Tampere University

Hannu Hakkola

# Modeling Single Cell Properties from Histological Images

Faculty of Medicine and Health Technology (MET)
Master's thesis
May 3, 2021

# Abstract

Hannu Hakkola: Modeling Single Cell Properties from Histological Images
Master's thesis
Tampere University
Master's Degree Programme in Bioinformatics
May 3, 2021

---

In modern pathology, digitized images of histological sections are routinely used in defining phenotypes and characteristics for different areas of the tissue. Digital images are created from stained histological slides by specialized scanners and are further analyzed with a computer. Traditionally this type of digital pathology analysis is limited to analyzing the tissue section in local patches or in the sub-sampled section level. We propose a novel approach derived from methodology in precision digital pathology and network analysis to study the single-cell level local neighborhoods of the tissue while preserving spatial information in the form of network connections. We show that our tool can successfully be used in advanced and precise assessment of local properties combining multiple stainings and further apply this method to a multi-stained histological mouse aortic root dataset.

**Keywords:** precision digital pathology, digital pathology, network analysis, deep learning, quantitative histology.

# Contents

# 1 Introduction

Digital pathology concerns the examination and analysis of digitized histological tissue sections. As in traditional pathology, small tissue samples are first embedded in formalin or they are frozen, which enables cutting the tissue in thin sections. These thin sections are then embedded in glass slides that can be examined with a microscope. The digitized images are derived by imaging the glass slides with a specialized scanner, resulting in digital whole slide images (WSI) that are of high pixel resolution. The high resolution (and thus file size) commonly limits the analysis of these images in either local patch level preceded by tiling the image into small sub-sections or in global section level which requires sub-sampling the original image.

Usually, the different local patterns of interest in the image are differentiated with different histological stainings. A common staining composed of hematoxylin and eosin (later H&E), for example, colors the cell organelles with shades of blue and red based on whether the organelle is basophilic or acidophilic, respectively[1]. Commonly, H&E is combined with immunohistochemical (IHC) stainings which can be produced to describe the binding of specific antigens. The local tissue properties highlighted by the different stainings are then used in further analysis, for example in detecting cancerous regions in the tissue.

Depending on the research objective, the analysis process may be aided by various computational approaches. Machine learning, especially that of which considers the use of deep neural networks and cluster computing, has enabled the automation of routine histological analysis tasks, for example the detection of cancerous regions in constantly increasing precision with at times even outperforming pathologists [2][3][4][5].

By studying the tissue histology with traditional digital pathology approaches, the spatial distribution of cells across the tissue is often ignored. This distribution entails a biological and practical significance and is far from being uniform. Relative cell locations alternate between denser and more sparse regions which correspond to different biological phenomenon and thus different importance regarding the research objective. For example, cell signaling and interactions can be closely related to the proximity of the cells involved if being contact-dependent or paracrine types of signaling[6].

The natural way of modeling the spatial cell distribution and interactions would be to present the cells as parts of a network. Modeling the cells this way would resemble the biological circumstances in the tissue in a greater precision than the traditional approach. Further, this kind of a model would open many novel subjects of research: the increasing amount of organized digital information related to social

networks as a result of the growing social media industry has resulted in advances of network analysis utilizing deep learning methods, such as [7] and [8]. These advances include for example the prediction of node properties, e.g. social influence from a social network[9]. In biological context, this could mean for example predicting the relative importance of local tissue areas related to a disease.

By combining methodology from deep learning based precision digital pathology and network analysis, we propose a novel method for analyzing single-cell level local neighborhoods in digital histological images utilizing multiple histological stainings. Our method is based on a five-stage spatial analysis pipeline which utilizes various machine learning and image analysis algorithms. It creates a network model for single cells spanning multiple histological stainings and can be used in precision digital pathology analysis in defining local environments for different objects of interest. Not easily seen with the human eye, the network of neighboring cells may reveal novel properties of the tissue and reveal important similarities between different tissue regions when combined with features extracted with different histological and other imaging methods.

We demonstrate the performance of our tool in a dataset consisting of 3 different sections of the root of the aorta of *Mus Musculus*, each stained with 4 different stainings, totaling in 12 images. The dataset contains information about macrophage locations across the root of the aorta, which in the case of the dataset, are used in studying the early development of coronary artery disease (CAD). We show that our tool can be successfully used in precise data exploration and in discovering novel local properties of tissues.

This type single-cell analysis in imaging, even when the cell locations are to be approximated, could also be used together with other single-cell and/or spatial analyses. If, for example, combined with spatial transcriptomics methods[10][11], running our pipeline could work as an external validation step.

# 2 Literature review

This chapter provides a more comprehensive and general theoretical background for the methodology used in developing the spatial analysis pipeline. We look at the history and trends of machine learning and image analysis technologies related to digital histopathology, their practical implementation and future directions. A more detailed overview of the various methods and algorithms used in the proposed pipeline is provided later in Chapter 3.

## 2.1 Machine learning

Our pipeline relies heavily on multiple image analysis algorithms which are in many cases powered by machine learning. In biomedical research, the development and application of computer aided image analysis dates back to at least the sixties[12]. The simultaneous increase in available computing power and the development of special scanners capable of whole slide imaging has enabled various novel research opportunities. To understand the core components of the rather complex task of our pipeline, we need to look at the underlying trends that have enabled the success of machine learning in biomedical image analysis.

The term *machine learning* was popularized by Arthur Samuels in the late 50's[13]. A machine is said to learn if instead of being explicitly programmed to perform better in a certain task, it improves its performance in said task by experience. This definition was formalized by Tom Mitchell[14]. Machine learning has been applied successfully to multitude of problems. From natural language processing to enhancing audio, it has shown repeatedly to be able to solve problems that are difficult to show with an explicit algorithm.

### 2.1.1 Various types of machine learning problems

The term *supervised learning* refers to a subset of machine learning problems that have a collection of predefined independent variables, or inputs, that are used in predicting the values of outputs (dependent variables)[15]. The input variable selection and the measurement of the input variable is a crucial component in machine learning. If the goal is to recognize handwritten digits from grayscale images, an optimal input dataset would be a one that contains large amounts of examples from all of the digits. It is also important that the there are roughly equal amount of examples of all of the digits, resulting in a *balanced dataset*.

In contrast, in *unsupervised learning* there are no predefined sets of matching inputs and outputs. The goal in this type of machine learning is to infer relevant

properties of some distribution without supervision[15]. Often the goal is to find some kind of *clustering* in the input dataset and find a meaningful way of categorizing the data into different groups[15]. An example of an unsupervised learning would be topic modeling, where the goal is to divide an uncategorized set of input texts into a more abstract groups based on an inferred topic of the texts. In an unsupervised learning model, the topics are not predefined. An example of this kind of topic modeling would be the popular topic modeling algorithm called Latent dirichlet allocation[16]. In this thesis, we are mostly concerned with supervised learning problems. Therefore, the various types of unsupervised learning methods are not discussed in detail.

Supervised machine learning problems can further be categorized in two general groups: there are *regression problems* and *classification problems*. These groups vary by the nature of the output variables[15]. For example, the digit classification problem introduced earlier is a classification problem where each input training example is assigned a qualitative class (a digit). If the problem was inversed in a way that instead of recognizing handwritten digits, we wanted to train a machine learning model to *produce* these digits, we would have a regression problem. In that case, the output measurement is a quantitative variable and there is a way of measuring *how* realistic the machine-created digits are[15].

### 2.1.2 Common challenges in machine learning

There are various types of machine learning problems, ranging from simple to complex. The problems vary in difficulty, and there are multitude ways of attempting to solve these problems, as machine learning is mostly about approximations in contrast with finding an exact solution to a problem. Almost all of these problems have got a few common challenges that need to be addressed.

#### Problems considering the data

Every real-world dataset comes with a degree of uncertainty in them. Image datasets may contain unclear images, categorical datasets may contain outliers, time series datasets may contain temporal noise and so on. Even the annotations of the dataset may contain mistakes in the first place. Further, as stated earlier, an optimal dataset is balanced. For example, if in a multiclass classification problem, 90% of the input data belong to a single class, the problem becomes more challenging. Various methods exist to address these challenges: this is referred as *data cleaning* and *data preprocessing*[17].

#### Model selection

As there are various types of machine learning problems, there are also a multitude of ways to solve them. A comprehensive review of machine learning algorithms and their use cases is beyond the scope of this thesis. There are, however, a few general factors that contribute to the suitability of a given model to a particular task.

Firstly, machine learning models vary by complexity. A more complex model may have a larger computational cost when compared to a simple model, but they may also perform better in the given task. Usually, an increase of complexity also means a decrease in interpretability. For example, in the case of tree-based machine learning models, a single decision tree is easily interpretable. This means that we can easily see the factors that contributed to a decision made by the model. A more complex tree-based model, in contrast, needs an external way of interpretation. Still, we can calculate relative feature importances from a more complex random forest model which illustrate the relative importances of the input variables used in making a decision[15].

Secondly, in choosing a model, we have to address the bias-variance trade-off. A model with a small bias will produce correct results *on average*, but may have a large variance in its predictions. This, among other challenges, is also generally linked to model complexity[15].

#### Generality of the solution

A key challenge in machine learning is the generality of the proposed machine learning application. In other words, does the distribution that the model learned from the training data accurately resemble the "true distribution" that we want the model to learn? To test the generality of the model, an input dataset is often divided to *training dataset* and *testing dataset*. The model is trained using the training data and evaluated using the testing data. Note that even after the train-test split, we still have to assume that the input data correctly resembles the true distribution we want the model to learn. When a machine learning model performs well with the input data but does not generalize, it is *overfitted*[18].

### 2.1.3 Deep learning

Machine learning remains a rapidly progressing field of research. The recent growth and progress of machine learning can be attributed to increase in computing power, data storage capability and data availability. Modern smartphone and smartphone-related service business models are almost always at least partly based on data collection as smartphones (and for example, services on the cloud infrastructure) enable individualized data collection and simultaneous market for machine learning

based individualized services[19]. Many of these services utilize a specific subcategory of machine learning that has gained a lot of attention during the recent years: *deep learning*.

When compared to many traditional machine learning methods that require careful, context-specific feature engineering, deep learning has the advantage of learning the relevant features from the data itself. This process is described with the term *representation learning*[20]. A deep neural network creates complex representations of the original input data through a series of simple non-linear operations in so-called neurons, which enable learning abstractions of complex functions[21]. A single neuron has an *activation*, a *weight* that is analogous to the strength of the signal (the activation is multiplied by the weight) for the given neuron, and a *bias* that is analogous to being a threshold that needs to be exceeded for the neuron to "fire" (or to contribute to the total sum of signals for the consecutive neuron). These definitions are obviously highly simplified and incomplete - a thorough explanation is beyond the scope of this literature review. The terms *neuron*, *activation*, *weight* and *bias* are however core concepts in deep learning and they are used later in the text.

In the case of a simple classifier, the neurons are stacked in consecutive layers that can be thought of advancing in abstraction: the first layer is given the raw input data (although, as stated in [20], this data is often pre-processed in various ways), the second layer a slightly more complex representation of the input data, leading to the highest form of abstraction in the output layer (i.e. the different output classes). All of the layers that are not input or output layers are called *hidden layers*. The consecutive representations of the layers are learned in parallel utilizing an algorithm called *backpropagation*[22].

Backpropagation, in simplified terms, works by averaging the most relevant proposed differences in the weights and biases of all of the neurons for all of the training examples (although, as this is computationally expensive, a randomized set of mini-batches is often used instead) defined by a *cost function*. These changes are analogous to moving in a direction towards some local minimum in the gradient of the cost function (with the mini-batches, the direction is an approximate one).

The cost function should accurately describe what we want the network to learn. A simple classifier, that is given 10 input variables and 3 output classes, can be trained by calculating a mean squared error of the output vector given by the network and a vector where all of the wrong classes have a zero value and the right class has a one. Backpropagation, by trying to minimize the cost function, then slowly changes the parameters of the neural network until a local minimum is reached. The definition of a relevant cost function is therefore a key component in successfully training a neural network[22].

### 2.1.4 Convolutional neural networks

Deep neural networks excel in many machine learning tasks. A classifier like the one described in the previous paragraph still has a relatively simple task (from a purely computational perspective) to perform. With neural networks, the computational load quickly increases when images are used as an input. Computerized image data is commonly represented as a two-dimensional matrix of values. Depending on the representation, a colorized image may be constructed as a matrix of values pointing to a color palette indices or, for example, a series of matrices corresponding to the red, blue and green channels (RGB). The channels can then be combined to represent all the colors in the range allowed by the number of bits used in storing the color values. In an RGB image, the smallest units of the image (that is, pixels) have a location somewhere in the matrix and 3 values corresponding to the 3 channels. Therefore, even a simple $32 \times 32$ RGB image has got $32 \times 32 \times 3 = 3072$ input values (and if the image is given as a whole to the network, 3072 input neurons). This often renders the use of fully connected neural networks (FCN) impractical because of the computational cost.

Even though there has previously been various computationally efficient solutions (compared to FCNs) for neural network image analysis, the most commonly used solutions nowadays rely on convolutional neural networks, or CNNs. Modern CNNs are originally based on the research of LeCun et al., where a neural network was successfully trained to classify handwritten ZIP postal codes[23]. A convolutional neural network has filters, defined by width, height and the number of channels (or the depth) that slide across the previous representation of the input data (the first being the original input image) and thus processing only a small spatial subset of the input data at one time. This creates feature representations of increasing complexity through the network. The parameters of the filters (or kernels) are learned through backpropagation[23]. The convolutional layers are used together with fully connected layers to provide, for example in the case of a digit recognition problem, the final 10 output neurons that represent the digits 0-9. Many recent successful medical image analysis applications rely on CNNs[24].

### 2.1.5 Machine learning with Python

In practice, most machine learning applications require processing large amounts of data, which is then analyzed in algorithms of varying computational complexity. At the same time, iteration is often necessary in the construction and development of a machine learning pipeline. Therefore, a language like Python that is a high-level language suitable for quick prototyping but is often used with more efficient programming language (C/C++) library bindings is a natural choice for machine learning

development. Some common Python machine learning libraries are Scikit-learn[25] for general machine learning development and TensorFlow[26] and PyTorch[27] for deep learning development. Both of TensorFlow and PyTorch depend on a highly optimized multidimensional array computing libraries. In both of the implementations, the libraries are written to take advantage of the modern GPU systems capable of highly parallelized and efficient computing[26][27]. In our pipeline, we use both Scikit-learn and TensorFlow (through the Keras API) in different pipeline components.

The main challenge in using Python for machine learning development is that pure Python is not built for the type of computing that the computationally intensive machine learning applications require. For example, its global interpreter lock prohibits true multithreading (even though there are ways to circumvent this limitation)[27]. The popularity of Python among developers is largely due to its user-friendliness and relatively easy learning curve. In a development task like the pipeline introduced in this thesis, the researcher(s) already need domain-specific knowledge in machine learning and digital pathology: expertise in efficient computing is yet another domain to learn.

## 2.2 Digital pathology

To even consider applying any computerized quantitative operation on histological images, there needs to be a way of representing tissue sections in a digital format. Traditionally, histopathological analysis was conducted by observing tissue sections in glass slides through a microscope. However, the development of a WSI scanner, first described in 1999[28], enabled the digitization of histological images.

A WSI scanner consists of light source, slide stage, objective lenses and a camera[29]. The capturing of the tissue images is done in parts: the scanner captures either tiles or lines of the whole tissue section that are then digitally connected to each other. Depending on the scanner, the imaging may be fluorescent, brightfield or multispectral. Fluorescent imaging can be used in detecting special fluorescently labeled slides, while the brightfield imaging corresponds to the standard brightfield microscopy. Multispectral imaging can be applied to both of the imaging method, and it is used in capturing spectral information of the light[29].

The scanners also vary in their focusing strategies and possible levels of magnification as well as performance[29]. Depending on the scanner and the specifications of the scanning (such as magnification and color bit depth), the resulting images also vary in file size. Generally, for viewing and interpreting H&E or IHC stained (the stainings are described in detail in Chapter 3) images, X20 magnification will result in an appropriate resolution[29].

Many of the resulting digitized images contain a large amount of information.

This proposes challenges in the processing and storage of these images. The images are often compressed and require specialized programs for viewing and manipulation. Besides compression, limiting the resolution or the scope of view when viewing the images can be used as a means to ease the processing of large WSI images[29].

The resolution of the WSI images is often described by the micrometers per pixel ratio. A typical ratio for an X40 WSI image would be $\frac{0.25\mu m}{pixel}$, which may result in a file size of over 1GB for a single WSI image[29]. A typical format for storing these large is images is the JPEG2000 format. JPEG2000 is a popular lossy compression method, which means that information is lost when an image is compressed using this specification[30]. The format, however, has other potential benefits, as it enables random access in the image meaning that a block of the image can be decoded for viewing without having to decode the whole area of the image[30].

### 2.2.1 Applications of digital pathology

The digitization of histological images (in addition to machine learning applications) has various benefits. When stored digitally, the images can be shared between experts around the world, which enables remote diagnostics, or telepathology. Historically, the general adoption of telepathology has been slow due to limitations in bandwidth and storage capabilities[29]. When combined with the latest technologies, telepathology offers various possibilities. A famous study by Google researchers combined augmented reality technology and machine learning to produce a real-time diagnosis through a microscope[31].

Another interesting application of the digitization of histological images is the 3D reconstruction of the original tissue. The 3D reconstructions are composed of consecutive histological images in a process called *registration*, that is also applied in this thesis. Registration considers various translations and modifications performed to the images in order to align them on top of each other for the 3D reconstruction. Various algorithms exist for registering the images[32]. They vary by complexity, number of tunable parameters and the modifications that they perform to the images. The algorithms may use affine transforms, deformable or elastic transforms or some combination of these methods. As the WSI images that are used in registration are large, the registration process is often time-consuming, especially because it is often an iterative one[32].

The resulting 3D reconstructions obtained from the registration of the images have multiple use cases. While the reconstructions themselves provide a novel view for histopathological analysis and can be used as is in education, they are not limited to direct computer examination. The reconstructions may be used together with, for example, virtual reality and 3D printing technologies. While the current obvious applications may be limited to educational purposes, these technologies provide a

promising new perspective in digital pathology[33][34].

### 2.2.2 Image analysis

A core task in analyzing histological images is the segmentation of cells and nuclei from an image. There exists various algorithms for this task, ranging from a simple intensity thresholding to a deep neural network. The applicability of an algorithm depends at least on the data and available computational resources[35]. If an image contains only relatively uniformly colored cells with clear boundaries, a successful cell segmentation can likely be performed even with a simple, deterministic algorithm. However, the input images may contain many unwanted regions and artifacts that interfere with the detection. Further, the cells and the nuclei are far from being uniformly shaped and colored[35]. A general algorithm capable of detecting nuclei from any staining is extremely difficult to write, as the input data will contain large amounts of variance. In this thesis, we rely on a solution introduced in [36] as our dataset contains images with multiple histological stainings. Even for a deep neural network, the task proves to be difficult, as discussed later.

### 2.2.3 Machine learning in digital pathology

The success of machine learning based image analysis has accelerated its adoption to digital pathology. Machine learning has enabled the automation of even more abstract tasks: not only are we able to robustly detect nuclei from the images - we can also automate the detection of cancer in the same level as an expert pathologist[24]. Recently, machine learning applications have been used primarily either as an aid for diagnosis and prognosis or identifying novel features about a disease[24]. Recall that this resembles the distinction we made earlier between supervised and unsupervised learning. In the case of supervised learning, we have matching inputs (WSI images) and outputs (classes of healthy and cancerous samples). In unsupervised learning, we want to find properties of the underlying distribution without supervision. In this case, we want to identify novel properties from a disease.

For a machine learning model, the problems in digital pathology can be, in their most simple form, binary classification problems (i.e. does the tissue in this image contain cancer or not). A more complex problem would be the grading of a cancerous tissue by Gleason grading[24], which resembles a multi-class classification problem. The requirements of this kind of research, besides computational resources, include a large, expertly annotated dataset. The availability of quality data is still considered to be one of the key challenges needed to overcome for clinical adoption of machine learning[37].

### 2.2.4 Challenges in the clinical adoption of machine learning

The adoption of machine learning techniques into clinical use depends on various factors. As stated earlier, quality of the data is a key limiting factor in machine learning based digital pathology. As many of the state-of-the-art machine learning solutions in digital pathology rely on deep neural networks, another problem arises: interpretability[37]. Recall that machine learning models have degrees of complexity which result in different degrees of interpretability of the models. A deep neural network is a complex model that is, in general, hard to interpret. While there exists some methods in interpreting the functionality of a deep neural network, its learning still remains analogous to a black box. Visual attention maps can help in determining some of the areas that the network uses as a base of making a decision, but the core problem prevails: we cannot make sure that the network is learning biologically relevant features from the data and not simply *overfitting* the model on the available training data. After all, a model used in clinic should be a maximally generalizable one.

For example, it has been shown that CNNs, when trained with histological images, have a varying performance when other parameters are modified in the research setting. As stated earlier, the different WSI scanners may have slightly different focus strategies in the imaging process that may result in variation in the input data across scanners. Different research settings provide also other types of variation into the process, which results in data that may have systematic differences between datasets collected in different institutions. While these differences may not be even clearly visible for a human, the performance of a CNN may vary drastically across these datasets[37]. To address this problem, it is proposed that the models need to be properly validated using data from various sources and scanners before adopting the models to the clinic[37].

Further, undesirable behavior of a machine learning model can be also achieved intentionally. A model designed for clinical use can be trained with other intents than purely the desire to provide better care and thus many undesirable biases can be built into a machine learning model[38]. The adoption of machine learning has already spiked controversy when implemented in other public affairs, such as jurisdiction[38].

While there are many areas in healthcare that can lead to unethical practices, machine learning proposes two key ethical challenges: transparency and responsibility. The problem of building an unethical machine learning model is that the unethical biases can not be easily verified: machine learning almost always introduces a degree of opacity into the process. And even if a machine learning model was somehow verified to be completely unbiased, it can still make mistakes. A large

scale adoption of machine learning into the clinic would inevitably lead to a loss of lives because of wrong predictions made by a machine learning model. In a traditional health care system, there are already strong ethical foundations in a case of mistreatment, but this is not yet the case with machine learning[38]. When the goal of implementing machine learning based clinical applications is to provide better healthcare, the ethical side cannot be ignored.

### 2.2.5 Future trends of digital pathology and the relevance of this thesis

As discussed in previous sections, digital pathology has enabled completely novel research opportunities and perspectives, especially when combined with machine learning. In this section, we look at the possible future trends in digital pathology and a few approaches that have similar goals to the pipeline proposed in this thesis.

While the focus in this chapter has been mainly on the development of digital pathology and machine learning, the collection and computational analysis of other forms of biological data has also progressed in parallel. One of the key trends in the future of digital pathology is its combination with other fields of biological research. As a concrete example, histological images can be combined with various types of other biological information. For example, combining histological images with molecular (i.e. proteomic, transcriptomic) information can be used in understanding the different microenvironments in the tissue[39].

The successful combination and utilization of the data from various sources is a key challenge in the future of digital pathology (and more broadly, computational biology). In a novel study[40], histological information was combined with machine learning and spatial transcriptomics. Spatial transcriptomics in simplified terms, refers to a novel technique with which spatial information can be coupled with gene expression profiles. In other words, gene expression can be analyzed in spatial context. In the study, deep learning technique based on convolutional neural networks was used in predicting spatial gene expression from histological images stained with the H&E stain[40]. Similar studies integrating previously independent biological data sources are likely to be conducted in the future.

Combining other biological data with histological images will increase the precision with which we can analyze the different local environments found in the studied tissue.As we gain more precise spatial information from other biological fields, the demand for increasing precision in analyzing histological images increases as well. The aim of the pipeline proposed in this thesis is to differentiate and model single cell level regions from histological images using various image analysis and machine learning techniques. In the future, the combination of increasingly precise histolog-

ical image analysis and molecular information should increase our understanding of the underlying biological conditions in a truly interdisciplinary way.

# 3 Materials and methods

The pipeline can be divided roughly in five distinct phases: image registration, data post-processing, cell area approximation, staining-specific local feature extraction and assignment and network analysis. In Figure 3.1 the full analysis pipeline is illustrated as a series of distinct operations. The dataset used here as a case example is a mouse aorta root histological dataset that is described in detail in Section 3.1.

## 3.1 Data

Frozen histological sections of the root of the aorta of *Mus Musculus* were digitized with a slide scanner (Pannoramic 250 Flash). The dataset consists of sub-sampled histological images stained with Movat staining and various immunohistochemical stainings explained in detail in Table 3.1. The aspect ratio of the images was approximately 1:1 with the average pixel resolution being circa $1890 \times 1890$. The directory structure of the dataset is specified in Figure 3.2. This dataset, originally used in[41], contains information about type M1 and M2 macrophage locations across the root of the aorta. Generally, macrophages are large leukocytes specializing in digesting various type of pathogens in a process called phagocytosis. The two main subgroups, M1 and M2, act as a pro-inflammatory pathogen eliminating cells and cells associated with tissue repair and wound healing, respectively[42]. In the case of this dataset, the macrophages were studied as an indicator of atherosclerotic plaque related to coronary artery disease[41]. In addition to targeting macrophages, the dataset contains sections stained with Movat staining, providing information about the general tissue structure[43]. In detail, the Movat staining is described in Table 3.2. All of the macrophage-targeting IHC stainings used here contain hematoxylin as a background or contrast stain (blue). The targets are colored brown.

## 3.2 Multi-stain image registration for preserving spatial features

To preserve the spatial relations between consecutive histological section, the images need to be aligned with each other in process called registration. The ideal way of doing this is to use an automatic algorithm, such as [44]. In practice, a universal working registration algorithm is yet to be developed (this is discussed in detail in Section 5.3.1), especially for a dataset containing multiple histological stainings. As our case was particularly challenging for an automatic algorithm, we resorted to manually registering the images with GNU Image Manipulation Software (GIMP)[45]. In this case, the input images were small enough in terms of pixel

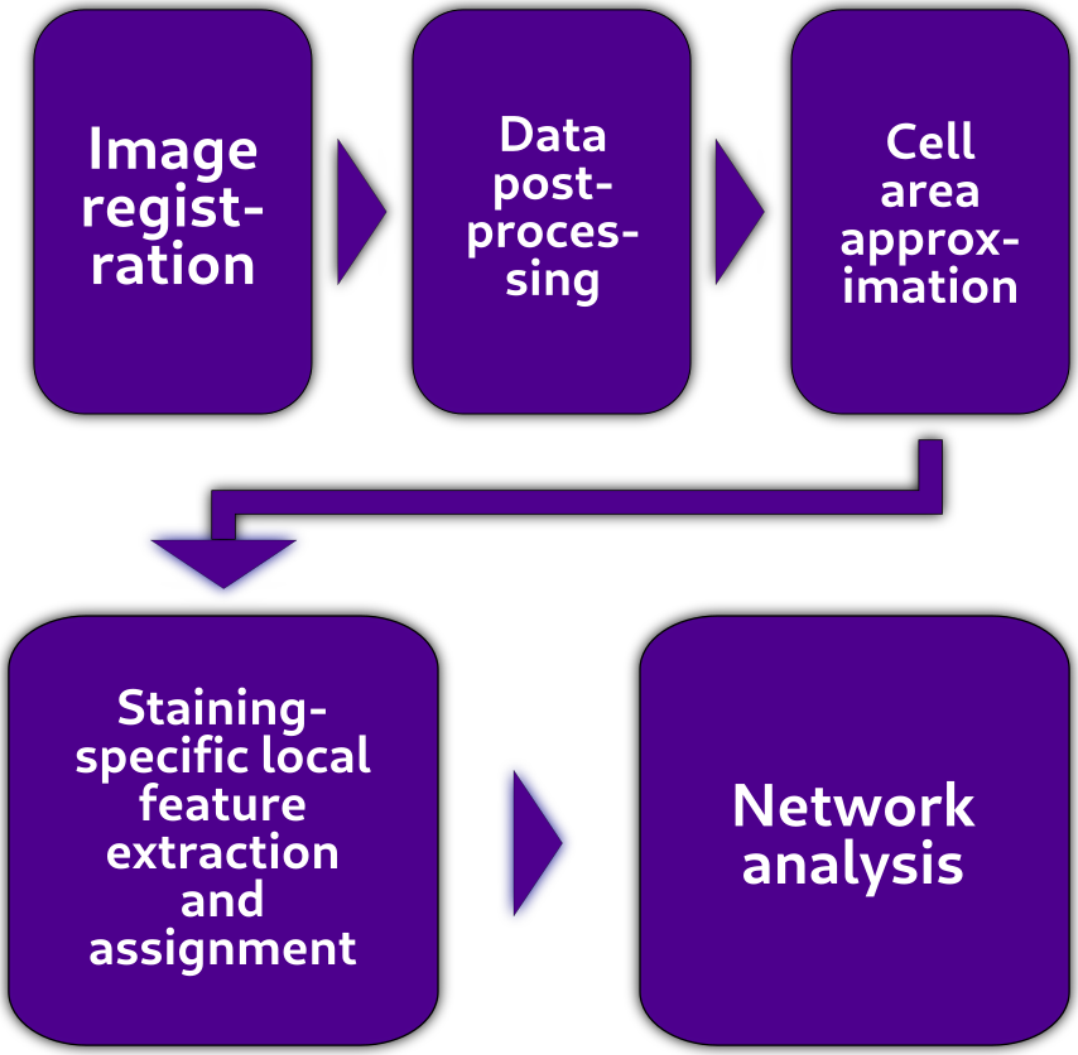


***Figure 3.1*** *The full pipeline in an abstract level. Single cell areas are approximated from series of consecutive registered histological sections and local features are extracted from these areas. This enables the use of network analysis methods to neighboring cells utilizing local features from multiple stainings*

resolution, which makes it possible to register the images manually.

Ideal manual registration is defined here as the maximum of supposed registration function $G_r$, that is a combination of moving, rotating and affinely transforming the two input images $X$ and $Y$ such that we maximize the overlap between supposedly continuous local tissue structures ($F_l$) and minimize the area in which one of the input images contains tissue structures and the other one contains only background ($F_b$) while preserving a suitable degree $k$ of local structural similarity ($F_s$) in the context of similar structures (i.e. avoiding too extreme modifications of local components). $F_s$ therefore describes the process of observing local tissue components of certain type (i.e. nuclei) and evaluating the possible modifications in the

## Staining specification

| Name | Description |
|---|---|
| Movat | Standard general histological staining, targets various cellular structures |
| Mac-3 | IHC (immunohistochemical) staining, targets all macrophages |
| iNOS | IHC staining, targets type M1 macrophages |
| MRC1 | IHC staining, targets type M2 macrophages |

***Table 3.1*** *Dataset staining specification*

## Movat staining

| Target | Color |
|---|---|
| nuclei | black to bluish-grey |
| cytoplasm | red |
| elastic fibers | red |
| collagen fibers | yellow (coarse fibers red) |
| cartilage | red or yellow |
| calcified cartilage | sea green |
| osteoid | red |
| mineralized bone | yellow |

***Table 3.2*** *Movat staining specification*

context of the phenotypes of these components (i.e. not making a modification that stretches a nuclei to cover half of the tissue area) Without preserving a degree of local contextual structural similarity, the registration would be a process of forcing two images into a single shape. By maximizing local contextual structural similarity, the images could only be moved and rotated (not stretched) as it would mean that the size of some local structures would be altered. In reality we want to avoid too extreme alterations for the input images while aiming for a maximally uniform tissue shape. This process is described more formally in Equation 3.1 and further illustrated in Figure 3.3.

$$\max G_r(X,Y) = \max F_l(X,Y) - \min F_b(X,Y) + k \times \max F_s(X,Y) \tag{3.1}$$

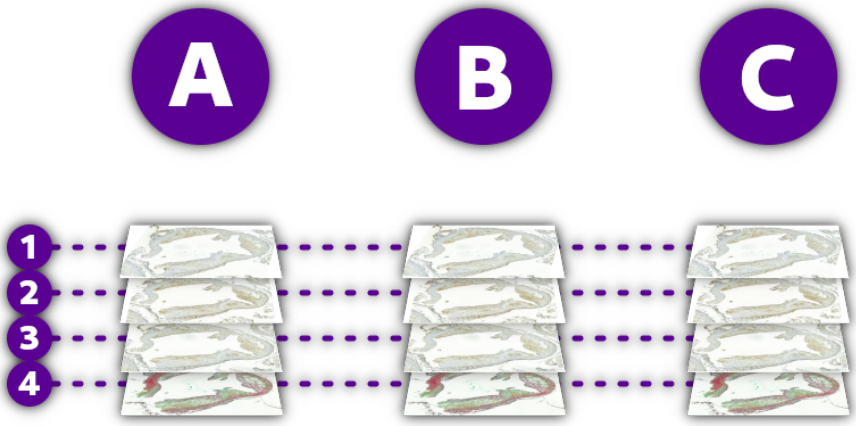


***Figure 3.2*** *Structure of the dataset. The data consists of 3 section stacks (A, B and C) which include 4 sections stained with 4 different stainings: Mac-3, iNOS, MRC1 and Movat stainings (1-4) totaling in 12 images.*

## 3.3 Approximating single cell areas from histological images

To model single cells as a network, the cell locations need to be defined. We define an approximate cell area by detecting nuclei locations from a reference staining and applying Voronoi decomposition to these locations. In detail, the process consists of four steps:

1. Splitting the registered images from the previous step to tiles
2. Defining a reference staining (Movat) for cell segmentation
3. Detecting the nuclei from all of the images with a method based on [36] utilizing domain adapted convolutional neural networks
4. Applying a Voronoi decomposition method for the detected nuclei such that a cell area is defined as all of the points closest to a given nuclei

The registered images were split in to tiles sized $224 \times 224$ pixels each or smaller if the tile was near the border of the image and the original image could not be split evenly to $224 \times 224$ sized tiles. The tiling here was performed to limit the memory consumption and does not affect the pipeline steps following this procedure. Further, the resulting nuclei detected were mapped back to a matrix that has the original image dimensions.

A reference staining is needed in the Voronoi approximation process as nuclei locations from multiple images are to be combined inside approximated cell areas.

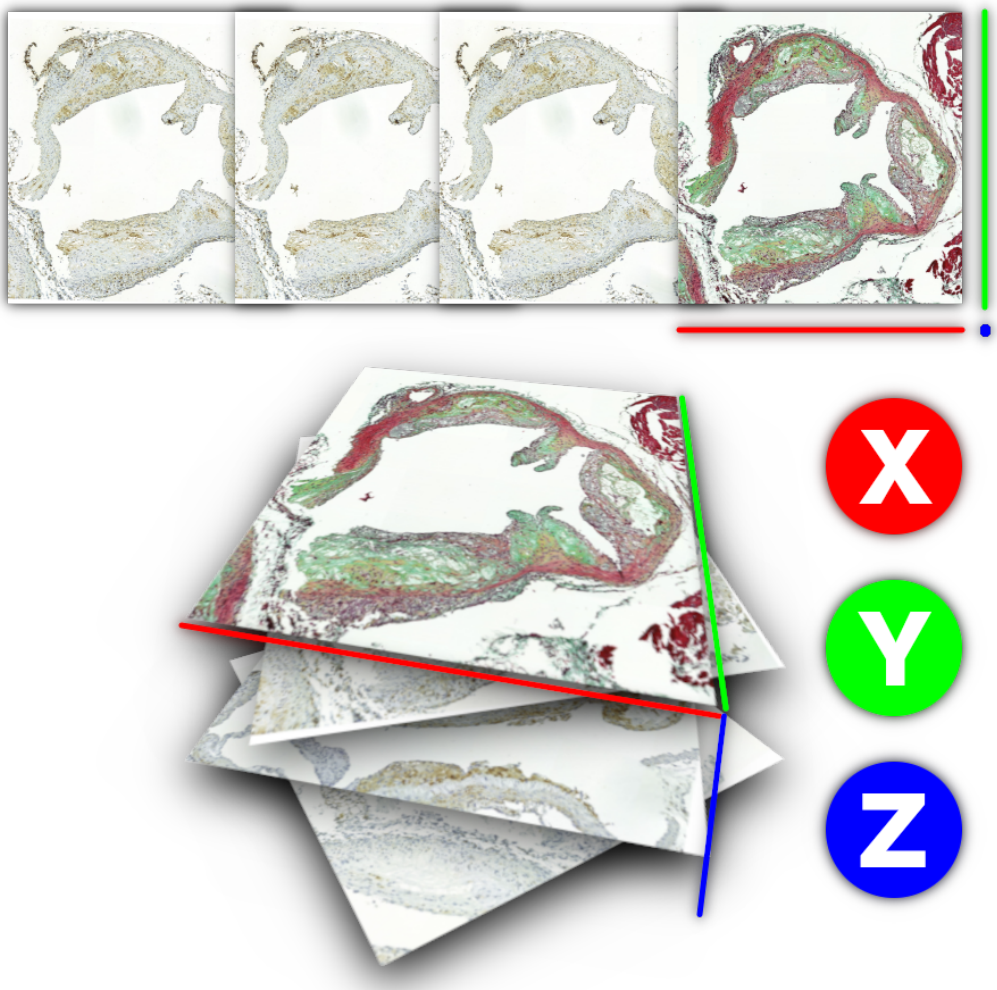


***Figure 3.3*** *Multi-stain registration: different images are aligned to each other with a two-step process to preserve tissue-scale spatial features with maximum accuracy. The X, Y and Z labels and their corresponding colors (red, green and blue, respectively) represent the 3 dimensions of the Z-stack.*

In other words, nuclei detected from the reference staining is used in forming the approximated cell areas to which nuclei detected from other stainings are assigned based on their location.

As in [36], the convolutional neural network was trained in a domain adaptation step[46] with our dataset, e.g. the same data that was later used in predicting the cell locations. Given baseline training input dataset $X_b$ containing manually annotated nuclei (target $Y_b$), a convolutional neural network $C_b$ is trained to output nuclei annotations such that $C_b(X_b) \approx Y_b$. The iterative domain adaptation is performed by training the baseline model with the dataset obtained by first predicting the nuclei locations from a novel domain target dataset $X_d$, resulting in a set of new

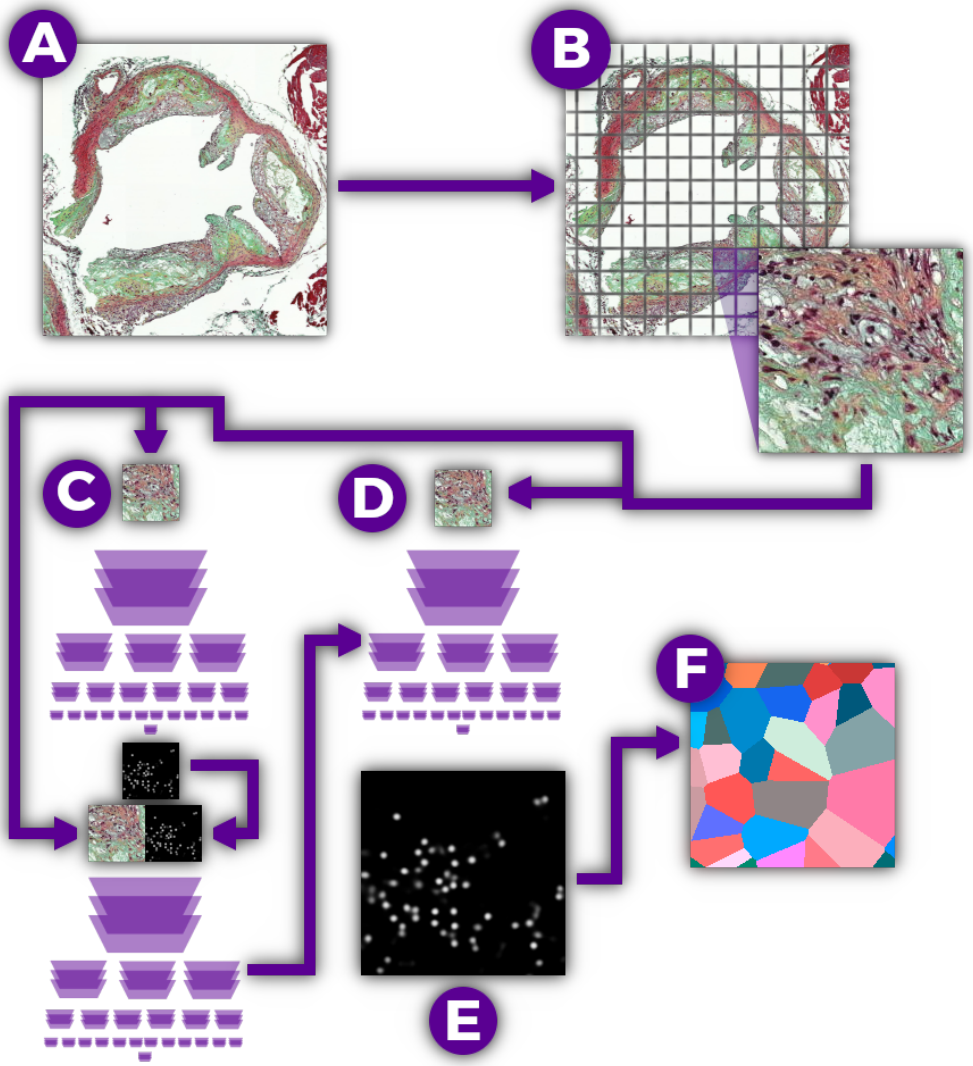


***Figure 3.4*** *Cell area approximation. Input images (A) are first tiled into approximately equal tiles (B) that are then used as input for domain adaptation step (C), where a pre-trained classifier is trained utilizing its own predictions. Using the domain adapted classifier (D), nuclei locations are then predicted from the tiles (E). From these locations, cell areas are approximated utilizing Voronoi approximation (F)*

target data $Y_d = C_b(X_d)$ for a new domain adapted classifier $C_d$. This process can be repeated iteratively. In our case, the model was trained with H&E stained histological sections and we performed the domain adaptation step with all of our images, having four different stainings.

After detecting the nuclei, Voronoi decomposition was performed for the nuclei locations such that the locations of the detected nuclei were stored globally from all of the tiles ensuring that the decomposition was applied to the whole tissue and not to an area of a single tile. The whole cell area approximation process is presented in Figure 3.4.

## 3.4 Semi-automatic segmentation of staining-specific features

The next step in the pipeline is to assign relevant features to the detected cells. In our case, we used features that are directly derived from the staining colors. This is done by annotating small areas of the tissue having the target color and performing a machine learning based automatic segmentation. The resulting segmentation map can be then directly applied to the detected cell locations.

Staining-specific features were extracted from all of the images utilizing a semi-automatic segmentation framework based on a Plotly/Dash[47] application titled "Interactive Machine Learning: Image Segmentation". The segmentation is based on comparing the local features from small manually annotated regions to the whole image utilizing a RandomForestClassifier in Python library Sci-Kit Learn[25]. In contrast to the original random forest algorithm[48], this implementation averages the prediction probabilities across classifiers instead of making a majority vote. The segmentation step is illustrated in Figure 3.5.

The segmentation is performed by applying multiple Gaussian filters on the image and varying the standard deviation ($\sigma$) parameter of the filter, as demonstrated in Section 3.4.1, and, consequently, calculating the pixel intensity, edges utilizing the Sobel operator[49] (Section 3.4.2) and Hessian matrix eigenvalues, similarly as in [50] (Section 3.4.3) from the local area. The Hessian matrix eigenvalue analysis is used in extracting the principal directions using which the local second order structure of the image can be decomposed[50]. Small manually annotated areas were then used as input for the classifier[48] (Section 3.4.4). The segmentation for the whole image was predicted from this data.

### 3.4.1 Gaussian blur

The Gaussian blur is performed by convolving the image with a Gaussian kernel that is defined here (Equation 3.2), where $x$ and $y$ are the location parameters. The source image is convolved with a Gaussian kernel multiple times by varying the standard deviation ($\sigma$) parameter.

$$\begin{aligned} g(x,y) = \frac{1}{2\pi\sigma^2} \times e^{-}\frac{x^2+y^2}{2\sigma^2} \\ \sigma = 2^a, \\ a \in \{-1,0,1,2,3,4\} \end{aligned} \tag{3.2}$$

### 3.4.2 Sobel edge detection

Sobel edge detection is performed by convolving the source image with kernels specified in Equation 3.3.

Let $A$ be the source image. We have

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{3.3}$$

where * describes the convolution operation used in signal processing.

### 3.4.3 Texture analysis with eigenvalues of a Hessian matrix

The Hessian matrix is obtained by using signal processing convolution with a second derivative of a Gaussian kernel in the $r$ and $c$-directions, respectively, as described in Equation 3.4.

$$H = \begin{bmatrix} H_{rr} & H_{rc} \\ H_{rc} & H_{cc} \end{bmatrix} \tag{3.4}$$

### 3.4.4 Random forest classifier

Random forest is an ensemble machine learning algorithm, that is constructed from a set of decision trees. Given a set of features $X$, we sample the feature space randomly with replacement such that we end up with multiple estimators (Equation 3.5)

$$\begin{gathered} h(X_1)_1, h(X_2)_2, \ldots h(X_i)_i, \\ X_1, X_2 \ldots X_i \subset X \end{gathered} \tag{3.5}$$

each of which "sees" a different subset of features from $X$. The number of estimators is noted with $i$. This random sampling is also performed in the depth direction: each node $n$ (or split) is performed with a different random subset of features (Equation 3.6):

$$n(X_1)_1 \rightarrow n(X_2)_2 \ldots n(X_j)_j \tag{3.6}$$

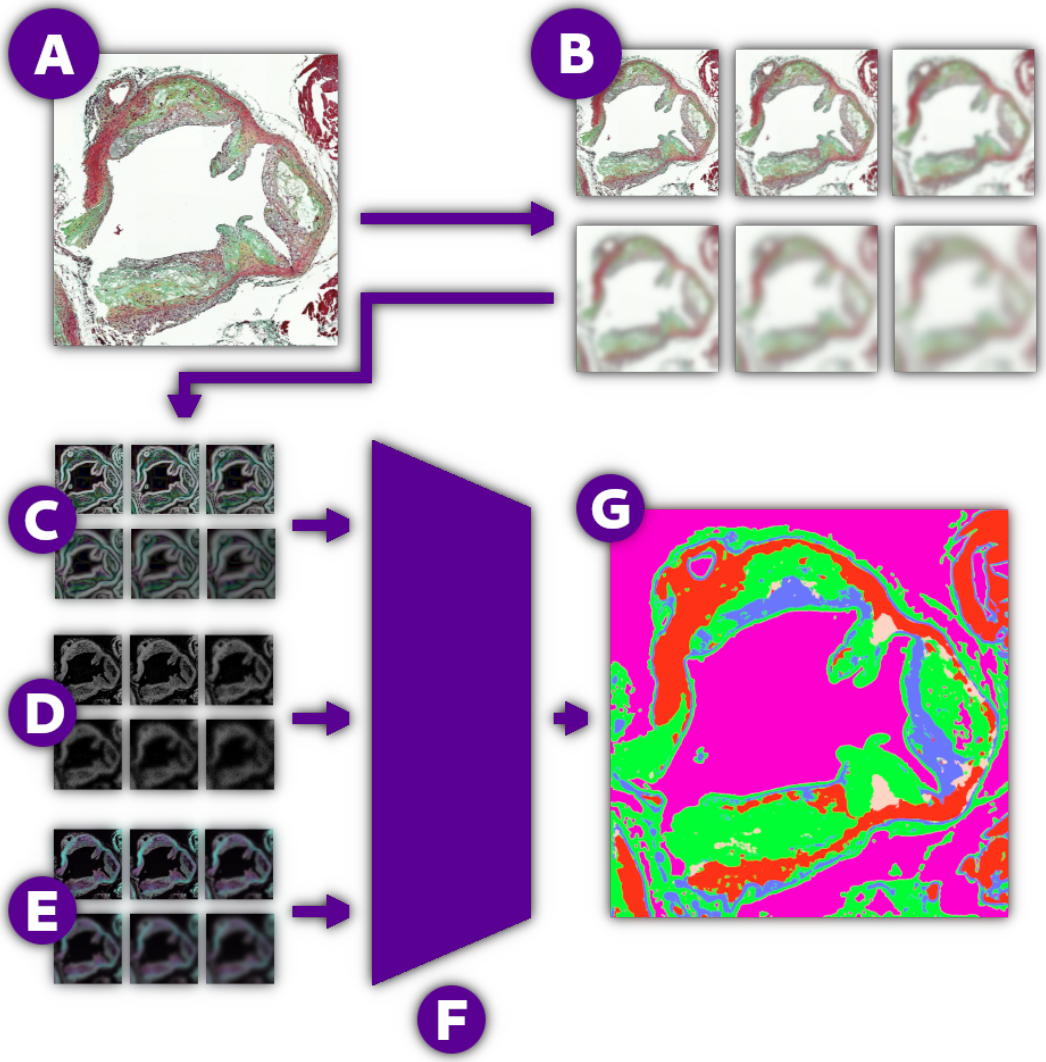


***Figure 3.5*** *Semi-automatic segmentation process. Small areas from the input images (A) are annotated corresponding to the resulting classes. Varying Gaussian filters are then applied to the local areas in the image (B). Three categories of features are extracted from these filtered areas: gradient intensity from the Sobel operator (C), local pixel color intensity (D) and texture analysis using eigenvalues from Hessian matrix applied to the area (E). These features are then used as a training set for a random forest classifier (F) that is then used to predict the annotated areas globally, resulting in a segmentation mask (G).*

where $j$ is the maximum depth of the decision trees.

## 3.5 Interactive network exploration tool

To visualize the cell network containing the assigned features interactively, a web-based data exploration tool was created. The purpose of the tool is enabling the observation and manual analysis of the detected cells and their properties such that

the cells are visualized in context of the original input data (i.e. the histological section images).

An interactive network data exploration tool was constructed utilizing the Plotly Python/Javascript framework[47]. The tool enables interactive observation of single cell network features plotted on top of the original images. On each cell the sum of staining specific features from all detected nuclei in the area was calculated an assigned to a specific color according to its value. A screenshot explaining the functionality of the tool is presented in Figure 3.6.

In detail, the computation and creation of the cell network is based on querying an computationally effective index to k-dimensional set of points, as provided by the Scipy Python package[51]. The original algorithm was described in [52]. This index is used in calculating which of the cells are to be considered neighbors based on a constant maximum distance. This constant value can be defined to either represent an approximate maximum distance with which cells can interact based on the studied biological phenomenon or simply some meaningful number that is able to separate the dense and sparse regions in a meaningful way. The computationally intensive calculations can be done beforehand and the interactive tool can be "compiled" into a static HTML file, resulting in client-side interactivity even though the tool handles large amount of data. These files can be shared and opened with any modern web browser, which makes co-operation easy between end users, even without complex initial setups.

## 3.6 Network feature quantification for single cells

To quantify network features of cells, the following descriptors were calculated:

1. Local feature statistics across individual cells
2. Major local feature combinations in network

Local feature statistics are defined as the distribution of detected staining-specific features per cell. In our test case, these features are very simple in nature and represent only the direct observations provided by the stainings (i.e. blue/purple colors in hematoxylin-based stainings are detected as nuclei). These features are obviously not limited to direct features inferred from staining colors. The features can be manually annotated or they can be a combination of multiple local characteristics in the images.

Local features are also visualized as a heatmap resembling their correlations with each other. This information is provided here as an example of a summary-level statistic that can be obtained from our pipeline, and a helpful validation for manual observations obtained by exploring the cell network interactively.

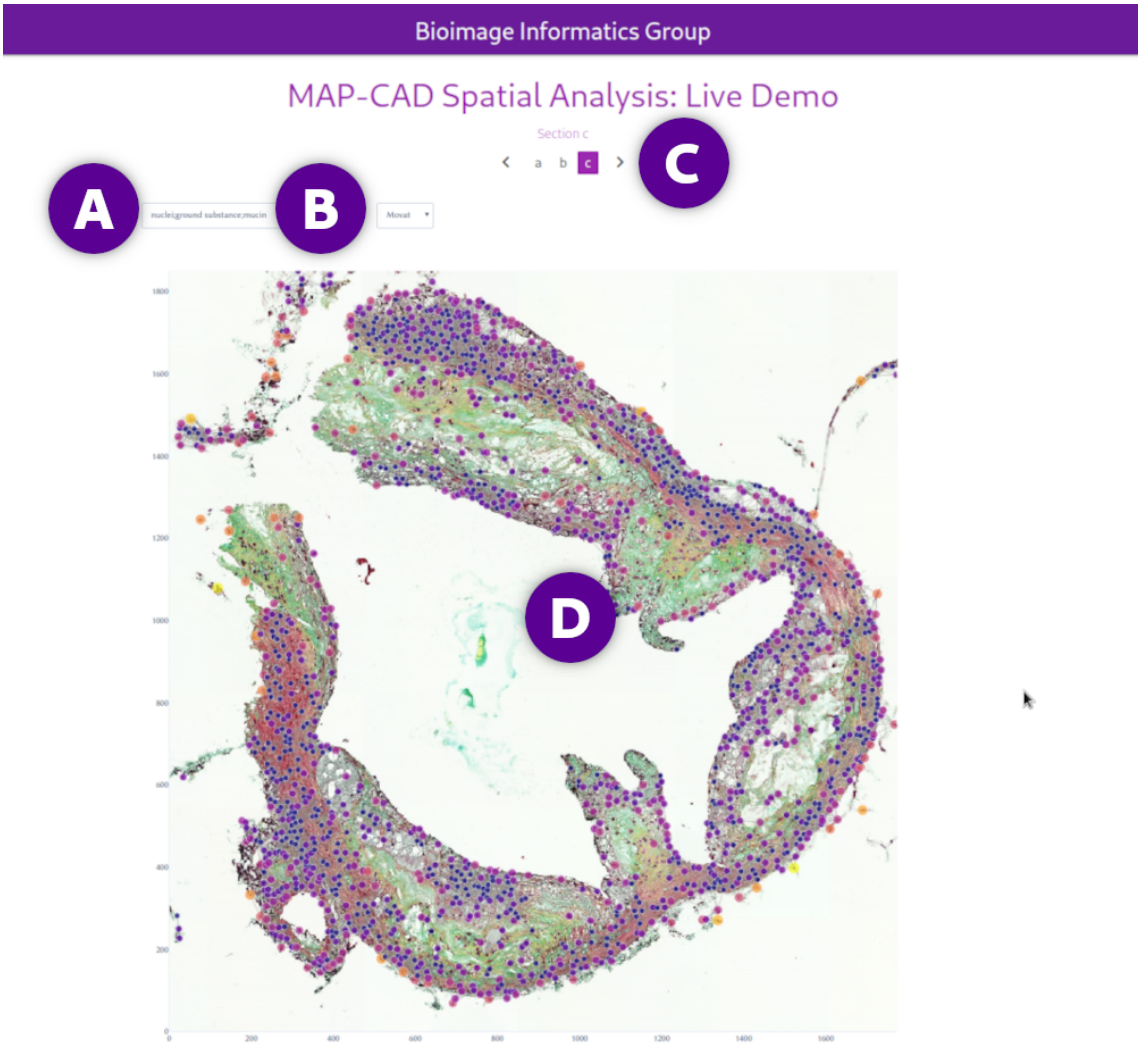


***Figure 3.6*** *Interactive network data exploration tool. The tool enables interactive data exploration by plotting different local features (A, dropdown menu from which to change the feature plotted) on top of different stainings (B, menu to select staining) across all of the sections (C, section changer). The network is plotted as a node/edge scatterplot with node colors determined by the feature value (D, view of the tissue with the network plotted)*

In addition and as a technical validation, error distribution statistics for Voronoi approximation were calculated. Error distribution is defined as the Euclidean distance between the nuclei in the approximated cell area and the nucleus used in the approximation, that is the nucleus that is nearest to all the points that define the approximated cell.

# 4 Results

The following results demonstrate the performance of our proposed pipeline in two ways. Firstly, technical performance is determined by evaluating the nuclei detection accuracy and the network approximation accuracy. Secondly, dataset-specific metrics related to the local features are calculated as an example of some the possible capabilities of the method in quantitative summarization of used features.

## 4.1 Nuclei detection accuracy

Nuclei detection accuracy was evaluated by predicting a confidence image from the trained, domain adapted network for 10% of the data totaling in 65 image tiles. The same tiles were manually annotated by approximately determining the area of the nuclei. This results in a binary image where the value 0 represents background and the value 1 represents nuclei. The confidence images, in turn, were thresholded with multiple threshold values ranging from 0.1 to 0.5 that were determined experimentally. Larger values resulted in a total loss of nuclei and smaller value were visually confirmed to not be suitable for the task. Visually the threshold value 0.2 was determined to give the most realistic nuclei segmentations.

The evaluation of the binarized confidence and ground truth images was performed by first calculating the centers of white "blobs" in the images and then calculating the distance between each of the blob between the confidence images and ground truth images. Because a pixel-level detection accuracy is almost impossible to attain, we allowed the centers of two blobs to be 10 pixels apart at maximum. As exactly one detected nuclei should be paired with one that is manually annotated, simply comparing nearest blobs would result in different scores depending on the processing order, as noted in the original article[36]. This happens because the nuclei are "consumed" when classified as either successful or non-successful detection. While the random error produced this way is likely to be small, it can be fixed by considering the pairing of detected and real nuclei as a form of an linear sum assignment problem, to which there exists various computational solutions that work in a deterministic way. The pairing in this case was performed with SciPy[51]. From a set of pairs, a F1 score was calculated for each of the thresholds used to binarize the confidence images. F1 score is defined to be the harmonic mean of precision and recall, and can be calculated with the formula in Equation 4.1, where $TP =$ number of true positives, $FP =$ number of false positives and $FN =$ number of false negatives.

**Nuclei detection accuracy**

| Threshold | F1 score |
|---|---|
| 0.1 | 0.57 |
| 0.2 | 0.57 |
| 0.3 | 0.56 |
| 0.4 | 0.52 |
| 0.5 | 0.48 |

***Table 4.1*** *The nuclei detection accuracy. Nuclei detection was evaluated by manually annotating nuclei in a subset of the used data. The confidence images outputted by the detection network were thresholded with different threshold ranging from 0.1 to 0.5 and F1 scores were calculated for each of the threshold.*

$$\begin{aligned} &\frac{TP}{TP + \frac{1}{2} \times (FP + FN)} \\ &\text{F1 score calculation formula.} \\ &TP = \text{number of true positives} \\ &FP = \text{number of false positives} \\ &FN = \text{number of false negatives} \end{aligned} \tag{4.1}$$

All of the scores are presented in Table 4.1.

## 4.2 Network properties

The following statistics and visualizations describe the network of cells obtained from the pipeline. As the sections A, B and C are not uniformly shaped and have slightly varying features with each other, the results are presented individually. Depending on the research setting, these features could be used in determining a difference between different histological samples, for example.

### 4.2.1 Voronoi approximation

The difference between the nuclei locations in the reference staining and nearest nuclei locations from other stainings is presented in Table 4.2 and in Figure 4.1. We can see that the distributions are long tailed, as there are many cells with relatively small approximation errors and then few cells with extremely large deviations in location from the nearest reference staining cell. The mass of the distribution will be at 0 as all of the nuclei are compared with one reference staining (i.e. the nuclei in the reference staining are also compared to itself). The distribution, of course,

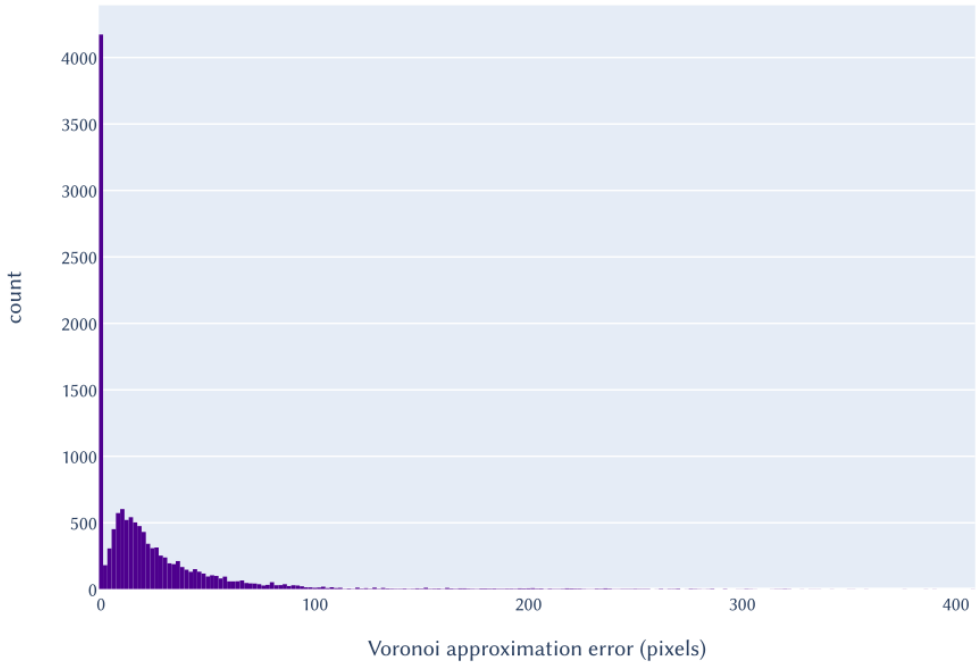


***Figure 4.1*** *Voronoi error in pixels distribution histogram. As all of the nuclei are compared to one reference staining, the greatest peak of the values is in 0.*

## Voronoi approximation error distribution

| Measure | Value (pixels) |
|---|---|
| mean | 24.1 |
| std | 39.6 |
| min | 0.0 |
| 25% | 0.0 |
| 50% | 12.5 |
| 75% | 29.6 |
| max | 407.2 |

***Table 4.2*** *The distribution statistics of the differences between detected nuclei from all of the stainings and one reference staining is presented here.*

will vary with the reference staining choice. In Figures 4.2, 4.3, 4.4, the log-transformed error value corresponds to the size and color of the scatter plot node such that the larger nodes represent larger approximation errors and smaller nodes smaller errors. The Figures represents sections A, B and C, respectively.

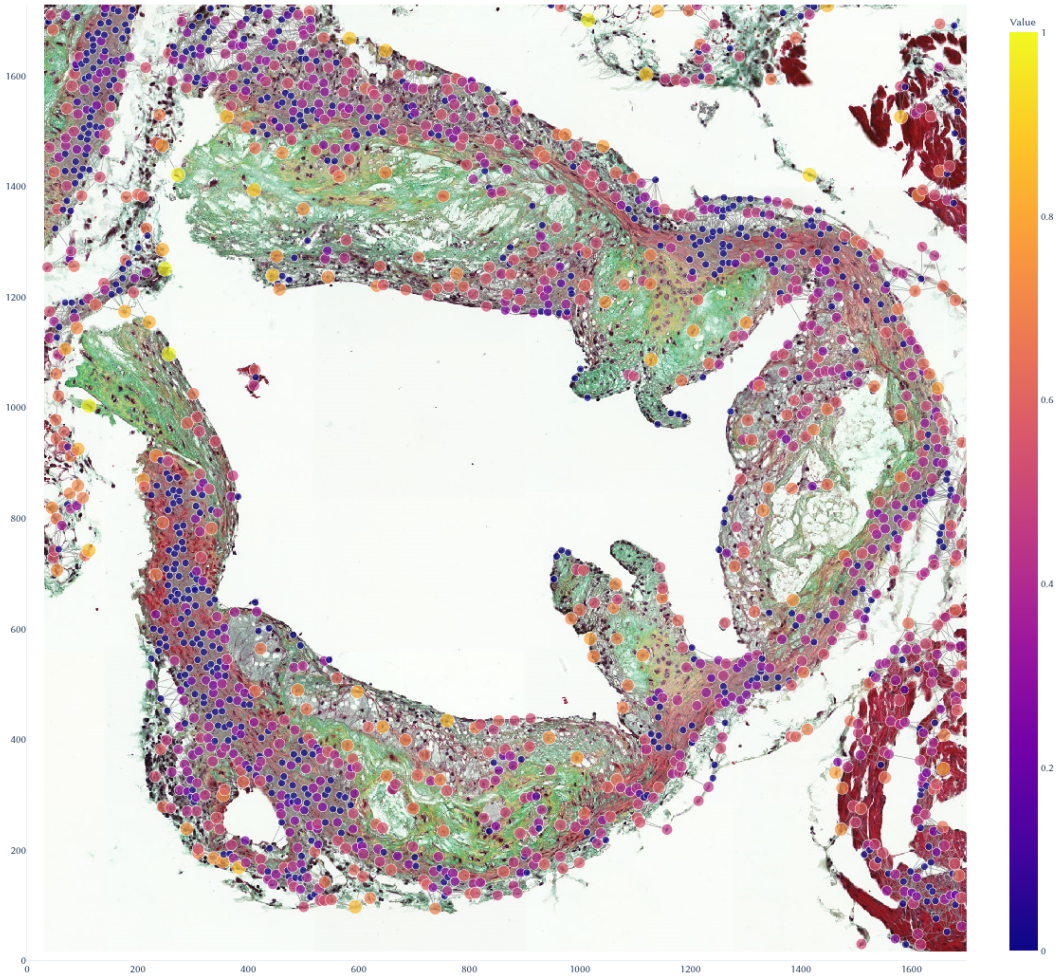


***Figure 4.2*** *Voronoi error distribution for Section A. The error in pixels is scaled between 0 and 1 and log-transformed for visualization purposes (0 = smallest point, purple; 1 = largest point, yellow). Both the color and size of the points are determined by the value.*

### 4.2.2 Local feature distribution

Local features extracted from the stainings were assigned to each cell according to its relative location. Figures 4.5, 4.6 and 4.7 represent the distributions for these features for sections A, B and C, respectively. Note that because these features are rather local and non-overlapping ones for other features than those derived from hematoxylin included in all IHC stainings, there will be a peak in the distributions at the value zero.

### 4.2.3 Local feature combinations

Local feature correlations for sections A, B and C are presented in Figures 4.8, 4.9 and 4.10, respectively. The correlations were obtained by comparing the different

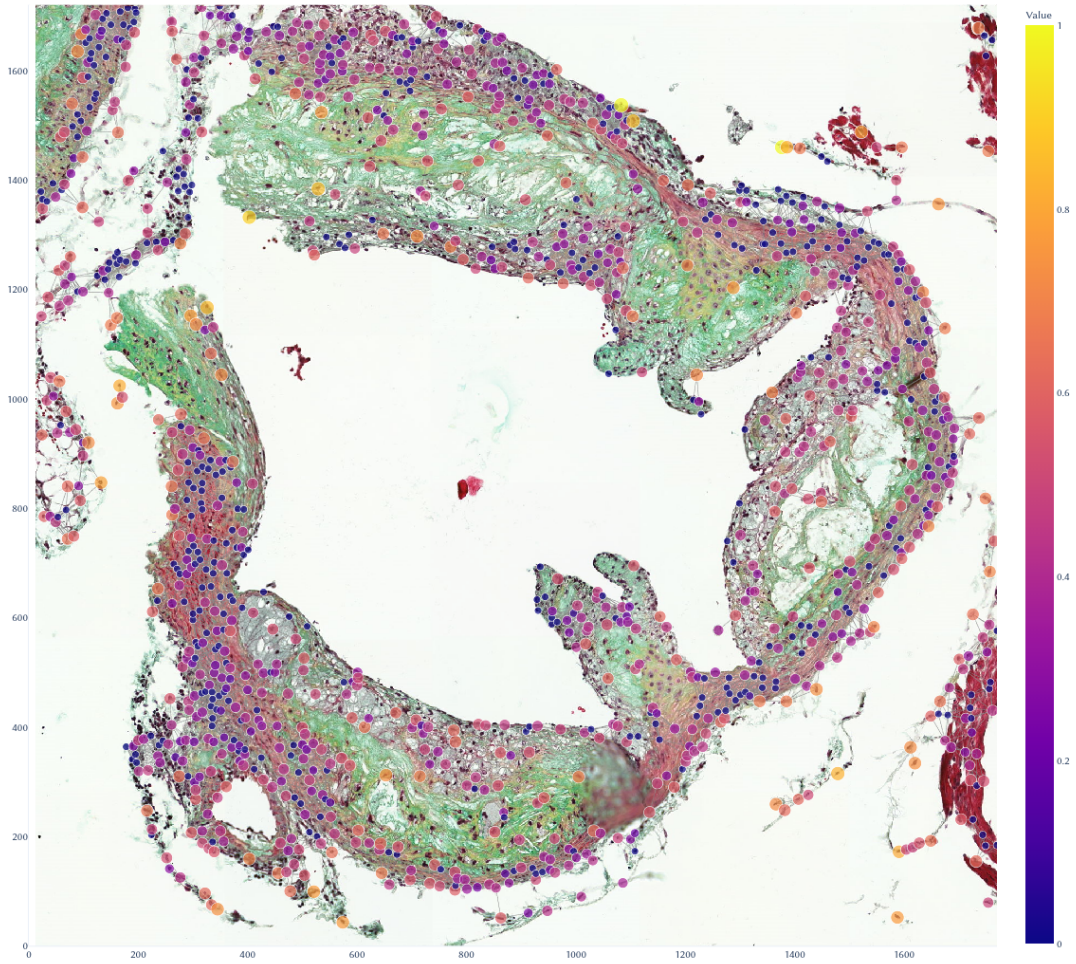


***Figure 4.3*** *Voronoi error distribution for Section B. The error in pixels is scaled between 0 and 1 and log-transformed for visualization purposes (0 = smallest point, purple; 1 = largest point, yellow). Both the color and size of the points are determined by the value.*

features obtained with each other in a pairwise manner.

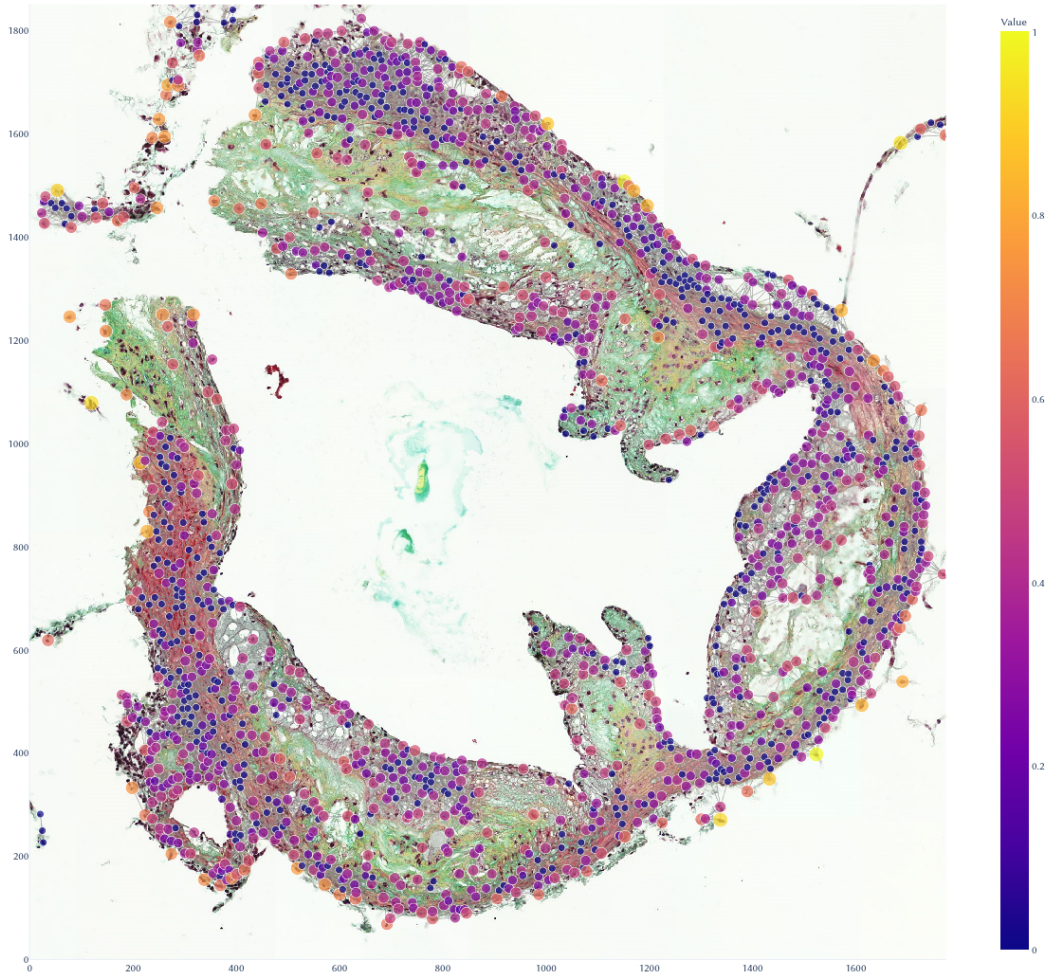


***Figure 4.4*** *Voronoi error distribution for Section C. The error in pixels is scaled between 0 and 1 and log-transformed for visualization purposes (0 = smallest point, purple; 1 = largest point, yellow). Both the color and size of the points are determined by the value.*

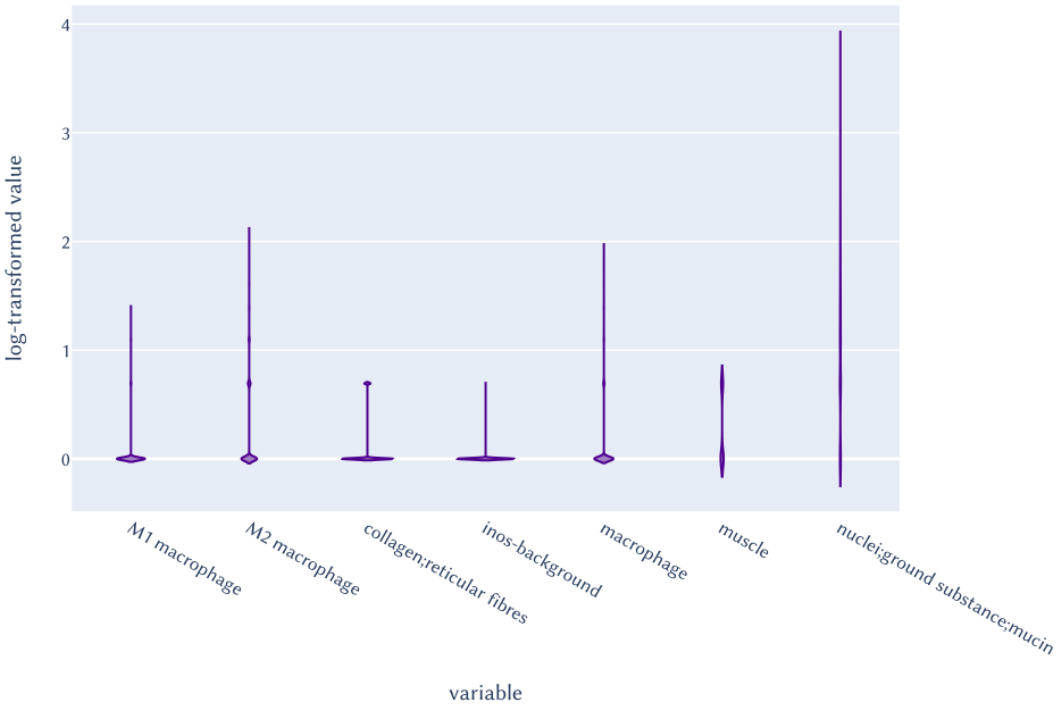


***Figure 4.5*** *Local features distribution for Section A*

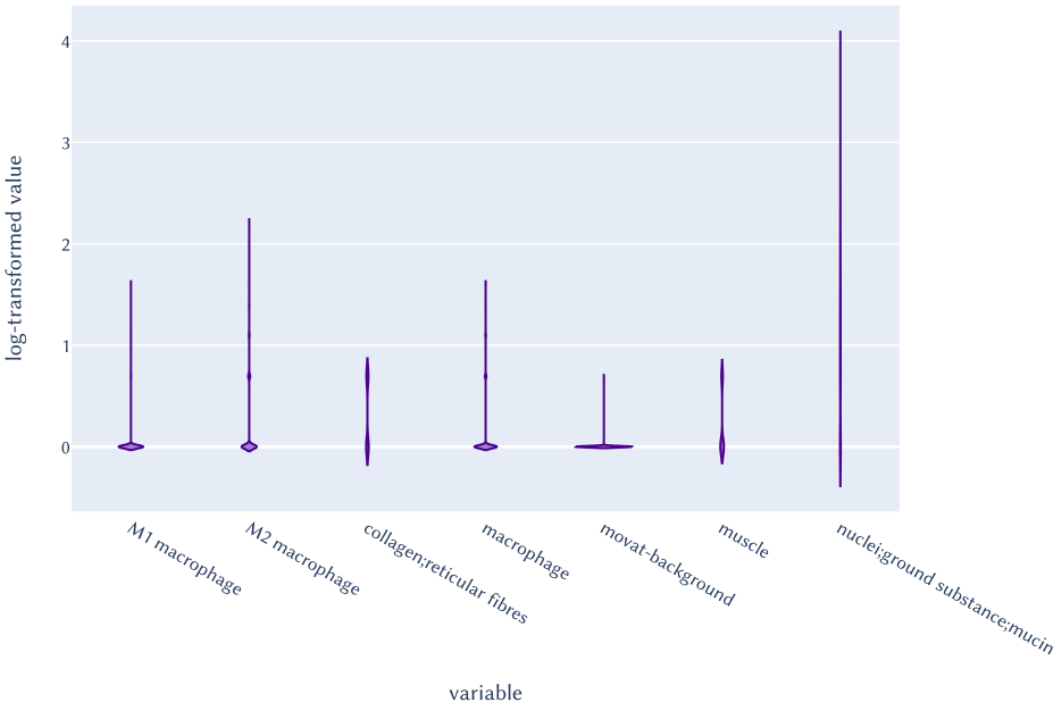


***Figure 4.6*** *Local features distribution for Section B*

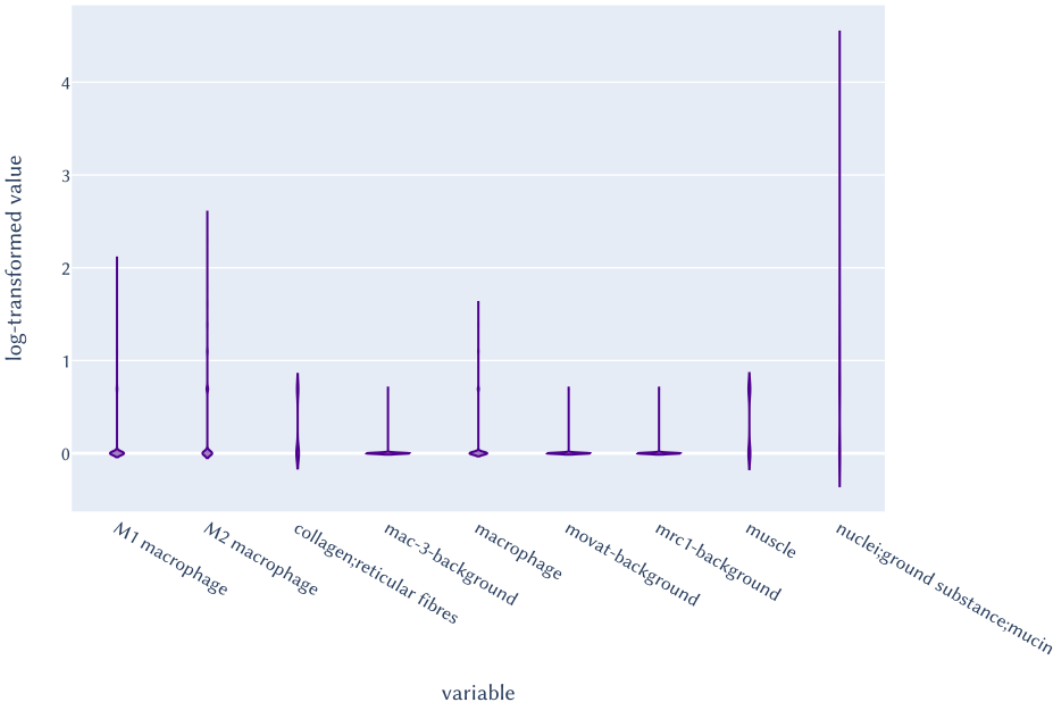


***Figure 4.7*** *Local features distribution for Section C*

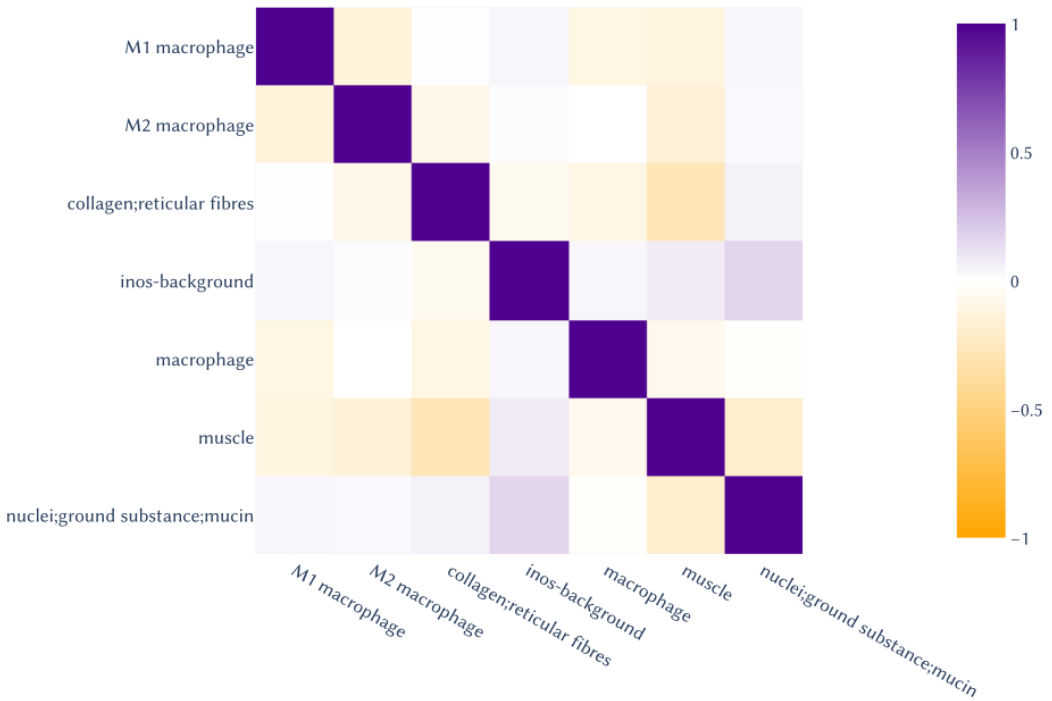


***Figure 4.8*** *Local features correlation for Section A*

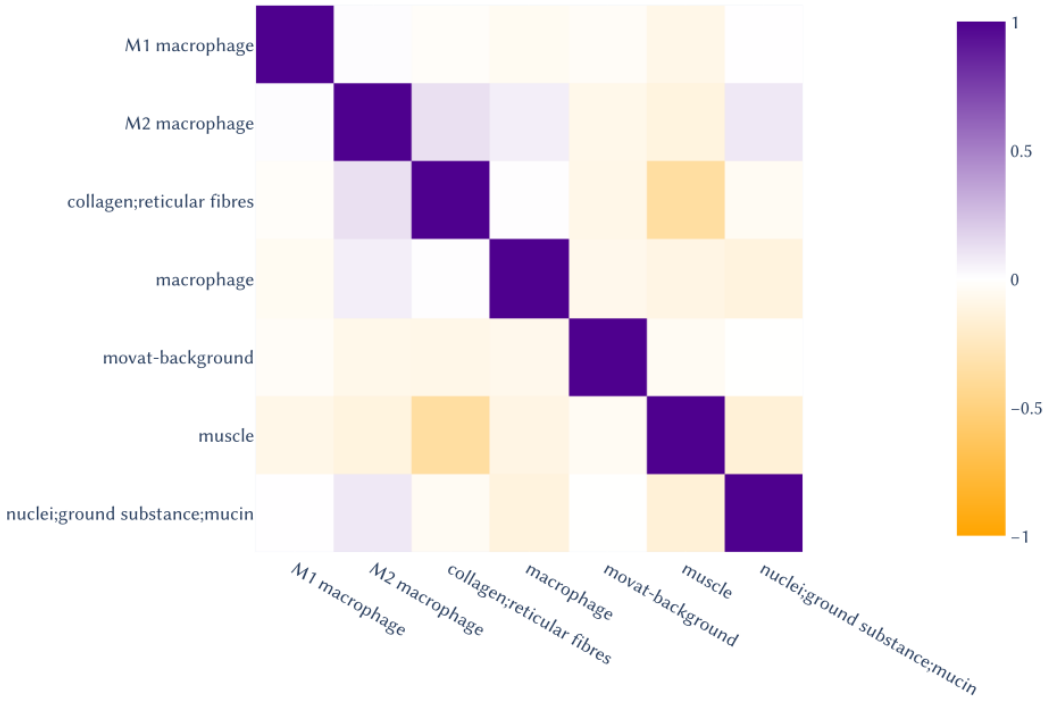


***Figure 4.9*** *Local features correlation for Section B*

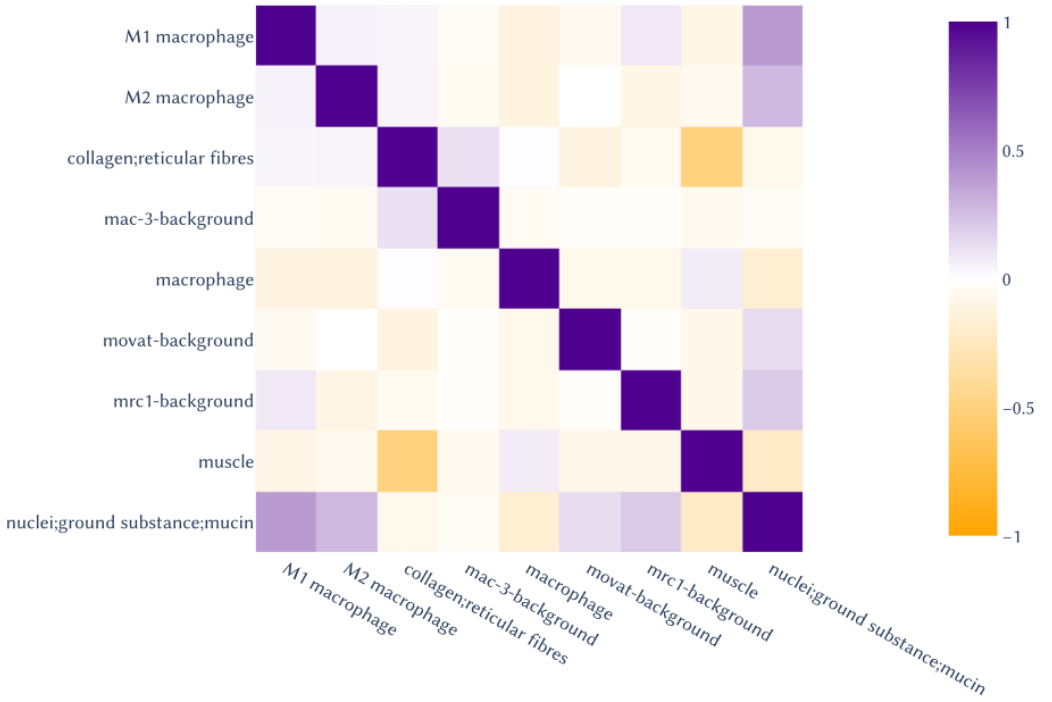


***Figure 4.10*** *Local features correlation for Section C*

# 5 Discussion

## 5.1 Evaluating the performance of the pipeline against its objective

The objective of this tool, the background of which is explained in Chapter 1, is to provide an exploration framework that meets the following requirements:

1. The tool must enable the analysis of tissue properties in an approximately single cell scale
2. The tool must enable combining features from multiple different histological stainings
3. The tool must model the network-like properties of multiple cells
4. The tool must be straightforward and easy to use

We present here the evaluation of the method in above context:

**1. The tool must enable the analysis of tissue properties in an approximately single cell scale** The tool meets the requirement in principle, even though true single cell segmentation is not obtained in three dimensions. Segmenting nuclei from consecutive sections results in a cylinder-shaped segmentations that do not accurately resemble cells shapes in three dimensions. Other factors that contribute to obtaining single cell analysis resolution are the base image resolution of the dataset, which was rather poor in our test case, and the nuclei detection accuracy. The nuclei detection step resulted in a rather poor evaluation F1 score, partly likely due to the image resolution and partly due to the complexity of the task. However, even considering these limitations, a rather good approximation of the varying cell densities (and thus areas that are likely functionally different) across the tissue is obtained. Therefore even in its current implementation, the tool enables local micro-scale analysis of tissue properties. See Sections 4.1 and 5.3.2.

**2. The tool must enable combining features from multiple different histological stainings** This requirement can be considered to be satisfied - however, in an ideal case, the registration would be done with an automatic algorithm as explained in Section 5.3.1. In Chapter 4, we demonstrate that the tool is able to aggregate relevant local information from the dataset by plotting local feature correlations and distributions in addition with the proposed tool itself. See Section 5.3.1.

**3. The tool must model the network-like properties of multiple cells** This requirement is met: the tool enables the analysis and observation of cells in relation to their neighbors. The tool is also easily extensible for performing more complex analyses to cell groups and clusters.

**4. The tool must be straightforward and easy to use** The end result can be run in the browser and hosted on a static site provider, meaning it that it does not require any complex server side setup. This makes using the tool easy. Further, most of the pipeline steps are either highly automated or require interaction on a level that does not require a lot of technical or biological knowledge. The most complex of the pipeline steps is currently the registration step, as it is currently performed manually.

## 5.2 Potential other applications and development

As demonstrated in earlier sections, our pipeline performs well with the dataset used in this study and provides a good framework for analyzing local properties from tissue histology across multiple stainings.

If complemented with additional data (for example signaling information that is known to be proportional to the relative counterpart locations), the tool could be used in modeling the signaling system and influence spatially across the tissue. This information can of course be analyzed together with other spatial information obtained from the tissue. Even if the nuclei detection step is only an approximate one, the relative nuclei (and cell) density is likely to be a factor in these spatial signaling systems.

## 5.3 Challenges and further improvements of the method

### 5.3.1 Evaluating registration performance

While there exists solid frameworks for analyzing the registration step performance[32], universal case- and tissue-independent baseline metrics for assessing image registration success are still to be developed. As noted in Section 3.2, the evaluation criterion are rather fuzzy, even though non-trivial. The registration step was first attempted with a state-of-the-art algorithm similar to [44], resulting in an unpredictable and too volatile results. The dimensionality of an already complex registration problem increases as the sections come from multiple stainings in contrast with having just one. In the case of uniformly stained images, the automated tools are to perform considerably better, even though even then the registration parameter space (depending on the algorithm) usually needs thorough exploration. While the tails

of the Voronoi approximation error distribution could lead us to some conclusions about registration performance, this type of indirect metric cannot be trusted as the Voronoi error is also influenced by nuclei detection accuracy and the relative nuclei locations across the sections.

It is however worth noting that using an automated algorithm for registration does not automatically result in better registration performance (even if only evaluated visually). If the best available metric is a qualitative one, manual registration can outperform other methods as it is tries to satisfy the qualitative criteria directly. The more crucial drawback of manual registration is that it requires more tedious manual work and the results do vary across the person conducting this step.[1]

### 5.3.2 Upstream processing influence on downstream analysis

The accuracy, both technical and biological, of our tool is heavily correlated with the success of the registration and nuclei detection analysis steps. If the images are not properly aligned with each other, correct spatial perspective is lost which negatively affects the downstream analysis. Likewise, if the nuclei are not correctly detected, accurate cell locations can not be defined and the spatial cell distribution does not accurately model the correct biological distribution. However, small local errors in detected nuclei locations will not, in the tissue level, result in large errors in the spatial cell distribution model.

### 5.3.3 Optimizing the interactive data exploration tool

The processing of large histological images and a large cell network information is a computationally heavy task, even if the computations for the analysis results can be done beforehand. The computational performance can be drastically improved by replacing browser-based technologies with those that are executed natively (without an extra layer of the browser), as the interactivity requires mainly client-side processing.

[1] An automated registration removes the work; as many of the algorithms resemble black boxes and produce largely different outcomes with tiny parameter changes, the variation of outcomes is not eliminated when using an algorithm.

# 6 Conclusions

A novel framework was proposed in this thesis to analyze multi-stained histological data in an approximate single cell level. The proposed method results in a network analysis tool that can be successfully used in combining multi-dimensional information in an intuitive way for tissue histology analysis. The method is a pipeline consisting of image registration, data post-processing, cell area approximation, staining-specific local feature extraction and assignment and network analysis, respectively, and utilizes state-of-the-art machine learning and signal processing algorithms. The modularity of the tool requires that the development process requires a rather broad understanding in the different technical and biological concepts in each of the steps. However, the modularity also makes it possible to develop the pipeline in parts together with a larger group of experts.

We demonstrate the performance of the tool with a dataset containing multiple different histological stainings and show that the framework enables user-friendly data exploration with local spatial precision and also the calculation of more general feature summaries easily. With additional data, the tool could be used together with spatial transcriptomics or spatial cell signaling models. The key challenges and future development targets of the tool include the computational optimization of the data exploration tool and the improvement in performing and evaluating the registration step.

The quantitative analysis of tissue histology is likely to develop towards a greater spatial precision and automation as well as a greater integration with other novel biological and technological innovations. In the future, it will thus be increasingly relevant to be able to combine not only various biological information sources, but also to recognize and apply new methodology from other rapidly developing fields of research. While being a single step in this process, our tools aims to be on the forefront in modern digital pathology analysis.

# References


[1] John KC Chan. "The wonderful colors of the hematoxylin–eosin stain in diagnostic surgical pathology". In: *International journal of surgical pathology* 22.1 (2014), pp. 12–32.

[2] Geert Litjens et al. "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis". In: *Scientific reports* 6 (2016), p. 26286.

[3] Kun-Hsing Yu et al. "Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features". In: *Nature communications* 7.1 (2016), pp. 1–10.

[4] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (2017), pp. 115–118.

[5] Andrew H Beck. "AI-powered pathology for precision medicine." In: *Cancer Immunology Research*. Vol. 8. 3. Amer Assoc Cancer Research. 2020, pp. 21–21.

[6] Bruce Alberts. *Molecular Biology of the Cell, Sixth Edition;* Garland Science, 2015, pp. 814–815.

[7] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[8] Shuiguang Deng et al. "On deep learning for trust-aware recommendations in social networks". In: *IEEE transactions on neural networks and learning systems* 28.5 (2016), pp. 1164–1177.

[9] Jiezhong Qiu et al. "Deepinf: Social influence prediction with deep learning". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2110–2119.

[10] Patrik L Ståhl et al. "Visualization and analysis of gene expression in tissue sections by spatial transcriptomics". In: *Science* 353.6294 (2016), pp. 78–82.

[11] Sanja Vickovic et al. "High-definition spatial transcriptomics for in situ tissue profiling". In: *Nature methods* 16.10 (2019), pp. 987–990.

[12] Anant Madabhushi and George Lee. "Image analysis and machine learning in digital pathology: Challenges and opportunities". In: *Medical image analysis* 33 (2016), pp. 170–175.

[13] Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[14] Tom M Mitchell et al. “Machine learning”. In: (1997).

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[16] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022.

[17] Dorian Pyle. *Data preparation for data mining*. morgan kaufmann, 1999.

[18] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[19] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260.

[20] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[21] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[22] Robert Hecht-Nielsen. “Theory of the backpropagation neural network”. In: *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[23] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[24] Balázs Acs, Mattias Rantalainen, and Johan Hartman. “Artificial intelligence as the next step towards precision pathology”. In: *Journal of internal medicine* 288.1 (2020), pp. 62–81.

[25] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[26] Martin Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467* (2016).

[27] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *arXiv preprint arXiv:1912.01703* (2019).

[28] Jonhan Ho et al. “Use of whole slide imaging in surgical pathology quality assurance: design and pilot validation studies”. In: *Human pathology* 37.3 (2006), pp. 322–331.

[29] Mark D Zarella et al. “A practical guide to whole slide imaging: a white paper from the digital pathology association”. In: *Archives of pathology & laboratory medicine* 143.2 (2019), pp. 222–234.

[30] Charilaos Christopoulos, Athanassios Skodras, and Touradj Ebrahimi. "The JPEG2000 still image coding system: an overview". In: *IEEE transactions on consumer electronics* 46.4 (2000), pp. 1103–1127.

[31] Po-Hsuan Cameron Chen et al. "An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis". In: *Nature medicine* 25.9 (2019), pp. 1453–1457.

[32] Kimmo Kartasalo et al. "Comparative analysis of tissue reconstruction algorithms for 3D histology". In: *Bioinformatics* 34.17 (2018), pp. 3013–3021.

[33] Kaisa Liimatainen et al. "Virtual reality for 3D histology: multi-scale visualization of organs with interactive feature exploration". In: *arXiv preprint arXiv:2003.11148* (2020).

[34] Kaisa Liimatainen et al. "3d-printed whole prostate models with tumor hotspots using dual-extruder printer". In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2019, pp. 2867–2871.

[35] Fuyong Xing and Lin Yang. "Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review". In: *IEEE reviews in biomedical engineering* 9 (2016), pp. 234–263.

[36] M. Valkonen et al. "Generalized fixation invariant nuclei detection through domain adaptation based deep learning". In: *IEEE Journal of Biomedical and Health Informatics* (2020), pp. 1–1. DOI: 10.1109/JBHI.2020.3039414.

[37] Kaustav Bera et al. "Artificial intelligence in digital pathology—new tools for diagnosis and precision oncology". In: *Nature reviews Clinical oncology* 16.11 (2019), pp. 703–715.

[38] Danton S Char, Nigam H Shah, and David Magnus. "Implementing machine learning in health care—addressing ethical challenges". In: *The New England journal of medicine* 378.11 (2018), p. 981.

[39] Björn Koos et al. "Next-generation pathology—surveillance of tumor microecology". In: *Journal of molecular biology* 427.11 (2015), pp. 2013–2022.

[40] Bryan He et al. "Integrating spatial gene expression and breast tumour morphology via deep learning". In: *Nature biomedical engineering* (2020), pp. 1–8.

[41] Johanna MU Silvola et al. "Aluminum fluoride-18 labeled folate enables in vivo detection of atherosclerotic plaque inflammation by positron emission tomography". In: *Scientific reports* 8.1 (2018), pp. 1–15.

[42] Paola Italiani and Diana Boraschi. "From monocytes to M1/M2 macrophages: phenotypical vs. functional differentiation". In: *Frontiers in immunology* 5 (2014), p. 514.

[43] Attila J Olah et al. "Differential staining of calcified tissues in plastic embedded microtome sections by a modification of Movat's pentachrome stain". In: *Stain technology* 52.6 (1977), pp. 331–337.

[44] Johannes Lotz, Nick Weiss, and Stefan Heldmann. *Robust, fast and accurate: a 3-step method for automatic histological image registration*. 2019. arXiv: 1903.12063 [cs.CV].

[45] The GIMP Development Team. *GIMP*. Version 2.10.12. June 12, 2019. URL: https://www.gimp.org.

[46] Vishal M Patel et al. "Visual domain adaptation: A survey of recent advances". In: *IEEE signal processing magazine* 32.3 (2015), pp. 53–69.

[47] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[48] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[49] Irwin Sobel and Gary Feldman. "A 3x3 isotropic gradient operator for image processing". In: *a talk at the Stanford Artificial Project in* (1968), pp. 271–272.

[50] Alejandro F Frangi et al. "Multiscale vessel enhancement filtering". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 1998, pp. 130–137.

[51] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature methods* 17.3 (2020), pp. 261–272.

[52] Songrit Maneewongvatana and David M Mount. "It's okay to be skinny, if your friends are fat". In: *Center for geometric computing 4th annual workshop on computational geometry*. Vol. 2. 1999, pp. 1–8.

# APPENDIX A. Code

The code for this thesis can be found in these repositories:

- `https://github.com/BioimageInformaticsTampere/mapcad-network` (Network analysis tool)
- `https://github.com/BioimageInformaticsTampere/mapcad-segment` (Interactive segmentation step)
- `https://github.com/BioimageInformaticsTampere/NucleiDetection/tree/refactor` (Nuclei detection)
- `https://github.com/BioimageInformaticsTampere/mapcad-nucdetect-utils` (Nuclei detection evaluation)