures used in computer vision tasks. The vertical axis shows top 1 accuracy on ImageNet classification. The horizontal axis shows the number of operations needed to classify an image. Circle size is proportional to the number of parameters in the network [23]. Taken from [23].

of whole cells, organelles, and nuclei.

3.2 Network training

To train a network, available data is usually preprocessed and split into training and validation datasets. The training data is used to learn the model weights, while the validation data estimates the model's effectiveness. Both the training and validation data are grouped into smaller batches. Model weights are updated when one batch of training data passes through the model. Training is performed in epochs. An epoch is concluded after each complete pass of the training data through the network [27].

3.2.1 Image augmentations

Image transformations like random rotation or flipping along the horizontal or vertical axis can be used to create augmented images. The main goal of creating augmented images is to increase the training dataset size and prevent overfitting [28].

Augmentations of image range can also be used, for example, histogram equalization to increase the image contrast [28]. A possible variation of histogram equalization is the Contrast Limited Adaptive Histogram Equalization (CLAHE) [29]. In CLAHE, the image is divided into small blocks called tiles. In each tile, the histogram is equalized using its cumulative distribution function. If any histogram bin is above the specified contrast limit, those pixels are clipped and distributed uniformly to other bins before applying the histogram equalization. The clipping is applied to reduce noise amplification [30].

3.2.2 Fine-tuning

Fine-tuning is the process of initializing a predefined model architecture with its pretrained weights. Weights, which are commonly used for fine-tuning, were trained on the ImageNet dataset. This initialized model can then be retrained with new data. The observed benefits of utilizing pre-trained weights are better model performance and shorter training times. Fine-tuning using weights pretrained with ImageNet has been shown to be beneficial for models intended for the medical image and microscopy domain [31, 32].

3.2.3 Loss function

The loss function measures the dissimilarity between the ground truth and the predicted segmentation. It is used during the training phase for the network to learn predictions that are close to the ground truth in terms of an evaluation metric. A commonly used metric is the Binary cross entropy (BCE) [33, 34], defined as:

$$L_{\text{BCE}(y,\hat{y})} = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log(\hat{y}) + (1-y_n)\log(1-\hat{y})\right], \tag{3.1}$$

where y_n is the n-th ground truth pixel, and $\hat{y_n}$ is the corresponding model output.

The loss function is minimized by the optimizer, for example, stochastic gradient descent (SGD) [33].

3.2.4 Optimizer

The optimizer updates the model weights in order to minimize the loss function [35]. The update is based on the loss function derivative, and for the SGD optimizer, it is described by the equation

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta L(\theta), \tag{3.2}$$

where θ_t represents the model parameters at time t, η_t is the learning rate at time t, and $\nabla_\theta L(\theta)$ is a vector of first parital derivatives of $L(\theta)$ with respect to parameters θ.

SGD can be generalized by linearly combining the gradient direction with a constant multiple of the previous parameter update. The multiple constant is called the Nesterov momentum γ. The weight update with momentum is defined as [35]:

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta L(\theta) - \gamma(\theta_t - \theta_{t-1}). \tag{3.3}$$

3.2.5 Segmentation quality evaluation

Quality evaluation metrics measure the similarity between the segmentation result and the ground truth. Binary segmentation results can be evaluated with several metrics. Most of them are computed from the number of following predictions:

- True Positives (TP) - correctly recognized positive samples
- True Negatives (TN) - correctly recognized negative sample
- False Positives (FP) - incorrectly recognized negative samples
- False Negatives (FN) - incorrectly recognized positive samples

A commonly used metric in binary segmentation tasks, which uses both correctly and incorrectly recognized samples for its computation, is Dice score [36].

Dice score is computed as

$$D = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{3.4}$$

It can also be expressed as

$$D = \frac{2 \cdot |S \cap G|}{|S| + |G|}, \tag{3.5}$$

where

- $|S|$ is the number of positive pixels in the created binary mask.
- $|G|$ is the number of positive pixels in the ground truth binary mask.
- $|S \cap G|$ is the size of the intersection areas of the created and ground truth masks.

Dice score values lie in the range [0, 1], where 0 means no overlap, and 1 means perfect overlap [37].

Chapter 4

Methods

This chapter contains the details of the methods used in model development and evaluation.

4.1 Model training

The training dataset contains 11 images, the validation dataset contains 3 images, and the test dataset contains 22 images. Images for all three datasets were chosen so that each dataset contains images that visually differ from each other. Image differences are mainly caused by image contrast and fenestration frequency and size.

Each image usually contains more than one cell. In every image, there is one fully captured cell and only parts of other cells. Available semiautomatic masks only contain the segmented fenestrations of the fully captured cell in the image (See Figure 4.1). The partially captured cells were cropped out of the training and validation images because we do not have their segmentation masks. These cropped images were then used to create non-overlapping image patches of the size 224 × 224 pixels for training and validation.

Training and validation patches were not mixed together, meaning that training images were used to create training patches, and validation images were used to create validation patches. Using one image for both validation and training patches could lead to the model performing well on the validation data. All patches created from a single image have similar characteristics, which could lead to the better performance on the validation data, but the performance on new unseen images could be worse.

The 11 training images were split into 1 334 patches, and the 3 validation images were split into 472 patches. Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to these patches to make the image contrast and intensity more even over the whole dataset. The CLAHE clip limit was set to 10, and the tile size was set to 11 × 11 pixels. CLAHE can intensify the noise present in the images. To reduce this effect, median filtration with the kernel size of 3 × 3 pixels is performed following CLAHE.

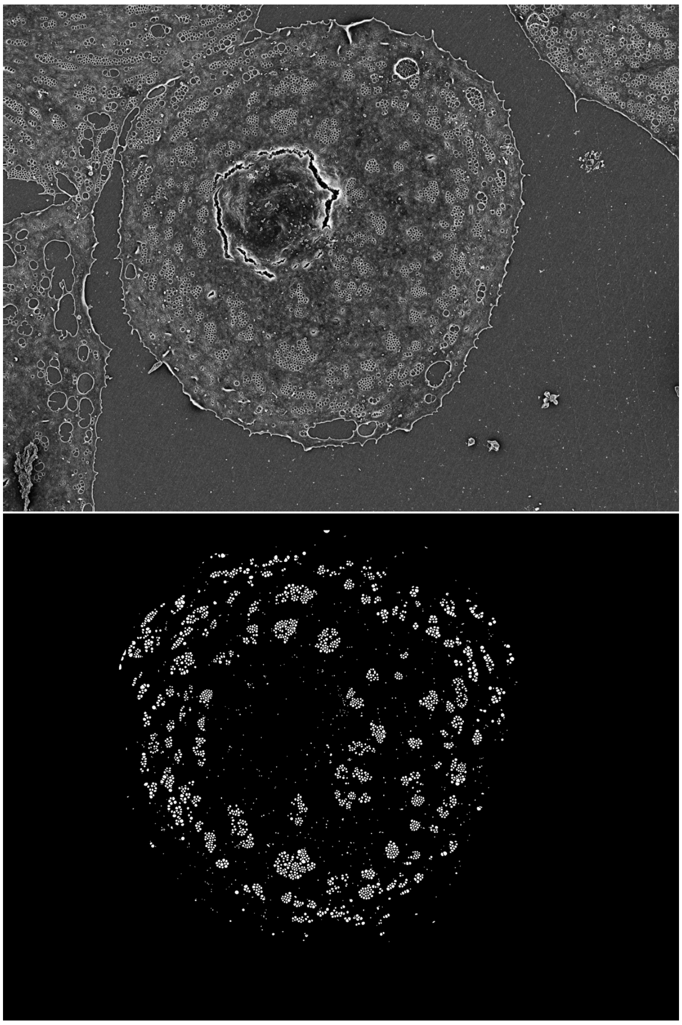
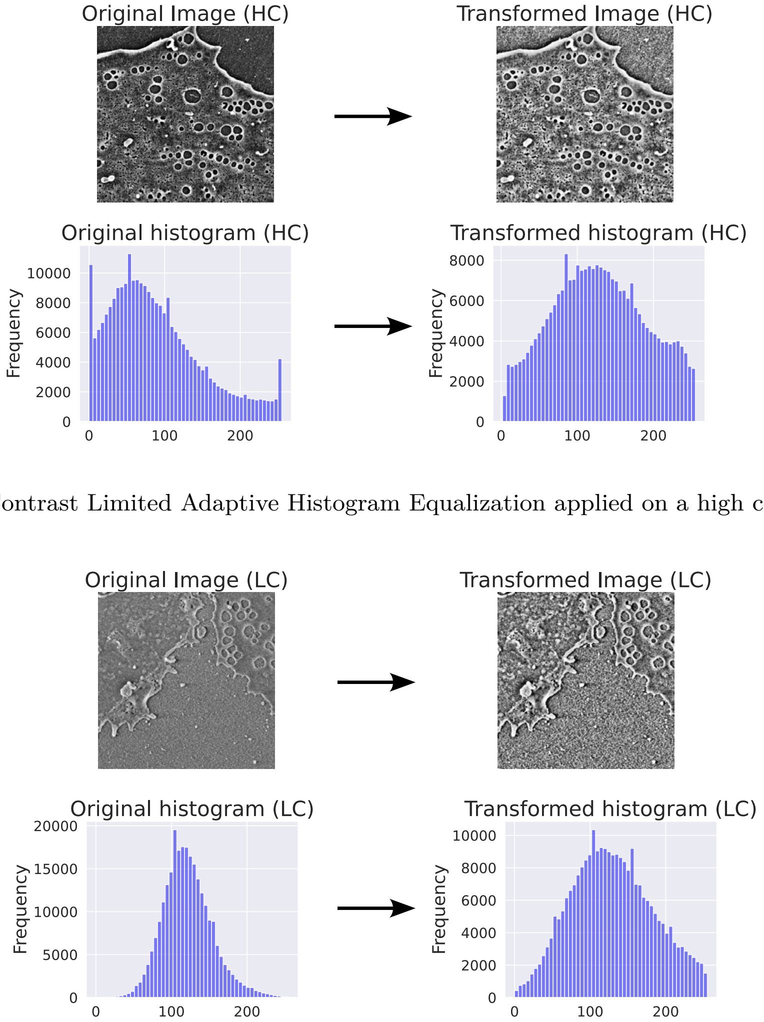


Figure 4.1: An example image and its semiautomatic mask. The image contains one full cell and parts of other cells. The segmentation mask contains only the fenestrations of the fully captured cell.

The effect of CLAHE on high contrast and low contrast images is shown in Figure 4.2.

The image range was normalized to have an image mean of 0 and a standard deviation of 1. The image patch size and image range were chosen so the pretrained ImageNet weights could be used for model initialization. Augmentations used for the training data are the following: horizontal and vertical flip, both with a 50% probability, and a resizing of images with the scale randomly chosen from the interval [0.9, 1.1].

The experiments with different model architectures included trying VGG and ResNet encoders of different depths, and trying simpler encoders like EfficientNet. Different loss functions and their combinations were also tested. A UNet with a ResNet34 encoder was chosen as the final architecture. Binary cross entropy was used as the loss function to train the model. The model was



(a) : Contrast Limited Adaptive Histogram Equalization applied on a high contrast image.

(b) : Contrast Limited Adaptive Histogram Equalization applied on a low contrast image.

Figure 4.2: The effect of Contrast Limited Adaptive Histogram Equalization (CLAHE) on images shown on high contrast (a) and low contrast (b) images and their respective histograms.

initialized with ImageNet weights, and no layers were freezed before training (the pretrained weights are used just to speed up the training). The sigmoid function is used as the final activation layer. The batch size is 32 patches, and the stochastic gradient descent (SGD) optimizer was used with a learning rate of 0.04, weight decay of 0.01, and momentum of 0.07. The validation loss is evaluated in each epoch, and if the validation loss improves, the model's current weights are saved.

The learning rate is decreased by a factor of 10 if the validation loss does not improve over 5 epochs, and training is stopped if the validation loss does not improve over 10 epochs. The maximum number of epochs was set to 200. The final model stopped on the 101st epoch, and the training took about 1 hour on NVIDIA T4 GPU. The training process is symbolized in Figure 4.3.

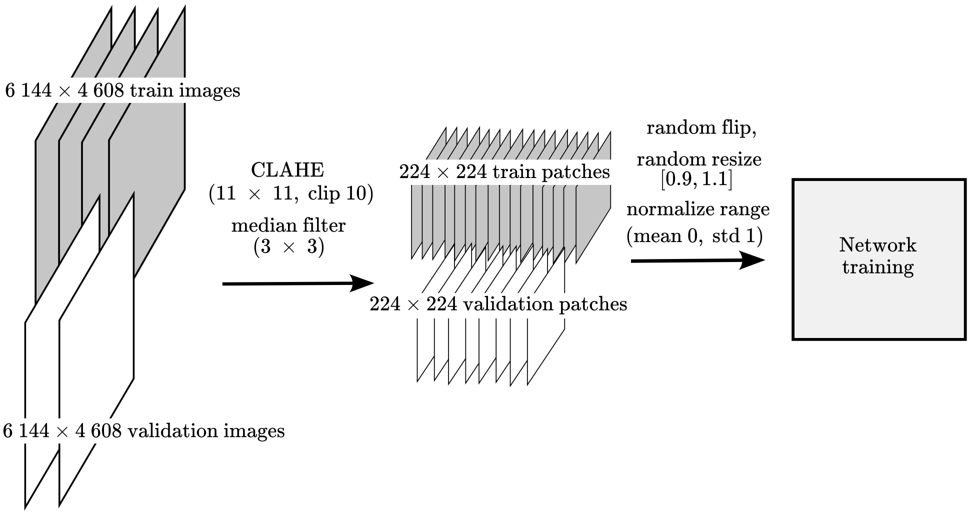


Figure 4.3: Illustration of the training process. Training and validation images are split into patches of size 224 × 224, and preprocessing is applied. The patches are fed into the network with the chosen transformations (random flip and image resize in the range [0.9, 1.1]). The training architecture is UNet with ResNet34 encoder. Binary Cross Entropy is used as the loss function, and SGD is used as the optimizer for training.

During inference, 20 pixels of the image borders are mirrored to avoid artifacts on the edges. Then, the image is split into overlapping tiles of 224 × 224 pixels with a 20-pixel overlap. These tiles are fed into the model, and the overlap is linearly weighted to ensure smooth transitions between tiles. A binary threshold of 0.4 is chosen for the model's output. The segmentation of 1 image takes about 6 seconds on NVIDIA T4 GPU. The inference process is symbolized in Figure 4.4.

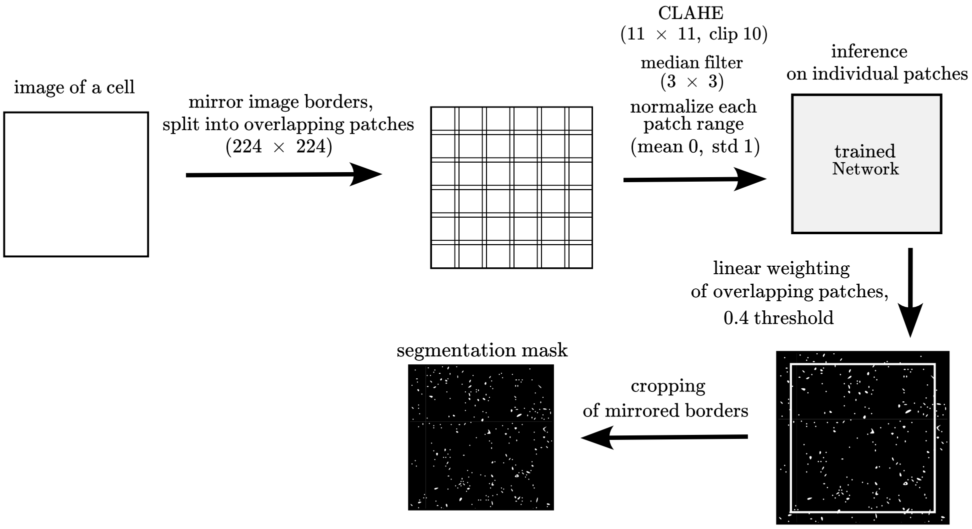


Figure 4.4: The inference process. 20 pixels of the image borders are mirrored, and the image is split into overlapping patches of 224 × 224 pixels with a 20-pixel overlap. Preprocessing, including CLAHE, median filter, and image normalization, are applied to individual patches. These image patches are then fed into the network, and the network outputs are put back together with a linear weighting on the patches' overlaps. The output of the network is thresholded with the value 0.4. Finally, the mask is cropped to get rid of the mirrored borders.

4.2 Evaluation

The model performance was evaluated using a binary segmentation evaluation metric (Dice score as described in 3.2.5) and statistics of fenestration properties described in section 2.5.

Before evaluation, manually created cell masks are applied to all of the segmentation masks including ground truth masks, semiautomatic masks, and automatic masks. The cell masks mark only the cell area of interest to exclude anything detected in the area of the cell nucleus and anything outside the cell of interest. An example cell mask is shown in Figure 4.5.

Ellipses are then fitted to all objects in the three datasets to remove anything with an equivalent circle diameter outside the range [50, 350] nm. For our evaluation, we do not consider anything outside this range to be a fenestration. This preprocessing performed before the mask comparison is described in Figure 4.6.

Dice scores are then computed to compare the similarity of each ground truth mask to its corresponding semiautomatic and automatic mask.

Statistics are then performed on the ground truth, semiautomatic, and automatic fitted ellipses. The number of detected fenestrations is computed for each image, and the correlation of the number of detected fenestrations is

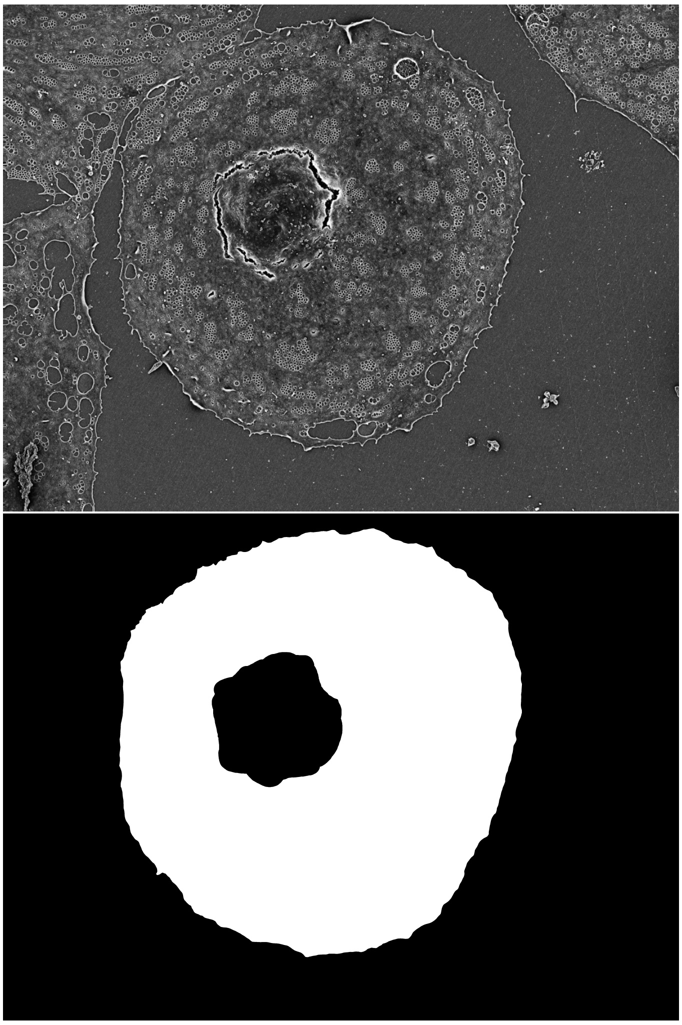


Figure 4.5: An example of an image and its cell mask, which is used to remove unwanted fenestrations. Fenestrations overlapping with the white pixels of the cell mask are kept, and fenestrations overlapping with the black pixels are removed.

computed for each method. Fenestration frequency can be calculated as the number of fenestrations in each cell divided by the cell area. Furthermore, cell porosity is computed for the segmented cell in each image [11]. Two approaches to computing the porosity are tested. In the first approach, the individual fenestration areas are computed as the areas of the fitted ellipses. In the second approach, the fenestration areas are computed directly from the white pixels in the segmentation masks. A histogram of fenestration roundness and equivalent diameter is also computed for the tested dataset.

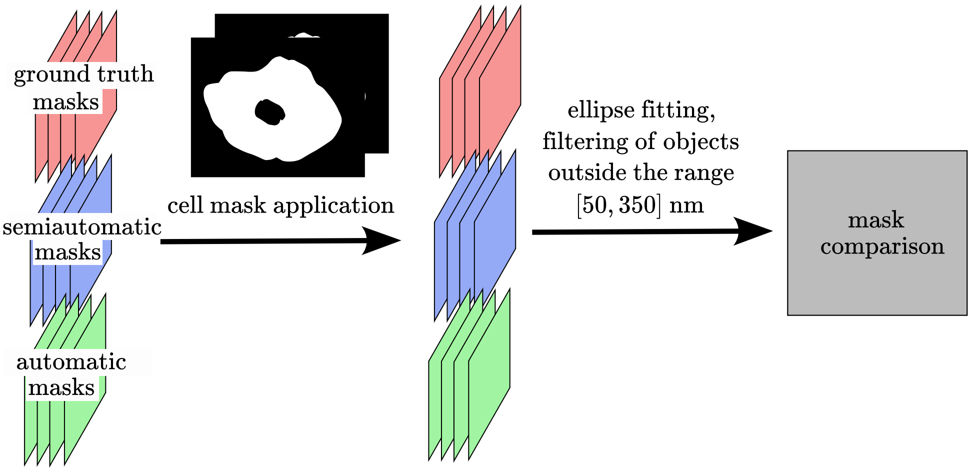


Figure 4.6: Illustration of the preprocessing of masks before their comparison. Cell masks are applied to the segmentation masks to remove anything outside the cell of interest and the cell nucleus. Ellipses are then fitted to the detected fenestrations to filter any fenestrations with a diameter outside the range [50, 350] nm. Ground truth masks, semiautomatic masks, and automatic masks are then compared.

Chapter 5

Results

Dice scores of each image computed for both methods can be found in Table 5.1. The mean dice score for the semiautomatic method is 0.92 ± 0.06. For the automatic method, it is 0.94 ± 0.02.

The histograms of the fenestration diameter and roundness for all detected fenestrations can be seen in Figure 5.1. The total number of fenestrations detected in all of the 22 test images is 73 906 in the ground truth dataset, 72 303 in the semiautomatic dataset, and 75 031 in the automatic dataset.

Porosity computed directly from white and black pixels for each cell can be found in Table 5.2, and the correlation plot in Figure 5.2. The tangent of the linear fit is 0.98 for the semiautomatic method and 1.02 for the automatic method. The coefficient of determination R^2 is 0.98 for the semiautomatic method and 1.00 for the automatic method. Porosity computed from the area of fitted ellipses for each cell can be found in Table 5.3, and the correlation plot in Figure 5.3. The tangent of the linear fit is 1.00 for the semiautomatic method and 1.04 for the automatic method. The coefficient of determination R^2 is 0.98 for the semiautomatic method and 1.00 for the automatic method.

Fenestration frequency for each cell can be found in Table 5.4, and the correlation plot in Figure 5.4. The tangent of the linear fit is 1.09 for the semiautomatic method and 0.99 for the automatic method. The coefficient of determination R^2 is 0.99 for the semiautomatic method and 1.00 for the automatic method.

The mean equivalent diameter of fenestrations in each binary mask is written in Table 5.5, and the correlation plot of the mean equivalent diameter of fenestrations in each image is shown in Figure 5.5. The tangent of the linear fit is 0.99 for the semiautomatic method and 0.95 for the automatic method. The coefficient of determination R^2 is 0.78 for the semiautomatic method and 0.97 for the automatic method.

Segmentation examples are shown in Figures 5.6 and 5.7. The examples shown here are small crops from the original images, as the full images are too large to be shown here. Comparison images of all 22 masks can be found in the thesis attachment.

method	**semiautomatic**	**automatic**
mask 1 (LC)	0.96	0.92
mask 2 (HC)	0.92	0.94
mask 3 (HC)	0.93	0.96
mask 4 (HC)	0.94	0.93
mask 5 (HC)	0.93	0.95
mask 6 (LC)	0.94	0.94
mask 7 (LC)	**0.98**	0.94
mask 8 (LC)	0.96	0.96
mask 9 (LC)	0.96	**0.96**
mask 10 (LC)	0.95	0.94
mask 11 (LC)	0.97	0.95
mask 12 (LC)	0.98	0.96
mask 13 (LC)	0.97	0.94
mask 14 (HC)	0.95	0.95
mask 15 (HC)	0.95	0.95
mask 16 (HC)	0.96	0.96
mask 17 (HC)	0.92	0.92
mask 18 (HC)	0.92	0.94
mask 19 (LC)	0.90	0.94
mask 20 (LC)	0.80	0.91
mask 21 (LC)	0.87	0.92
mask 22 (LC)	**0.69**	**0.90**
mean value	0.92 ± 0.06	0.94 ± 0.02

Table 5.1: Mask dice scores computed for the semiautomatic and automatic method, comparing the masks to the ground truth masks. The contrasts of the images from which the masks were computed are denoted in parentheses. (HC) = high contrast image, (LC) = low contrast image. The min and max value for each method is written in bold.

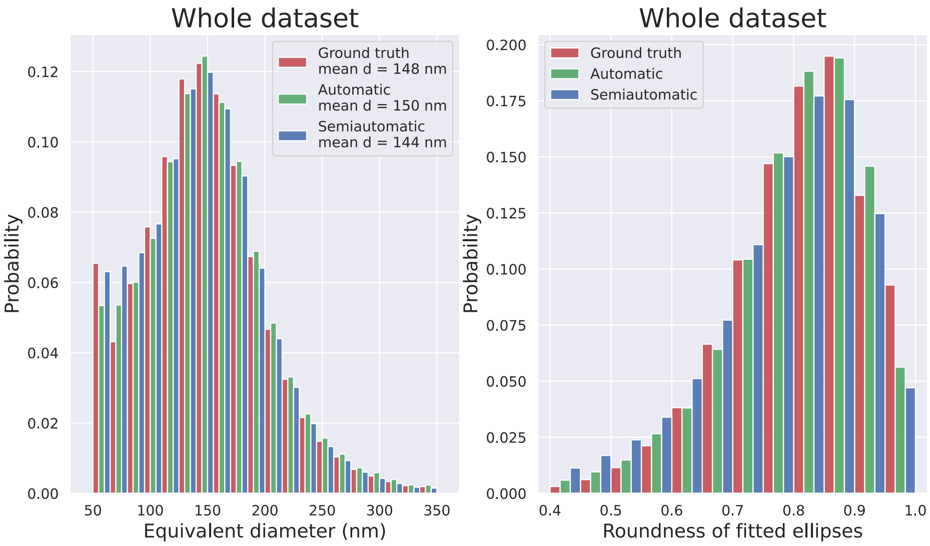


Figure 5.1: Histogram of fenestration size and roundness computed for the whole dataset of 22 test images. The equivalent diameter was computed with 20 bins in the range [50, 350]. The roundness diameter was computed with 12 bins in the range [0.4, 1]. The total number of fenestration is 73 906 in the ground truth masks, 72 303 in the semiautomatically generated masks, and 75 031 in the automatically generated masks. The mean diameter value is computed from all the detected fenestrations in each dataset.

method	**ground truth (%)**	**semiautomatic (%)**	**automatic (%)**
mask 1 (LC)	6.4	6.5	6.9
mask 2 (HC)	9.0	8.4	9.1
mask 3 (HC)	11.6	10.8	11.5
mask 4 (HC)	12.0	11.4	12.6
mask 5 (HC)	12.3	11.6	12.9
mask 6 (LC)	9.3	8.8	9.5
mask 7 (LC)	4.5	4.5	4.8
mask 8 (LC)	1.9	1.8	1.9
mask 9 (LC)	8.6	8.3	8.7
mask 10 (LC)	7.1	6.7	7.3
mask 11 (LC)	6.0	6.0	6.3
mask 12 (LC)	7.4	7.4	7.8
mask 13 (LC)	6.7	6.5	7.0
mask 14 (HC)	6.1	5.9	6.1
mask 15 (HC)	7.3	6.9	7.6
mask 16 (HC)	6.8	6.6	6.9
mask 17 (HC)	7.3	6.7	7.6
mask 18 (HC)	9.1	8.4	9.0
mask 19 (LC)	4.9	4.0	5.1
mask 20 (LC)	4.1	2.8	4.2
mask 21 (LC)	5.2	4.2	5.4
mask 22 (LC)	1.5	0.9	1.6

Table 5.2: Cell porosity expressed as the percentage of the cell area computed for each image. The fenestration area was computed directly from the number of fenestration pixels in the segmented masks. The contrasts of the images from which the masks were computed are denoted in parentheses. (HC) = high contrast image, (LC) = low contrast image.

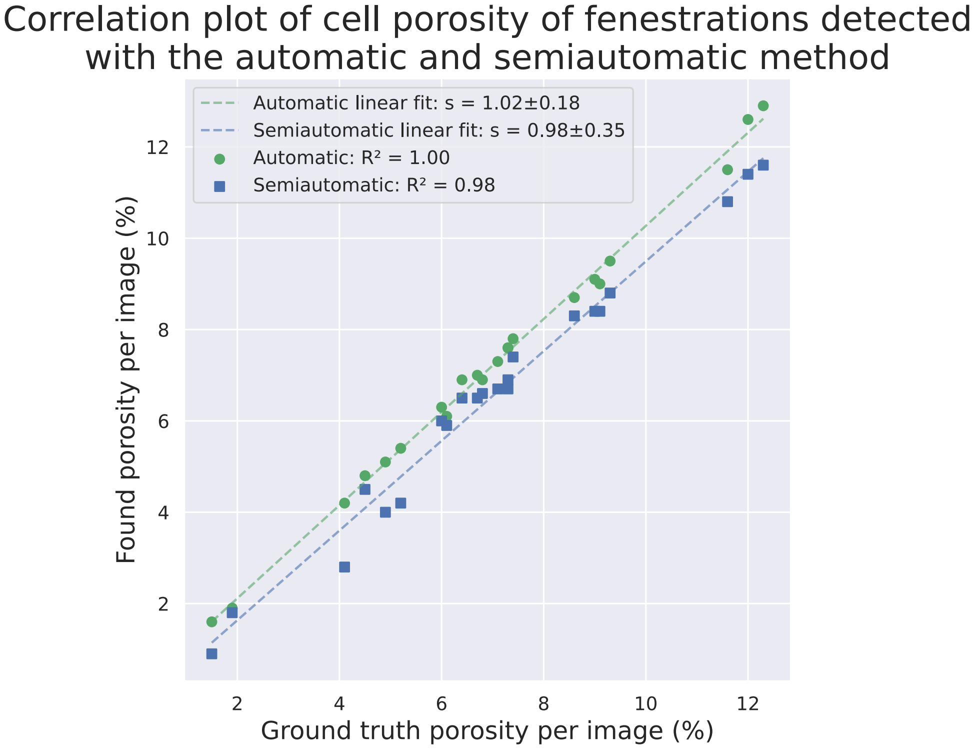


Figure 5.2: A correlation plot of the cell porosity in each image computed directly from mask pixels. Both data sets were fitted with a line with a tangent of 0.98 for the semiautomatic method and 1.02 for the automatic method. The coefficient of determination R^2 is 0.98 for the semiautomatic method and 1.00 for the automatic method.

method	**ground truth** (%)	**semiautomatic** (%)	**automatic** (%)
mask 1 (LC)	6.0	6.1	6.4
mask 2 (HC)	8.5	8.0	8.8
mask 3 (HC)	11.1	10.3	11.3
mask 4 (HC)	11.5	10.9	12.3
mask 5 (HC)	11.5	11.0	12.1
mask 6 (LC)	8.8	8.4	9.0
mask 7 (LC)	4.6	4.3	5.1
mask 8 (LC)	1.8	1.7	1.8
mask 9 (LC)	8.2	7.9	8.4
mask 10 (LC)	6.7	6.4	7.0
mask 11 (LC)	5.9	5.9	6.2
mask 12 (LC)	7.1	7.0	7.4
mask 13 (LC)	6.4	6.2	6.7
mask 14 (HC)	5.9	5.6	5.9
mask 15 (HC)	7.0	6.6	7.5
mask 16 (HC)	6.6	6.3	6.8
mask 17 (HC)	6.9	6.3	7.3
mask 18 (HC)	8.7	8.0	8.7
mask 19 (LC)	4.7	3.9	4.9
mask 20 (LC)	4.2	2.8	4.3
mask 21 (LC)	5.2	4.2	5.5
mask 22 (LC)	1.6	0.9	1.7

Table 5.3: Cell porosity expressed as the percentage of the cell area computed for each image. The fenestration area was computed from the area of fitted ellipses. The contrasts of the images from which the masks were computed are denoted in parentheses. (HC) = high contrast image, (LC) = low contrast image.

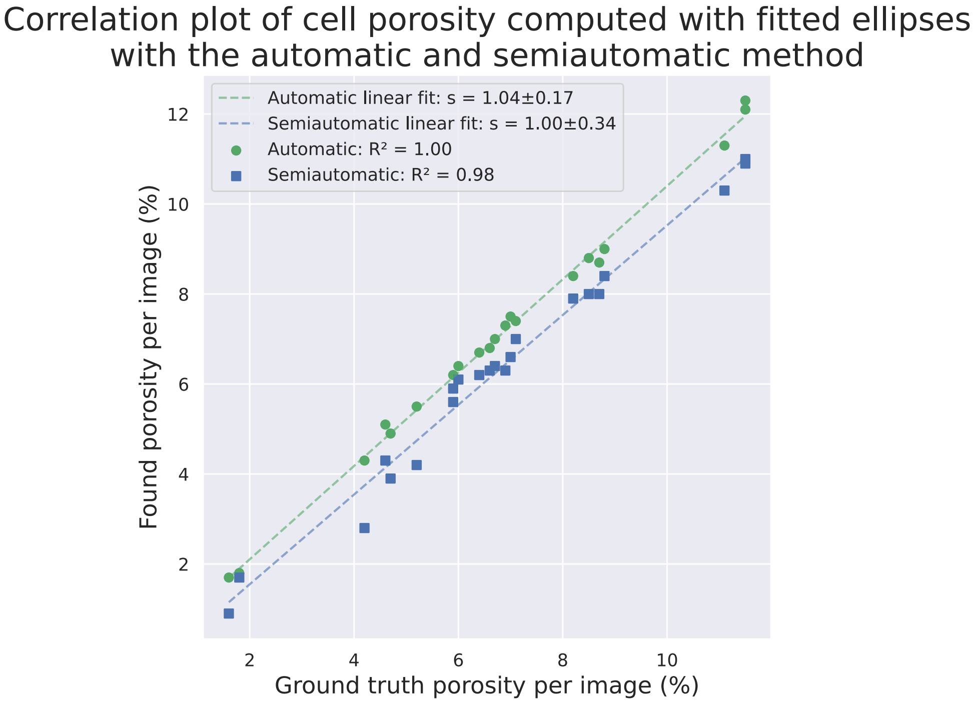


Figure 5.3: A correlation plot of the computed cell porosity in each image computed from fitted ellipses. Both data sets were fitted with a line with a tangent of 1.00 for the semiautomatic method and 1.04 for the automatic method. The coefficient of determination R^2 is 0.98 for the semiautomatic method and 1.00 for the automatic method.

method	**ground truth (fenestr./µm^2)**	**semiautomatic (fenestr./µm^2)**	**automatic (fenestr./µm^2)**
mask 1 (LC)	6.0	6.2	5.9
mask 2 (HC)	4.4	4.8	4.4
mask 3 (HC)	4.4	4.5	4.3
mask 4 (HC)	6.8	7.2	6.7
mask 5 (HC)	8.0	8.3	8.1
mask 6 (LC)	4.3	4.2	4.4
mask 7 (LC)	1.5	1.6	1.7
mask 8 (LC)	0.9	0.8	0.9
mask 9 (LC)	3.1	3.1	3.1
mask 10 (LC)	3.6	3.3	3.8
mask 11 (LC)	3.1	3.1	3.3
mask 12 (LC)	3.6	3.6	3.8
mask 13 (LC)	3.0	2.8	3.3
mask 14 (HC)	2.9	2.8	2.8
mask 15 (HC)	3.8	3.7	3.7
mask 16 (HC)	3.2	3.0	3.2
mask 17 (HC)	3.8	3.6	4.0
mask 18 (HC)	4.6	4.6	4.7
mask 19 (LC)	2.3	1.9	2.3
mask 20 (LC)	1.6	0.9	1.6
mask 21 (LC)	1.8	1.4	1.8
mask 22 (LC)	0.6	0.5	0.6

Table 5.4: Fenestration frequency in fenestrations per µm^2 computed for each image. The contrasts of the images from which the masks were computed are denoted in parentheses. (HC) = high contrast image, (LC) = low contrast image.

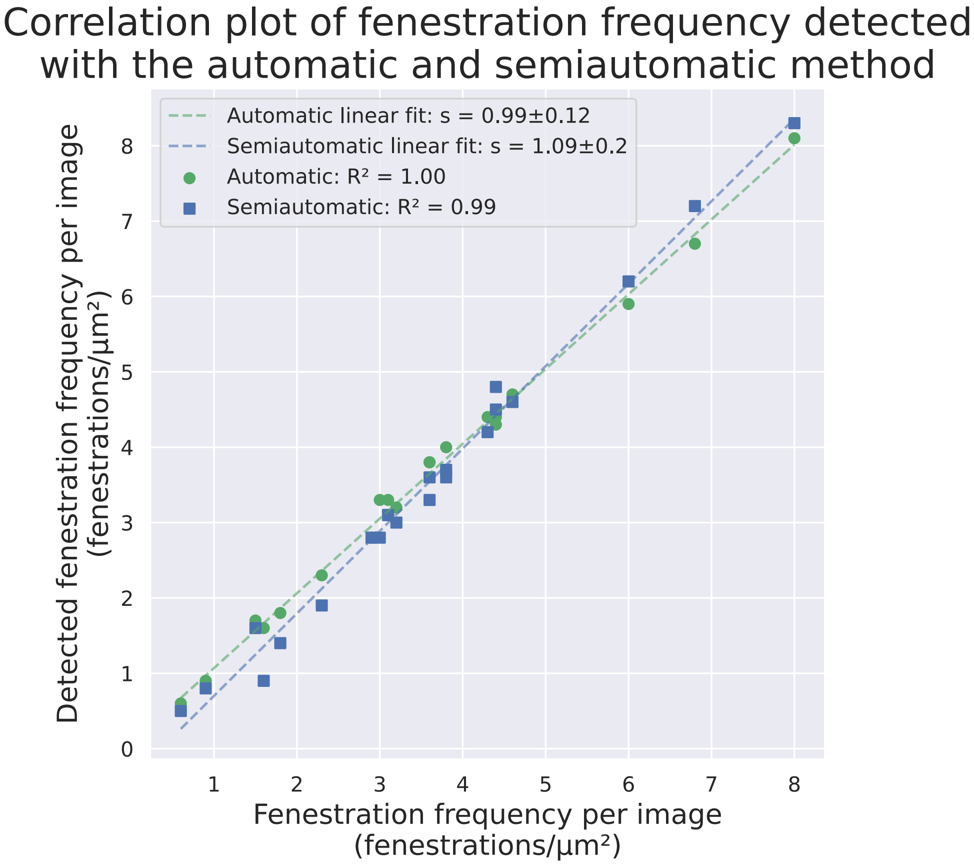


Figure 5.4: A correlation plot of the computed fenestration frequency in each image. Both data sets were fitted with a line with a tangent of 1.09 for the semiautomatic method and 0.99 for the automatic method. The coefficient of determination R^2 is 0.99 for the semiautomatic method and 1.00 for the automatic method.

method	**ground truth (nm)**	**semiautomatic (nm)**	**automatic (nm)**
mask 1 (LC)	106	105	111
mask 2 (HC)	131	124	133
mask 3 (HC)	138	130	143
mask 4 (HC)	142	139	148
mask 5 (HC)	144	142	144
mask 6 (LC)	145	140	144
mask 7 (LC)	146	146	145
mask 8 (LC)	148	135	151
mask 9 (LC)	149	154	148
mask 10 (LC)	149	153	147
mask 11 (LC)	150	152	153
mask 12 (LC)	150	153	154
mask 13 (LC)	151	151	150
mask 14 (HC)	153	150	152
mask 15 (HC)	155	151	155
mask 16 (HC)	156	161	152
mask 17 (HC)	165	135	173
mask 18 (HC)	168	187	167
mask 19 (LC)	173	161	174
mask 20 (LC)	178	174	177
mask 21 (LC)	182	188	184
mask 22 (LC)	184	175	183

Table 5.5: Mean equivalent circle diameter of fenestrations in each binary mask. The contrasts of the images from which the masks were computed are denoted in parentheses. (HC) = high contrast image, (LC) = low contrast image.

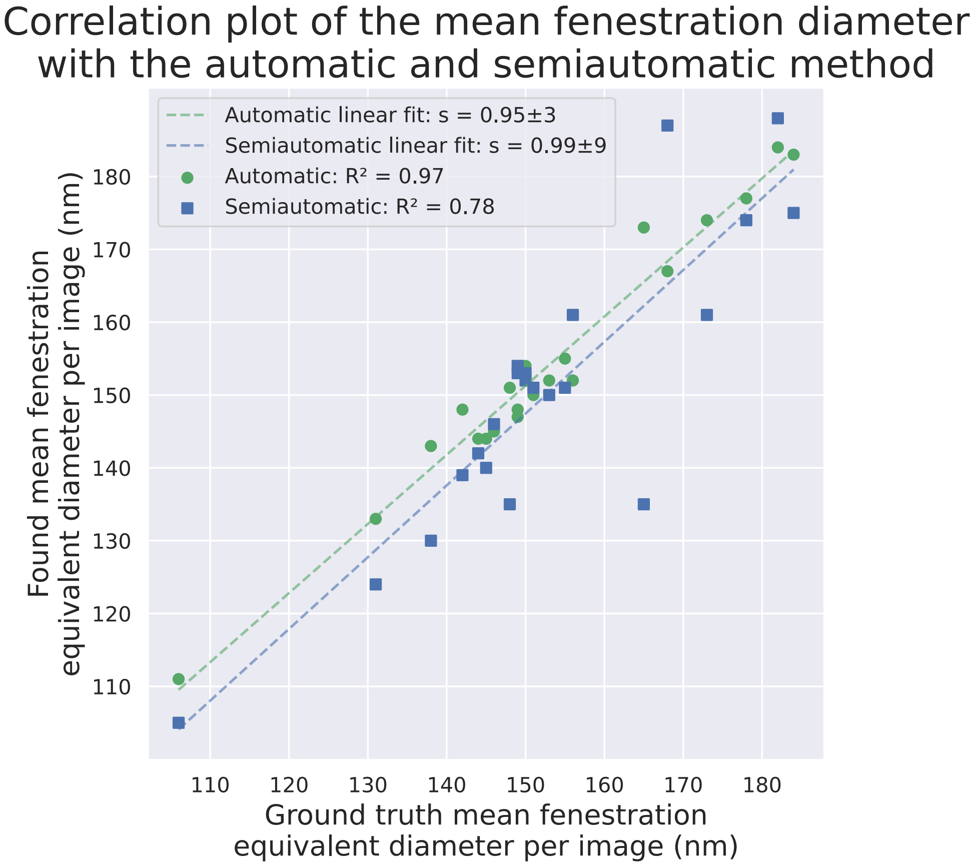
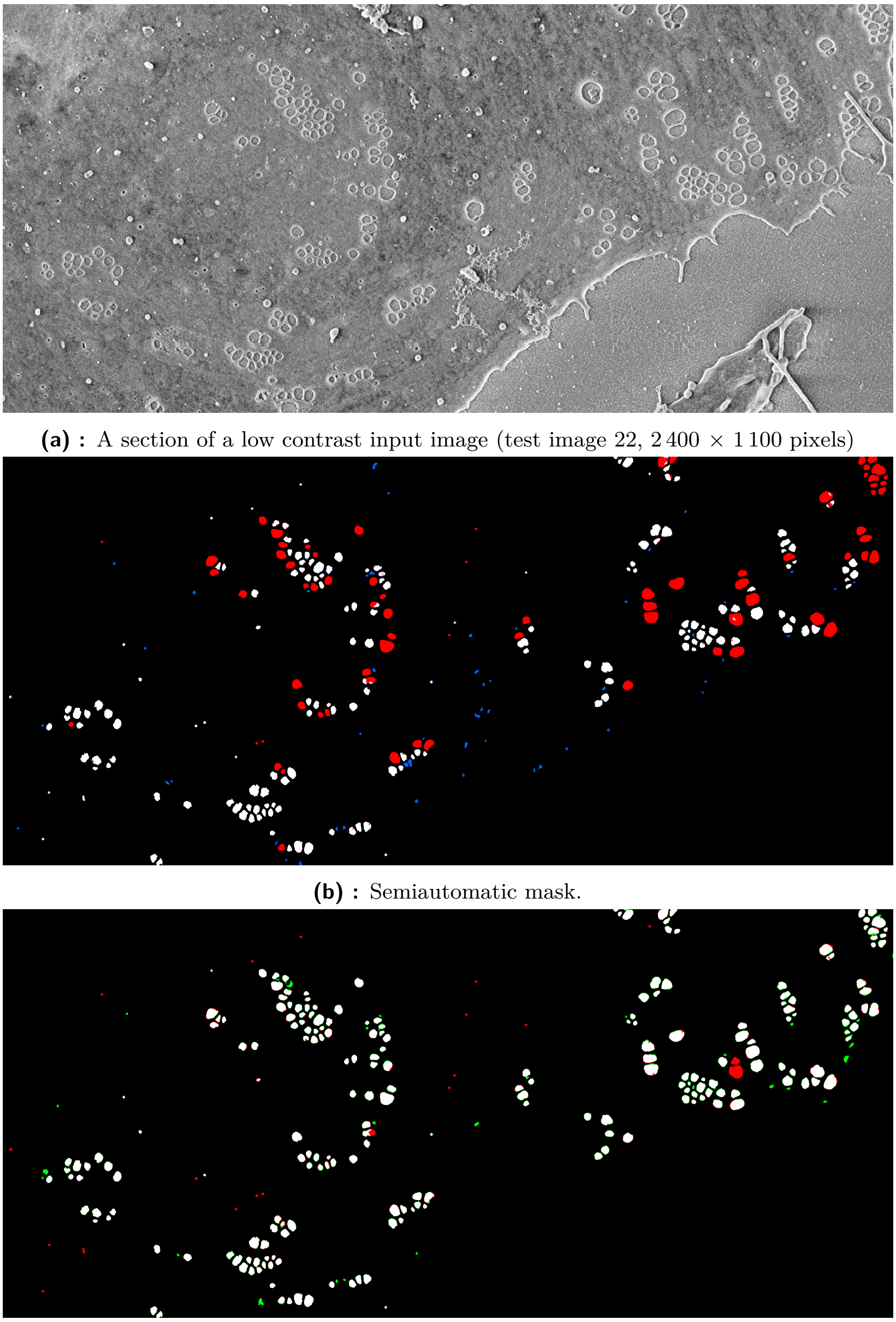


Figure 5.5: A correlation plot of the mean equivalent diameter of fenestrations in each image. Both data sets were fitted with a line with a tangent of 0.99 for the semiautomatic method and 0.95 for the automatic method. The coefficient of determination R^2 is 0.78 for the semiautomatic method and 0.97 for the automatic method.

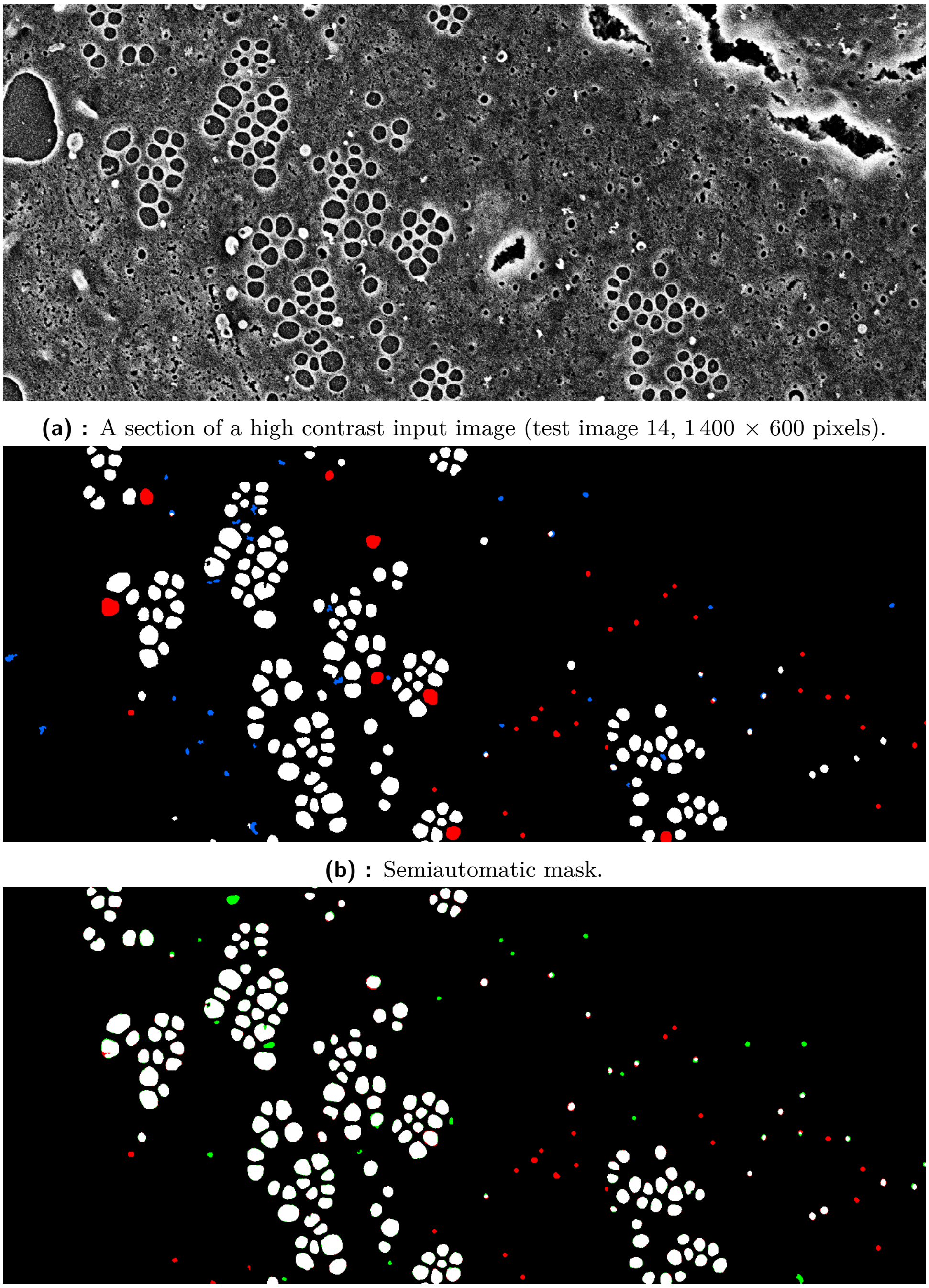


(a) : A section of a low contrast input image (test image 22, 2 400 × 1 100 pixels)

(b) : Semiautomatic mask.

(c) : Automatic mask.

Figure 5.6: A comparison of a semiautomatic mask with the ground truth (b) and automatic mask with the ground truth (c). Segmented mask overlap with the ground truth is denoted in white, undetected fenestrations by each method are shown as red, and false detections are shown as blue in (b) and green in (c).



(a) : A section of a high contrast input image (test image 14, 1 400 × 600 pixels).

(b) : Semiautomatic mask.

(c) : Automatic mask.

Figure 5.7: A comparison of a semiautomatic mask with the ground truth (b) and automatic mask with the ground truth (c). Segmented mask overlap with the ground truth is denoted in white, undetected fenestrations by each method are shown as red, and false detections are shown as blue in (b) and green in (c).

Chapter 6

Conclusions

The main goal of this thesis was to develop an automatic method for the segmentation of scanning electron microscope (SEM) images of *in vitro* liver sinusoidal endothelial cells (LSECs). LSEC segmentation masks are useful for the computation of fenestration properties. A commonly used LSEC segmentation method is semiautomatic, requiring manual parameter setting by the user. This can be time-consuming, as the contrast of each SEM image can vary.

In this thesis, a method for automatic fenestration segmentation was developed and compared to the semiautomatic method. The comparison is performed on segmentation masks created from 22 images with both methods.

The proposed method shows good performance both on low contrast and high contrast images, which are contained in the available dataset. The mean dice score is 0.92 ± 0.06 for the semiautomatic method and 0.94 ± 0.02 for the automatic method.

Creating good ground truth data of segmented fenestrations is extremely difficult, and we are aware that the ground truth images could contain some mistakes, but the evaluation of the proposed method would not have been possible without these ground truth estimates.

The main advantage of the automatic method over the semiautomatic is that the automatic method requires no manual parameter setting or adjustment.

Future work could involve making the proposed method more robust to images with lower contrast than the ones used in this thesis. SEM images with very low contrast and high noise content often cause the semiautomatic method to fail (even manual detection would be difficult for these types of images). After developing the automatic method, we tried to use it on these images, and the results were not satisfactory. These images have not been published yet, and we do not have their ground truth masks. Although these experiments were not included in this thesis, we plan to continue developing the proposed automatic method to handle a wider range of SEM images, from low to high contrast.

Bibliography

[1] M. Muto, M. Nishi, and T. Fujita. Scanning Electron Microscopy of Human Liver Sinusoids. *Archivum histologicum japonicum*, 40(2), 1977. doi:10.1679/aohc1950.40.137.

[2] S. Shetty, P. Lalor, and D.H. Adams. Liver sinusoidal endothelial cells — gatekeepers of hepatic immunity. *Nature Reviews Gastroenterology & Hepatology*, 15, 2018. doi:10.1038/s41575-018-0020-y.

[3] S. Hilmer, V. Cogger, R. Fraser, A. McLean, D. Sullivan, and D. Le Couteur. Age-related changes in the hepatic sinusoidal endothelium impede lipoprotein transfer in the rat. *Hepatology*, 42(6), 2005. doi:10.1002/hep.20937.

[4] J. Gracia-Sancho, E. Caparrós, A. Fernández-Iglesias, and R. Francés. Role of liver sinusoidal endothelial cells in liver diseases. *Nature Reviews Gastroenterology & Hepatology*, 18(6), 2021. doi:10.1038/s41575-020-00411-3.

[5] K. Szafranska, L. Kruse, C. Holte, P. McCourt, and B. Zapotoczny. The wHole Story About Fenestrations in LSEC. *Frontiers in physiology*, 12, 2021. doi:10.3389/fphys.2021.735573.

[6] H. Mao, K. Szafranska, L. Kruse, C. Holte, D. L. Wolfson, B. S. Ahluwalia, C. B. Whitchurch, L. Cole, G. P. Lockwood, R. Diekmann, D. Le Couteur, V. C. Cogger, and P. A. G. McCourt. Effect of caffeine and other xanthines on liver sinusoidal endothelial cell ultrastructure. *Scientific reports*, 13(1), 2023. doi:10.1038/s41598-023-40227-0.

[7] B. Zapotoczny, K. Szafranska, M. Lekka, B. Singh Ahluwalia, and P. McCourt. Tuning of Liver Sieve: The Interplay between Actin and Myosin Regulatory Light Chain Regulates Fenestration Size and Number in Murine Liver Sinusoidal Endothelial Cells. *International Journal of Molecular Sciences*, 23(17), 2022. doi:10.3390/ijms23179850.

[8] W. Zhou, R. Apkarian, Z. Wang, and D. Joy. Fundamentals of Scanning

Electron Microscopy (SEM). In *Scanning Microscopy for Nanotechnology: Techniques and Applications*. 2006. doi:10.1007/978-0-387-39620-0_1.

[9] Carl Zeiss Microscopy GmbH. Product Information - ZEISS GeminiSEM Family. URL https://en.uit.no/Content/764267/cache=20221702125237/3_GeminiSEM_rel_3_0.pdf.

[10] nanoScience Instruments - Scanning electron microscopy. URL https://www.nanoscience.com/techniques/scanning-electron-microscopy/.

[11] K. Szafranska, C. Holte, L. Kruse, H. Mao, C. Øie, M. Szymonski, B. Zapotoczny, and P. McCourt. Quantitative analysis methods for studying fenestrations in liver sinusoidal endothelial cells. A comparative study. *Micron*, 150, 2021. doi:10.1016/j.micron.2021.103121.

[12] H. Geissinger, R. Vriend, L. Meade, C. Ackerley, and M. Bhatnagar. Osmium-Thiocarbohydrazide-Osmium versus Tannic Acid-Osmium Staining of Skeletal Muscle for Scanning Electron Microscopy and Correlative Microscopy. *Transactions of the American Microscopical Society*, 102(4), 1983. doi:10.2307/3225852.

[13] P. Li, J. Zhou, W. Li, H. Wu, J. Hu, Q. Ding, S. Lü, J. Pan, C. Zhang, N. Li, and M. Long. Characterizing liver sinusoidal endothelial cell fenestrae on soft substrates upon AFM imaging and deep learning. *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1864(12), 2020. doi:10.1016/j.bbagen.2020.129702.

[14] L. von Chamier, R. Laine, J. Jukkala, C. Spahn, D. Krentzel, M. Lerche E. Nehme, S. Hernández-pérez, P. Mattila, E. Karinou, S. Holden, A. Can Solak, A. Krull, T.-O. Buchholz, M. L Jones, L. Alain Royer, C. Leterrier, Y. Shechtman, F. Jug, M. Heilemann, G. Jacquemet, and R. Henriques. Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nature Communications*, 12(1), 2021. doi:10.1038/s41467-021-22518-0.

[15] X. Zhao, L. Wang, and Y. et al Zhang. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57, 2024. doi:10.1007/s10462-024-10721-6.

[16] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI 2015: 18th international conference*, 2015. doi:10.1007/978-3-319-24574-4_28.

[17] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[18] J. Deng, W. Dong, R. Socher, L. Li, L. Kai, and F.-F. Li. ImageNet: A large-scale hierarchical image database. In *2009 IEEE*

Conference on Computer Vision and Pattern Recognition, 2009. doi:10.1109/CVPR.2009.5206848.

[19] F.-F. Li. ImageNet Large Scale Visual Recognition Challenge 2012 - Introduction and overview of results. URL https://www.image-net.org/static_files/files/ilsvrc2012.pdf.

[20] ImageNet 2014 - Large Scale Visual Recognition Challenge 2014. URL https://www.image-net.org/challenges/LSVRC/2014/results.php.

[21] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations (ICLR)*, 2015. doi:10.48550/arXiv.1409.1556.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi:10.48550/arXiv.1512.03385.

[23] C. Kawatsu, F. Koss, A. Gillies, A. Zhao, J. Crossman, B. Purman, D. Stone, and D. Dahn. Gesture Recognition for Robotic Control Using Deep Learning. 2017. URL https://www.researchgate.net/publication/320084139_Gesture_Recognition_for_Robotic_Control_Using_Deep_Learning.

[24] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment Anything. 2023. doi:10.48550/arXiv.2304.02643.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. H., S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2021. doi:10.48550/arXiv.2010.11929.

[26] A. Archit, S. Nair, N. Khalid, P. Hilt, V. Rajashekar, M. Freitag, S. Gupta, A. Dengel, S. Ahmed, and C. Pape. Segment Anything for Microscopy. *bioRxiv*, 2023. doi:10.1101/2023.08.21.554208.

[27] S. Prasad, D. Lin, Y. Li, D. Sheng, and O. Min. Rethinking of Deep Models Parameters with Respect to Data Distribution. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021. doi:10.1109/ICPR48806.2021.9413003.

[28] C. Shorten and T. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6, 2019. doi:10.1186/s40537-019-0197-0.

[29] M. Maison, T. Lestari, and A. Luthfi. Retinal Blood Vessel Segmentation using Gaussian Filter. *Journal of Physics: Conference Series*, 1376, 2019. doi:/10.1088/1742-6596/1376/1/012023.

[30] OpenCV - Histograms - 2: Histogram Equalization. URL https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html.

[31] H.E. Kim, A. Cosa-Linan, and N. et al. Santhanam. Transfer learning for medical image classification: a literature review. *BMC Medical Imaging*, 22, 2022. doi:10.1186/s12880-022-00793-7.

[32] A. Ramdan, A. Heryana, A. Arisal, R. Kusumo, and H. Pardede. Transfer Learning and Fine-Tuning for Deep Learning-Based Tea Diseases Detection on Small Datasets. In *2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2020. doi:10.1109/ICRAMET51080.2020.9298575.

[33] J. Ma, J. Chen, M. Ng, R. Huang, Y. Li, C. Li, X. Yang, and A. L. Martel. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71, 2021. doi:10.1016/j.media.2021.102035.

[34] S. Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020. doi:10.1109/cibcb48159.2020.9277638.

[35] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. Dahl. On Empirical Comparisons of Optimizers for Deep Learning. 2020. doi:10.48550/arXiv.1910.05446.

[36] D. Müller, I. Soto-Rey, and F. Kramer. Towards a Guideline for Evaluation Metrics in Medical Image Segmentation. 2022. doi:10.48550/arXiv.2202.05273.

[37] K. Zou, S. Warfield, A. Bharatha, C. Tempany, M. Kaus, S. Haker, 3rd Wells, W., F. Jolesz, and R. Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. *Academic radiology*, 11(2), 2004. doi:10.1016/s1076-6332(03)00671-8.

Appendix A

Implementation details

The implemented solution can be found on GitHub: https://github.com/marketakvasova/LSEC_segmentation/tree/main. The repository contains a Python notebook intended for the model's use and the notebook with which the model was trained.

There is also a notebook, which was used to experiment with the automatic creation of whole cell masks. However, this was not very successful, as the images have very high resolution, and the tested model downscales them quite a lot. This results in losing a lot of information about the cell edges.

The Python notebooks can be opened and run in Google Colab. The GitHub page also includes instructions on how to run the segmentation script locally.

The training and testing data, and the model weights can be found at https://drive.google.com/drive/folders/18O8pFbqFLx34X1dliWbPf9EkqeFO0ASK.

Appendix B

Attachments

The thesis attachments include:

- **evaluation_masks**: A folder containing the segmentation masks used for the result evaluation in Chapter 5. This includes all of the ground truth masks, semiautomatic masks, and automatic masks. All of the masks were edited as described in Figure 4.6. The objects which are too small or too large to be a fenestration were removed from these masks and cell masks were applied to them. This means these masks contain only the fenestrations that were directly used for the method comparison.
- **mask_comparison**: Includes the comparison of all 22 test masks. The comparison is done by merging the ground truth masks with the automatic and semiautomatic masks. Examples of these merged masks are shown in Figures 5.6 and 5.7.
- **README**: A file describing the attachment contents.
- **LSEC_segmentation_training.ipynb**: The Colab script which was used for model training and evaluation.
- **LSEC_fenestration_segmentation.ipynb**: A script for using the trained model.

Training images, input testing images, and model weights were too large to include in the attachments. They can be found on Google Drive: https://drive.google.com/drive/folders/18O8pFbqFLx34X1dliWbPf9EkqeFO0ASK.

Appendix C

Used software

The implementation was written in Python 3.10 and is supposed to be run in Google Colab[1]. Libraries used to create, train, and compare models are: Pytorch[2], Weights & Biases[3], Segmentation models pytorch[4], Segment anything model [5], scikit-learn[6], albumentations[7], tqdm[8]. Libraries used to handle and visualize data are the following: jupyter_bbox_widget[9], matplotlib[10], seaborn[11], numpy[12], OpenCV[13], scipy[14]. ChatGPT[15] code suggestions were used during the writing of the code that handles graph plotting.

Manual mask correction was done in GIMP[16].

This thesis was written in Overleaf[17] and Grammarly[18] was used for grammar corrections.

[1] https://colab.research.google.com/

[2] https://pytorch.org/

[3] https://wandb.ai/site

[4] https://github.com/qubvel/segmentation_models.pytorch

[5] https://github.com/facebookresearch/segment-anything

[6] https://scikit-learn.org/stable/index.html

[7] https://albumentations.ai/

[8] https://tqdm.github.io/

[9] https://github.com/gereleth/jupyter-bbox-widget

[10] https://matplotlib.org/

[11] https://seaborn.pydata.org/

[12] https://numpy.org/

[13] https://opencv.org/

[14] https://scipy.org/

[15] https://openai.com/index/chatgpt

[16] https://www.gimp.org/

[17] https://www.overleaf.com/

[18] https://www.grammarly.com/