DISSERTATIONES
INFORMATICAE
UNIVERSITATIS
TARTUENSIS
76

DANIEL MAJORAL LOPEZ

Deep neural networks for microscopy images

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
76

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

76

DANIEL MAJORAL LOPEZ

Deep neural networks
for microscopy images

UNIVERSITY OF TARTU
Press
1632

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in Computer Science on December 9, 2025 by the Council of the Institute of Computer Science, University of Tartu.

Supervisors

Dr.	Leopold Parts Wellcome Sanger Institute, UK University of Tartu, Estonia
Prof.	Raul Vicente Zafra University of Tartu, Estonia

Opponents

Prof.	Nataša Sladoje Uppsala University, Sweden
Dr.	Craig Glastonbury Human Technopole, Italy

The public defense will take place on January 23, 2026 at 11:00 in Narva Rd. 18-1019.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906 (print) ISSN 2806-2345 (pdf)
ISBN 978-9908-57-101-0 (print) ISBN 978-9908-57-102-7 (pdf)

University of Tartu Press
`http://www.tyk.ee/`

To Asia

ABSTRACT

In recent years, deep learning has emerged as a powerful tool that leverages vast quantities of data to perform an expanding range of new tasks, traditionally beyond the capacity of computers. This thesis explores deep learning applied to microscopy image analysis, aiming to extract meaningful information from complex biological images.

In the first chapter we apply several networks (DeepCell, U-Net and Mask R-CNN) to brightfield microscopy images, and demonstrate that a U-Net can accurately segment nuclei directly from brightfield images, even when using as few as 16 training images. Moreover, input data from multiple focal planes enhances the ability to differentiate nuclei in the samples.

In the second chapter, we develop a new method with two neural networks: Stagetool, that analyses sperm development in microscope images of mouse testis. Stagetool simultaneously classifies accurately tubule cross-sections and sperm cells into different development stages.

Finally, in the last chapter we question the principles behind the preceding work. As a result, we develop a novel method: Kaizen. Kaizen aggregates object hypotheses to form an internal representation of an external image. During inference, Kaizen makes new guesses on image locations not yet predicted keeping only the ones that improve the overall similarity between internal representation and image prediction. This process is repeated until the internal representation matches the input image.

In conclusion, we present a diverse set of tasks and proposed solutions, that can give the reader a glimpse of the challenges present in analysing microscopy images. The tools and methods developed here pave the way for more accessible and scalable imaging analyses, potentially reducing reliance on expert manual annotation. Furthermore the proposed techniques, particularly the Kaizen model, might provide a foundation for more interpretable and adaptive neural networks.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ORIGINAL PUBLICATIONS

Publications included in the thesis

I Dmytro Fishman, Sten-Oliver Salumaa, Daniel Majoral, Tõnis Laasfeld, Samantha Peel, Jan Wildenhain, Alexander Schreiner, Kaupo Palo, Leopold Parts. “Practical segmentation of nuclei in brightfield cellimages with neural networks trained on fluorescently labelled samples”. In: Journal of Microscopy 284.1 (2021), pp. 12–24. https://doi.org/10.1111/jmi.13038

Author’s Contribution: The author main contribution was to evaluate how adding multiple focal planes affects segmentation.

II Oliver Meikar, Daniel Majoral, Olli Heikkinen, Eero Valkama, Sini Leskinen, Ana Rebane, Pekka Ruusuvuori, Jorma Toppari, Juho-Antti Mäkelä, Noora Kotaja. “STAGETOOL, a novel automated approach for mouse testis histological analysis”. In: Endocrinology 164.2 (2023). https://doi.org/10.1210/endocr/bqac202

Oliver Meikar, Daniel Majoral, Juho-Antti Mäkelä and Noora Kotaja contributed equally to this work.

Author’s Contribution: The author conceived the deep learning approach, developed and trained the cell classification network, performed the evaluation of individual models, and wrote the first version of the scientific manuscript.

III Daniel Majoral and Marharyta Domnich. “Kaizen: Decomposing cellular images with VQ-VAE”. In: PLoS One 20.5 (2025). https://doi.org/10.1371/journal.pone.0313549

Author’s Contribution: The author conceived and designed the method, coded the algorithm and trained the neural network, performed the model comparison with other methods, and wrote the scientific manuscript.

Publications not included in the thesis

- Daniel Majoral, Ajmal Zemmar, Raul Vicente. “A model for time interval learning in the Purkinje cell”. In: PLoS computational biology 16.2 (2020). https://doi.org/10.1371/journal.pcbi.1007601
- Tambet Matiisen, Aqeel Labash, Daniel Majoral, Jaan Aru, Raul Vicente. “Do Deep Reinforcement Learning Agents Model intentions?” In: Stats 6(1) (2022), pp. 50–66. https://doi.org/10.3390/stats6010004
- Aqeel Labash, Florian Stelzer, Daniel Majoral, Raul Vicente. “Emergence of adaptive circadian rhythms in deep reinforcement learning”. In: International Conference on Machine Learning. PMLR. 2023, pp. 18153–18170. https://proceedings.mlr.press/v202/labash23a/labash23a.pdf

PREFACE

"Think lightly of yourself and deeply of the world."
— Miyamoto Musashi

Every human being comes with a particular baggage of experiences, beliefs, and inclinations. The majority of them are unconscious and, thus do not trust too much anyone who talks about himself (and we do it frequently). However, when I started, a long time ago, the Ph.D. already had ideas that might condition how I approached the Ph.D.

First, honesty in the results, I made a self-commitment for integrity in the research results, independently of how insignificant the misrepresentation and how costly the alternatively. There are plenty of more rewarding endeavors that favor dishonesty, so if one wants to perform research, it is better to avoid falling on some slippery slope. Second, in my opinion, we are probably wrong about everything, and more specifically, I'm wrong about everything. Even the brightest of humans have been terribly wrong at some point. Every generation claims with confidence to have figured out how the world around us works and is later proven wrong. Most theories are proven false or at least nuanced in some sense. One must include mathematics, because information is conserved, the output of mathematical calculations only reflects the assumptions (explicit or implicit) that are input. Therefore no matter how precise and correct mathematics is, the conclusions are as good as the assumptions, which in most cases are wrong. Although I am probably wrong about all this.

Thus my naive thinking upon starting to do research was to understand what assumptions were made, presume them wrong, try to get something out of it, and report the results in an honest way.

INTRODUCTION

“The creation of a thousand forests is in one acorn ”
— Ralph Waldo Emerson

The universe is more than 13 trillion years old and composed of at least 2 trillion galaxies, each one with around 100 billion stars. Turning around one of these stars there is a small planet called Earth with around 8 billion human residents. But the universe’s overwhelming immensity is not perceived only in the far away galaxies where stars are born, space is bent by black holes, and old stars explode in supernovas. The immensity of the universe is found also in the small things. For example, any of the 8 billion humans is composed of 30 trillion cells and a similar number of bacteria [SFM16].

Consider a person among billions in which one of her 30 trillion cells is not working properly, or a cancer cell has emerged and it will wreak havoc. Will we one day be able to detect it?, since in comparison looking for a needle in a haystack seems a trivial task. Part of the solution is to have high-precision automatized methods able to process immense amounts of data in a very short time. The work presented here advances in this direction: exploring the use of deep learning methods to analyze microscopy images.

This thesis is structured as follows. First in the Background section we introduce some concepts from deep learning and microscopy and the terminology used. A particular emphasis is put on basic concepts from artificial neural networks. Then we present the articles that form this thesis in three different chapters:

- Chapter 1 summarizes the work done in the article [Fis+21]: "Practical segmentation of nuclei in brightfield cell images with neural networks trained on fluorescent labeled samples". In this work we investigate the application of deep learning methods to detect nuclei in brightfield images. In brightfield microscopy, the sample is placed on a transparent stage and illuminated from below with white light. Thus, the typical appearance of a bright-field microscopy image is a dark sample on a white or grey background. Brightfield images are easier to obtain than fluorescent images but they are very challenging to process by traditional methods and even for humans. The results showcase that a U-Net (See U-Net background section) segments brightfield images better than other networks, and that U-Net segments accurately with as few as 16 training images. Finally, input data from multiple focal planes improves results.

- Chapter 2 presents the article [Mei+23]: "STAGETOOL, a novel automated approach for mouse testis histological analysis". Although the causes are unknown, men’s sperm counts have declined by 50–60% between 1973 and 2018 [Lev+17; Lev+22]. Thus there is an urgent need for better methods to determine the causes of male infertility. Mouse models are essential for this research because their reproductive biology shares key similarities with humans, allowing controlled experimental studies that are not possible in people. To tackle this problem we developed Stagetool. Stagetool analyses mouse testis cross-sections in DAPI stained images. Stagetool consists of two neural networks that classify tubule cross-sections into five developmental classes and cells into nine categories with high accuracy. Moreover, Stagetool works properly in knockout mouse models (mouse in which a specific gene has been inactivated) with sperm production defects. Finally, we showcase how Stagetool can be applied for automated profiling of antigen expression.

- Chapter 3 introduces the article [MD24]: "Kaizen: Decomposing cellular images with VQ-VAE". The article proposes a new method for computer vision, Kaizen, developed thanks to the experience gained through the work from previous chapters. To segment objects Kaizen maintains an internal image composed of generated object hypotheses over the external microscopy image. The internal image is compared to the external one to detect prediction errors and non-predicted objects. We test Kaizen in two fluorescence microscopy datasets and discuss the advantages over other popular methods.

Overall the diverse tasks and solutions presented in the three chapters offer a window into the challenges of microscopy image analysis. We finalize this thesis by discussing the results and drawing general conclusions.

1. BACKGROUND

In this section, we give some background to the methods that were applied. We introduce and explain the core terminology to avoid confusion in later sections. Hopefully, this section helps the reader not familiar with some concepts either from deep learning or microscopy. First, we discuss deep learning most general aspects and dig deeper into applications for computer vision. Later, we introduce some basic concepts of microscopy and an overview of the field's current state.

1.1. Deep learning

Deep learning is a specific type of machine learning that utilizes artificial neural networks with multiple layers to analyze data. Deep learning is a vast and active area of research and falls outside this thesis's scope to cover it all. We refer the reader to the deep learning book [GBC16] for more thorough explanations of some basic concepts.

1.1.1. Artificial neural networks

> "Thou shalt not make a machine in the likeness of a human mind."
>
> — Frank Herbert, Dune

Artificial neural networks (ANNs) were developed, taking inspiration from biological neurons whose connections might strengthen or weaken in response to input information. Similarly, ANNs are formed by groups of simple artificial neurons where the connections depend on parameters that might change their value in response to input information. The most common artificial neural networks are feedforward neural networks. The following subsection describes feedforward neural networks and how they learn.

1.1.2. Feedforward neural networks

> "Deep and simple are far, far more important than shallow and complicated and fancy."
>
> — Fred Rogers

In a feedforward neural network, information travels only in one direction: forward, from the input to the output. There is no backward connections, thus feedforward neural networks form trees or acyclic graphs.

A neuron in a feedforward neural network is a uni-dimensional variable. Neurons in a feedforward neural network form groups of vectors of variables called layers. A layer in a neural network is a group of neurons that process information simultaneously. Typically the layer closest to the input is called the first layer; the second closest is the second layer, and so on. The last layer is called the output layer.

In a standard layer the value of a particular neuron is given by a linear transformation of the values of the previous layer, plus some non-linear transformation ϕ:

$$X_j = \phi(\sum W_{ji} X_i + a_j) \tag{1.1}$$

Where W_{ji} is known as the weight matrix and determines the connectivity of the layer. For example, when the layer weight matrix has no zeroes by design or structure (every neuron in one layer is connected to every neuron in the next layer), the layer is called a fully connected layer. A network formed entirely by fully connected layers is known as a fully connected neural network. In contrast, some layers have very sparse connectivity (most of the elements are zero in the matrix), for example, convolutional layers, which we will describe in detail in the computer vision subsection.

1.1.3. Learning in neural networks

> “Live as if you were to die tomorrow. Learn as if you were to live forever."
>
> — Mahatma Gandhi

To learn in a neural network is to approximate a function. For example consider the problem of learning the following function:

$$y = f(x) \tag{1.2}$$

where x and y are variables that can have any number of dimensions. To approximate $f(x)$ we might employ a neural network with parameters θ, that is able to express a family of functions g:

$$y^* = g(x, \theta) \tag{1.3}$$

To learn $f(x)$ is to find the closest function $g(x, \theta)$ to it, or in other words, to find the values k for θ that minimize some distance L between the two functions:

$$k \in \theta : \quad L(f(x), g(x,k)) \leq L(f(x), g(x, \theta)) \quad \forall \theta \tag{1.4}$$

or the equivalent:

$$arg\ min_{\theta}\ L(f(x), g(x, \theta)) \tag{1.5}$$

The distance L is known as the loss function and should be carefully selected since it will significantly impact the final results.

However in general $f(x)$ is an unknown function, although the desired result of the function y is known in some cases. For example, x might be all the possible microscopy images of cells, and $f(x)$ is a process that marks cell locations in the image. Thus we do not know $f(x)$, but we might have a limited set of images S, annotated by an expert biologist. Therefore, in many cases, we can only evaluate the loss L in the subset of data we have available:

$$arg\ min_{\theta}\ L(y_s, g(x_s, \theta)) : \quad \forall x_s, y_s \in S \tag{1.6}$$

where x_s are images and y_s ground truth annotations.In practice equivalent to the above we can sum the loss for each element of the dataset and minimize it:

$$L(S) = \frac{1}{n} \sum_{s=1}^{n} L(x_s, y_s) \tag{1.7}$$

$$arg\ min_{\theta}\ L(S) \tag{1.8}$$

Equation1.8 constitutes an optimization problem, and there are several methods to find its solutions, but in the deep learning field, the most common approach is stochastic gradient descent.

1.1.4. Stochastic gradient descent

> “Gradient descent can write code better than you. I’m sorry."
>
> — Andrej Karpathy

The gradient (∇) of a multivariate function L(x) is the vector of the partial derivatives of L(x):

$$\nabla L(p) = \begin{bmatrix} \frac{\partial L}{\partial x_1}(p) \\ \vdots \\ \frac{\partial L}{\partial x_n}(p) \end{bmatrix} \tag{1.9}$$

If the function L(x) is defined and differentiable around a point p then it decreases fastest in the opposite direction of the gradient of L:

$$-\nabla L(p) \tag{1.10}$$

Thus, If we want to move towards a local minimum of the function, we can take small steps in the opposite direction of the gradient:

$$p_{n+1} = p_n - \gamma \nabla L(p_n) \tag{1.11}$$

where γ is know as the learning rate. In general a large learning rate converges faster, but if it is too large generates numerical instability and does not converge.

Equation 1.11 constitutes the gradient descent algorithm, which upon certain conditions will converge to a local minimum. If we apply gradient descend to solve equation 1.8 for some point w in the space of parameters θ (w denotes the weights, the numerical values of the network’s parameters), we get:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \gamma \nabla L(\mathbf{w}_n) \tag{1.12}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\gamma}{N} \sum_{i=1}^{N} \nabla L_i(\mathbf{w}_n) \tag{1.13}$$

where i is a data point within a dataset. However, in this case, the gradient descent algorithm in requires immense computational resources since it calculates the gradient for all the data at each step. Thus instead, at each step the gradient is approximated by a random data point gradient, in what is known as stochastic gradient descent:

Algorithm 1 Stochastic gradient descent

```
Initialize:
    γ ← K
    W ← Random
    Randomize Dataset
for i = 1, 2, ..., n in Dataset do
    w_{n+1} = w_n − γ∇L_i(w_n)
end for
```

Training with one data point per step can be highly inefficient and, in many datasets, might not converge due to the excessive variance of the gradient estimates [QK20]. In practice, a middle ground between the whole dataset and the one data point approach might be better. Therefore typically, training is performed with a batch of data. The number of data employed is known as batch size (i.e. 8, 16, 32 images in computer vision). The approach is called mini-batch gradient descent (although deep learning practitioners refer loosely to it as gradient descent).

Over time several innovations have been added to stochastic gradient descent algorithm, for example:

- Training schedule: In which the learning rate decreases during training to improve convergence [LA20].
- Momentum: The next update is a linear combination of the current gradient and past ones, improving convergence [Sut+13].
- Adaptive learning rates: The learning rate differs for each parameter and changes depending on how often a parameter is visited [DHS11].

1.1.5. Representations

A representation is a structured encoding of data from the external world into a machine-understandable format that captures relevant features or patterns. Such representations simplify learning and enhance a system's ability to perform tasks like classification, detection, or reasoning.

A change of representation has similarities to a change of variables in mathematics. A change of variables is a technique used to simplify problems by substituting original variables with new ones, often to make equations more tractable. For example, in the case of the following function:

$$f(x) = (2x+7)^2 \tag{1.14}$$

We can create a new variable:

$$u = 2x+7 \tag{1.15}$$

Transforming the original function into:

$$f(u) = u^2 \tag{1.16}$$

In the example above we have replaced a complicated expression in terms of x with a simpler expression in terms of u. Although the function itself has not changed, its form has become easier to work with because of the new representation. In the same way, an appropriate choice of representation in machine learning can simplify the structure of the problem and make it easier for a neural network to learn the underlying function.

To grasp the importance of having appropriate representations the reader can consider that learning a task for a neural network is equivalent to approximating some function: y = f(x). As seen in section 1.1.3, the task is equivalent to minimizing a loss function L(S) (where S is some dataset $x_s, y_s \in S$).

For some datasets, the multidimensional loss function can have many local minima and saddle points, making hard to find the global minima. Consider a change of representation T:

$$u_s = T_x(x_s) \tag{1.17}$$

$$v_s = T_y(y_s) \tag{1.18}$$

Similarly to the change of variables, now we perform a change of representation in our dataset. Before changing the variable caused a transformation in the function, now changing the representation causes a transformation in the loss function:

$$L'(u_s, v_s) = L(x_s, y_s) \tag{1.19}$$

There are an infinite number of possible changes in representation and corresponding transformations of the loss function. Specifically as illustrated in Figure 1 there is a transformation of the loss function in which the loss landscape will present only one minimum, making it trivial to find by gradient descent the optimal solution.

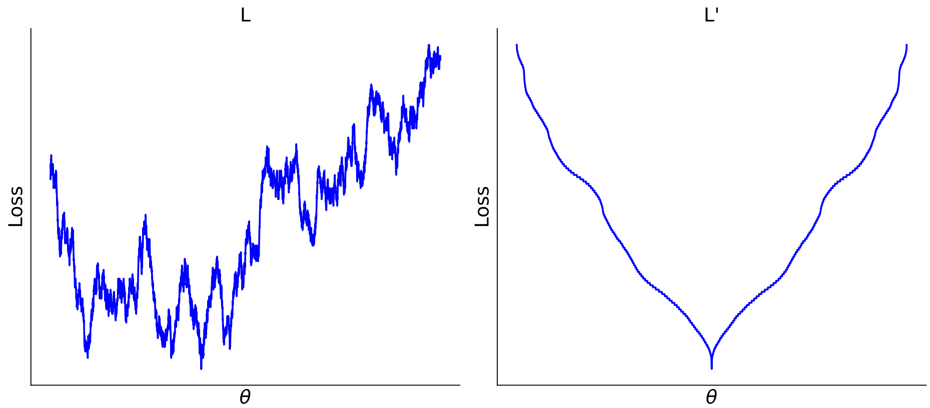

Figure 1: Loss transformation: The left panel shows a loss landscape generated with a random walk (at each step in the x axis goes randomly up or down). The right panel illustrates the same loss landscape after undergoing a transformation of representation, where the x axis is reordered in such a way that the loss has only one minimum

So we can conclude that for any task, a change in representation can make the task trivial. However, there are an infinite number of representations, so it is certainly not easy to find an appropriate representation. Nonetheless, when confronted with a hard problem, it is wise to try other data representations, because it might be easier to find a solution.

1.2. Computer vision

> “To see a World in a Grain of Sand
> And a Heaven in a Wild Flower
> Hold Infinity in the palm of your hand
> And Eternity in an hour "
> — William Blake, Auguries of Innocence

1.2.1. Convolutional layers

> “Among the infinite diversity of singular phenomena science can only look for invariants."
> — Jacques Monod

Kunihiko Fukushima [FM82] presented the first neural network (the neocognitron) whose output was unaffected by small changes in input position. This invariance to input translation is crucial in image recognition since a displaced object generally keeps all its attributes. However, in general, we want layers that preserve translational equivariance, meaning that when the layer input is shifted, the layer output experiences the same shift. The advantage of equivariance over invariance is that it preserves the location information, which is necessary for several tasks.

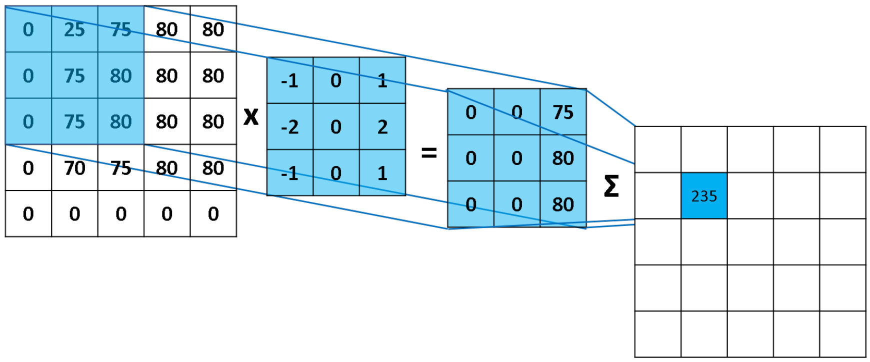

Figure 2: First step of a convolution with a 3x3 Kernel on a 5x5 input. Original Image by Rob Robinson: from https://mlnotebook.github.io/.

The layers designed to preserve translational equivariance are known as convolutional layers [LeC+98]. Convolutional layers are formed by several filters or kernels. Each filter is a set of parameters, tiny in width and height (typically 3x3 or 5x5) and with depth equal to the input. For example, for an RGB image with depth three, a convolution filter in the following input layer might consist of 5x5x3 (75 in total) learnable parameters. Filters are applied repeatedly across all the input (convolved) by computing the dot product between the filter and the local input values.

The convolution might be applied at each input position or skip some parts. The latter is specified by the stride with which we convolve the filter. For example, when the stride is 1x1, we apply the filter, and then the filter moves one position each time. Thus the filter is used at every possible location. When the stride is 2x2, we apply the filter and afterward move two positions. Therefore, the filter center skips the input's even rows and even columns.

Convolutional filters cannot be applied at the borders of the input, without altering the input, since the filter will partially fall outside. To address this problem, the input can be padded with zeros (or other options, such as mirroring the border values).

In summary, the hyperparameters that define a standard convolutional layer are: the number of filters N, the filter size K, the stride S, and the amount of padding P. The output size along one dimension of a convolutional layer for a given input dimension size W, can be calculated with the following formula:

$$W_{out} = \frac{W - K + 2P}{S} + 1 \tag{1.20}$$

1.2.2. Computer vision architectures

"Form ever follows function."

— Mies van der Rohe

The architecture of a neural network, refers to its structure and design, how its layers, neurons, and connections are organized and how data propagates through them. This section outlines several architectures that are necessary for understanding the later chapters of this thesis.

1.2.3. U-Net

U-Net [RFB15] is a convolutional neural network architecture designed for image segmentation tasks, such as biomedical imaging. Its architecture is characterized by a symmetric encoder-decoder structure with skip connections, enabling the network to capture both contextual and spatial information efficiently.

Figure 3 illustrates the U-net architecture. The network encoder first shrinks the input image information into a series of increasingly smaller spaces with a sequence of convolutions followed by max pooling operations. Afterward, the network decoder brings the image back to its original size with a succession of up-convolutions.

Following each max-pooling step in the contracting path, a skip connection integrates the resulting feature maps with those of matching spatial dimensions from the up-convolutional expansion path. This provides the decoder with high-resolution features, which are crucial for precise localization.

U-Net advantage is that it is a simple architecture that needs a small amount of labelled data for learning. The skip connections produce precise pixel-level segmentation. However, U-Net's fixed-size convolution kernels and pooling cause it to struggle with capturing long-distance dependencies [Wan+24].

1.2.4. Faster R-CNN

Faster R-CNN [Gir15] (Faster region-based convolutional neural network) is a two-stage object detection framework [Gir15]. It improves the original architecture R-CNN [DHS15] by integrating a Region Proposal Network (RPN) with the detection pipeline, enabling end-to-end training and faster inference.

Faster R-CNN proceeds in two stages. In the first stage a convolutional backbone (e.g. ResNet) converts the input image to feature maps. Then, the region proposal network (RPN) generates a set of candidate boxes of several sizes and positions in the input image.

In the second stage, features are extracted for every candidate box. The differently sized features are converted into a universal, fixed-size feature map using a technique called ROI align.

From the equally sized features, a small classifier predicts whether each box contains an object or is background. Another small network refines the anchor boxes to better align them with the actual objects by predicting offsets (adjustments).

Faster R-CNN is a flexible object detector that can detect objects at multiple scales and aspect ratios. Faster R-CNN supports various backbone networks and works well across different image domains. However, the two-stage design makes the architecture complex and harder to modify. One stage designs in general are faster but less precise.

1.2.5. Mask R-CNN

Mask R-CNN [He+17] extends Faster R-CNN by adding a third branch to predict object segmentations for all the detected objects.

To be able to segment objects precisely, Mask R-CNN introduces RoIAlign, which applying bilinear interpolation when differently sized features are converted into a universal, fixed-size feature map.

Mask R-CNN also incorporates feature pyramid networks (FPN), a top-down architecture using lateral skip connections to build a pyramid of features of different scales from a single-scale input. FPN endows Mask R-CNN with some scale invariance, thus detecting and classifying objects better at different scales. Figure 4 illustrates Mask R-CNN architecture.

Mask R-CNN main advantage is that provides object detection and segmentation in one network. Inherits the strong detection performance of Faster R-CNN, while providing pixel accurate segmentation. Some disadvantages are its complexity and slower inference time. Furthermore, Mask R-CNN shows systematic weaknesses when dealing with small objects or objects observed at oblique angles, since the candidate boxes are axis-aligned.

1.2.6. Autoencoder

An autoencoder[Kra91] is a type of artificial neural network used to learn an efficient compression of data. An autoencoder consists of two neural networks: an encoder that maps the input data to a lower-dimensional central layer, and a decoder that reconstructs the original input from the central layer. The central layer where the input data is compressed into a lower-dimensional representation is known as the autoencoder bottleneck.

The overall autoencoder architecture (encoder+decoder) is trained to reconstruct its input. The bottleneck in the autoencoder architecture ensures that the original set of data is transformed into a compressed representation. We have briefly discussed the importance of data representations in the section 1.1.5.

An important limitation of autoencoders is that they typically can not generate new data. In the following section, we discuss the Variational Autoencoder,

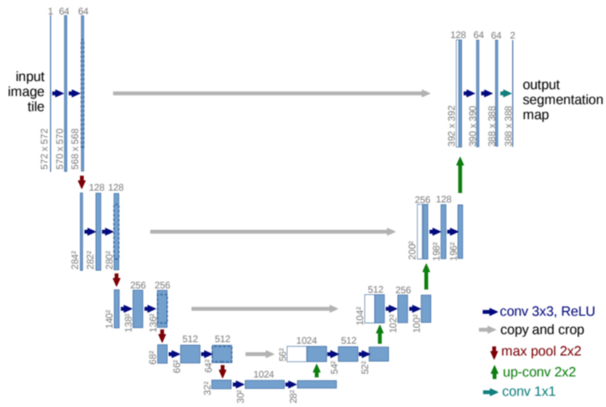

Figure 3: U-net architecture. Each blue box corresponds to a multi-channel feature map with the number on top. The x-y-size is provided at the lower left edge of the box. White boxes represent merged feature maps coming from skip connections (grey arrows).

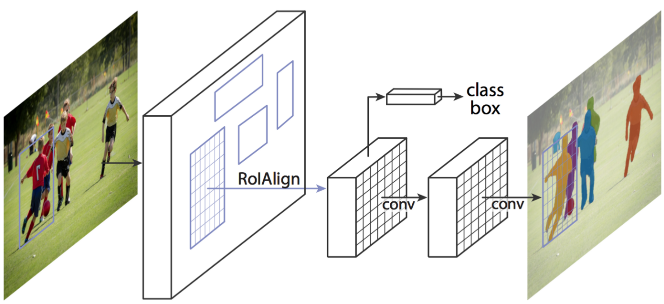

Figure 4: Mask R-CNN architecture. Mask R-CNN uses bilinear interpolation (RoiAlign) to extract a small feature map from a region of interest. With these features an object is classified and segmented

which is able to generate new data from the data distribution from which they were trained.

1.2.7. Variational Autoencoder

A variational autoencoder [KW13] is a probabilistic generative model with a similar architecture to an autoencoder. In that sense, the variational autoencoders possess an encoder, a decoder, and are trained to minimise the reconstruction error between the data after encoding and decoding and the initial data.

In contrast to autoencoders, in the variational autoencoder, a variational encoder maps the input variables to the parameters of a variational distribution. The decoder then takes any such sample and maps it back into the space of the input variables, effectively reconstructing the data.

Intuitively, by associating each input with a distribution (rather than a single point), the model spreads the representation out in the latent space. This makes the latent space smoother and "larger," allowing nearby points to generate similar outputs while still covering variability in the data.

In terms of Bayesian statistics, the decoder learns a probabilistic mapping between observed data x and a lower-dimensional latent representation z:

$$p_\theta(x \mid z) \tag{1.21}$$

In the variational autoencoder a prior distribution (standard normal distribution) is imposed on the latent variable:

$$p(z) = \mathcal{N}(z \mid 0, I) \tag{1.22}$$

This encourages the latent space to follow a known structure. The encoder has to learn the true posterior distribution:

$$p_\theta(z \mid x) \tag{1.23}$$

Applying the Bayesian rule, we get:

$$p_\theta(z \mid x) = \frac{p_\theta(x \mid z)\, p(z)}{p(x)} \tag{1.24}$$

Where $p(x)$ is the normalizing constant and is calculated as:

$$p(x) = \int p_\theta(x \mid z)\, p(z)\, dz \tag{1.25}$$

However, this integral is generally intractable, as the latent space is continuous and high-dimensional, and the Likelihood $p_\theta(x \mid z)$ is a complex and high-dimensional function. To make the calculation viable, variational autoencoders approximate it using a learned distribution q(z|x) called the variational posterior:

$$q(z \mid x) \approx p(z \mid x) \tag{1.26}$$

where $q(z \mid x)$ is a tractable distribution. Specifically, it is chosen to be Gaussian with mean and variance given by the encoder network:

$$q_\phi(z \mid x) = \mathcal{N}\big(z \mid \mu_\phi(x), \mathrm{diag}(\sigma_\phi^2(x))\big). \tag{1.27}$$

To find a training objective for our network we can incorporate the q(z|x) term into the log-likelihood as follows:

$$\log p_\theta(x) = \log \int q_\phi(z \mid x) \frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)} dz \tag{1.28}$$

Applying Jensen's inequality yields the *evidence lower bound* (ELBO):

$$\log p_\theta(x) \;\; \geq \;\; \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x \mid z)] - \mathscr{D}_{KL}[q_\phi(z|x)||p(z)] \tag{1.29}$$

The variational autoencoder is trained with the following loss:

$$\mathscr{L}_{vae}(x;\phi,\theta) = \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log p_\phi(x|z)\right] - \mathscr{D}_{KL}[q_\phi(z|x)||p(z)] \tag{1.30}$$

The first term, $\log p_\phi(x|z)$, can be seen as a reconstruction loss because it will push the model towards reconstructing the original x from its compressed representation, z.

The second term is a KL-divergence term. The KL-divergence measures how similar two distributions are. This KL-divergence term can be considered a regularization term on the reconstruction loss. It encourages the latent distribution to be similar to the specified prior distribution (typically a unit Gaussian distribution)

1.2.8. Vector Quantised-Variational AutoEncoder

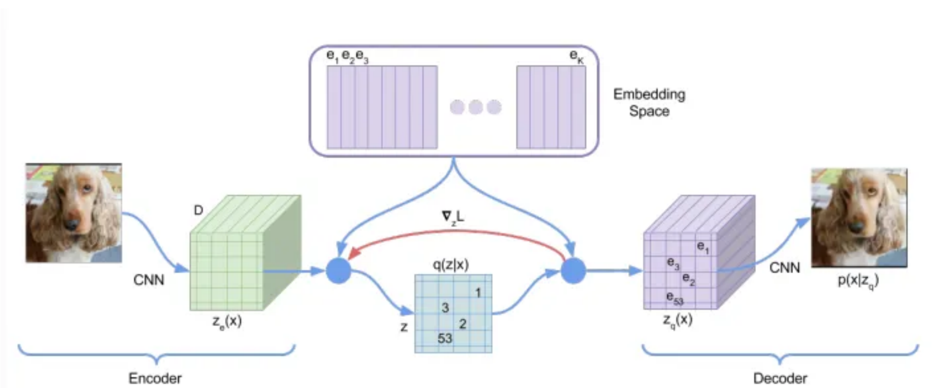

Figure 5: Scheme of VQ-VAE architecture. The VQ-VAE converts an image to discrete numbers, indexes to recover embeddings in a memory table. The embeddings are then used to reconstruct the original image. In red the gradient skipping the quantisation step. Image taken from [VV+17]

Many real-world data domains (language, phonemes, image descriptions, proteins, etc.) are inherently discrete. Variational Autoencoders struggle here because they operate on continuous latent variables. Van den Oord et al. proposed the Vector Quantised-Variational AutoEncoder (VQ-VAE) [VV+17] as an alternative to the Variational Autoencoder. The VQ-VAE replaces the continuous latent space with a discrete latent space, making the representation more interpretable and better suited for categorical data.

As illustrated in Figure 5, the VQ-VAE introduces a codebook . The codebook is a memory table with a set of learned embedding vectors e_i $(i \in 1, 2, ..., K)$ of dimension D.

Instead of encoding a set of inputs into a continuous latent space, the VQ-VAE encoder converts the input to a set of vectors $z_e(x)$ of dimension D. Then, the encoder maps each of these vectors to the nearest vector in the codebook. This process is called vector quantisation (VQ).

The VQ-VAE decoder employs the quantised embeddings $z_q(x)$ to reconstruct the input. Thus, the input to the decoder $z_q(x)$ as shown in equation 1.31.

Later, the nearest neighbour embeddings are recovered from the memory table. The set of nearest neighbour embeddings from all the latent variables will be the input to the decoder $z_q(x)$ as shown in equation 1.31.

$$z_q(x) = e_k \text{ where } k = argmin_j ||z_e(x) - e_j||^2 \tag{1.31}$$

Equation 1.31 is not differentiable. Thus when training, the gradient from the decoder $\nabla_z L$ is passed directly to the encoder, as seen in Figure 5. To train the VQ-VAE a loss with three terms is used as shown in equation 1.32

$$\mathscr{L}_{\text{vq-vae}} = \log p(x|z_q(x)) + ||sg[z_e(x)] - e||^2 + \beta ||z_e(x) - sg[e]||^2 \tag{1.32}$$

Where sg is the stop gradient operator, meaning no gradient propagates back from the operand. The first term from equation 1.32, is the reconstruction error and affects both the encoder and the decoder:

$$\mathscr{L}_{\text{reconstruction}} = \log p(x|z_q(x)) \tag{1.33}$$

The second term is the codebook error and optimises the embeddings towards the encoder output.

$$\mathscr{L}_{\text{codebook}} = ||sg[z_e(x)] - e||^2 \tag{1.34}$$

The third term, equation 1.35, is called commitment loss. This term is necessary because the gradients from the reconstruction loss of the encoder are a straight-through copy coming from the decoder, and they can grow arbitrarily far away from the embeddings. Specifically, this will happen if the embeddings do not train as fast as the encoder parameters. In this case the codebook loss can be too big and present large variance making the training fail.

$$\mathscr{L}_{\text{commitment}} = \beta ||z_e(x) - sg[e]||^2 \tag{1.35}$$

Where β is a hyperparameter that works well in a wide range (from 0.1 to 2.0).

1.2.9. Evaluation metrics

To quantitatively assess the performance of object detection and segmentation models, in this thesis four metrics were used: pixel accuracy, mean accuracy, mean Intersection over Union (mean IoU), and frequency-weighted Intersection over Union.

Pixel accuracy measures the proportion of correctly classified pixels across all classes. If n_{ii} denotes the number of correctly classified pixels for class i, and t_i represents the total number of ground truth pixels belonging to class i. Pixel accuracy follows the following formula:

$$\text{Pixel accuracy} = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{1.36}$$

Mean accuracy measures the average accuracy per class. For each class i, accuracy is defined as the ratio between correctly predicted pixels (true positives) and the total number of ground truth pixels of that class. The mean accuracy is computed as:

$$\text{Mean Accuracy} = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{n_{ii}}{t_i}, \tag{1.37}$$

where N_c is the total number of classes, n_{ii} denotes the number of correctly classified pixels for class i, and t_i represents the total number of ground truth pixels belonging to class i.

The Intersection over Union (IoU) for each class i quantifies the overlap between the predicted and ground truth regions, defined as:

$$\text{IoU}_i = \frac{n_{ii}}{t_i + p_i - n_{ii}}, \tag{1.38}$$

where p_i is the total number of pixels predicted as class i. The mean Intersection over Union is then computed by averaging across all classes:

$$\text{mIoU} = \frac{1}{N_c} \sum_{i=1}^{N_c} \text{IoU}_i. \tag{1.39}$$

The frequency-weighted IoU accounts for the relative frequency of each class in the dataset, weighting each class IoU by its proportion of pixels:

$$\text{Frequency-weighted IoU} = \sum_{i=1}^{N_c} \frac{t_i}{\sum_{j=1}^{N_c} t_j} \cdot \frac{n_{ii}}{t_i + p_i - n_{ii}}. \tag{1.40}$$

This formulation ensures that classes occupying larger areas in the image have a greater influence on the final score, to compensate class imbalances.

1.3. Microscopy

"Little drops of water, little grains of sand,
Make the mighty ocean, and the pleasant land.
So the little minutes, humble though they be,
Make the mighty ages of eternity.

— Julia Abigail Fletcher Carney, Little Things

Microscopy is a technique that involves using a microscope to observe and study objects or phenomena that are too small to be seen with the naked eye. Microscopy is an important tool in many fields, including biology, materials science, and engineering. It allows us to study the structure and behavior of small objects and phenomena, which can help to advance our understanding of the world around us.

1.3.1. Microscope Types

There are several different types of microscopes that are used for different purposes, including light microscopes, fluorescent microscopes, electron microscopes, and scanning probe microscopes.

Light microscopes are the most common type of microscope and use visible light and lenses to magnify the image of an object. They can be further divided into different types based on the way they produce the image, such as compound microscopes, which use multiple lenses to magnify the image, and stereo microscopes, which use two separate eyepieces to produce a three-dimensional image.

Electron microscopes use a beam of electrons rather than light to produce an image. They are much more powerful than light microscopes and can magnify objects up to a million times their actual size. However, they can only be used to observe non-living samples, as the electron beam used to produce the image can damage living tissue.

Scanning probe microscopes use a very fine probe to scan the surface of an object and create an image based on the interactions between the probe and the surface. They can produce images of objects at a very high resolution and are often used to study the properties of materials at the atomic level.

1.3.2. Illumination techniques

There are several types of illumination techniques used in microscopy, including:

- Bright field illumination: This is the most basic and commonly used illumination technique in microscopy. It involves shining a light source from below the sample and observing the sample through the eyepieces or a camera. The sample appears bright against a dark background.
- Dark field illumination: In this technique, the light source is positioned at an angle to the microscope objective lens, so that light is scattered by the sample and the background remains dark. This technique is useful for

observing structures that are transparent or translucent, as it enhances the contrast between the sample and the background.

- Fluorescent illumination: This technique uses a light source that emits light of a specific wavelength, which is absorbed by a fluorophore (a chemical that absorbs and re-emits light) in the sample. The absorbed light is then emitted at a longer wavelength, which can be observed using a filter or a specialized microscope. Fluorescent illumination is used to visualize specific structures or molecules in a sample.
- Phase contrast illumination: This technique is used to observe transparent or slightly opaque samples and is especially useful for observing living cells. It uses a special phase plate that is placed in the microscope's condenser to create a phase shift in the light passing through the sample. This causes the sample to appear bright against a dark background, similar to bright field illumination, but with greater contrast.
- Differential interference contrast (DIC) illumination: This technique is similar to phase contrast illumination, but it uses a polarized light source and specialized prisms in the microscope to create a more detailed image of the sample. It is often used to observe fine structures and details in transparent or translucent samples.

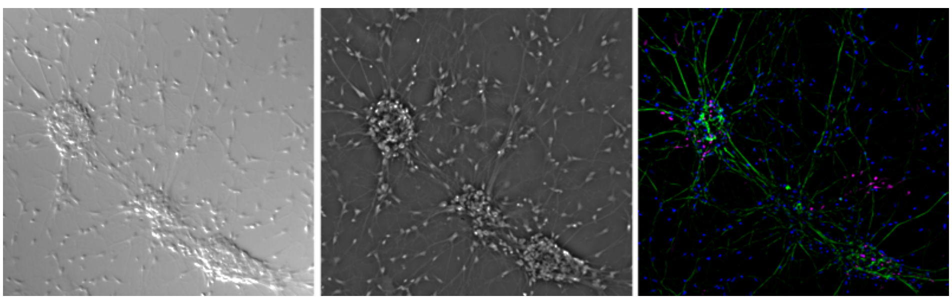

Figure 6: Different illumination techniques for the same neuron sample. Left) With differential interference contrast (DIC), cells can be identified but axons cannot, contrast is poor. Center) Phase contrast, greater contrast than DIC with cells and axons easily identified. Right) Fluorescence, cells and axons are easily identified, with selected proteins further highlighted in color by fluorescent markers (green for β-tubulin, a neuronal marker, blue for DAPI, a nucleus marker). Image from Leica Microsystems: Introduction to Widefield Microscopy.

The work presented in this thesis involves mainly two of these techniques, brightfield microscopy and fluorescent microscopy.

Brightfield microscopy is a type of microscopy that uses transmitted light to produce an image of a sample. In brightfield microscopy, the sample is placed on a transparent stage and illuminated from below. The light passes through the sample and is focused by the microscope's objective lens onto an eyepiece or a camera, producing an image of the sample.

Brightfield microscopy is the most basic and commonly used type of microscopy. It is often used to observe and study the morphology, or physical structure, of cells, tissues, and other biological material. Brightfield microscopy is particularly useful for the visualization of structures that are transparent or have a high refractive index, as they will absorb or scatter light and appear as dark structures against a bright background.

Brightfield microscopy has several limitations, including the lack of contrast in samples that are homogenous in color or have a low refractive index, and the inability to distinguish between different structures within the sample. To overcome these limitations, other techniques, such as fluorescent microscopy, can be used in combination with brightfield microscopy.

Fluorescent microscopy [Dru12; LC05] is a type of microscopy that uses fluorescent dyes or proteins to label and visualize specific structures or molecules within a sample. In fluorescent microscopy, the sample is first treated with a fluorescent dye or protein that attaches to the specific structure or molecule of interest. When the sample is excited by light of a specific wavelength, the fluorescent dye or protein will emit light at a longer wavelength, which can be detected by the microscope.

Fluorescent microscopy allows for the visualization of specific structures or molecules within a sample, as well as the localization of these structures or molecules within the sample. It can be used to study a wide range of biological samples, including cells, tissues, and organisms.

1.3.3. Cell Segmentation

Traditional cell segmentation methods rely on classical image processing techniques that use pixel intensity, edges, and geometric features to separate cells from the background.

The simplest method of image segmentation is the thresholding method. The thresholding method replaces each pixel in an image with a black pixel if the pixel intensity is less than a fixed value, called the threshold, or a white pixel if the pixel intensity is greater than or equal to that threshold value. The threshold can be set automatically by an algorithm. The most widely used method to set a threshold is Otsu's method [Ots+75], which searches for the threshold that minimizes intra-class variance.

However, thresholding alone is often insufficient for complex biological images, particularly when cell boundaries are faint, illumination is uneven, or cells overlap. To address these challenges, more advanced traditional methods have been developed.

Edge detection techniques, such as the Sobel or Canny operators [SF+68;

Can09], attempt to identify sharp intensity transitions that correspond to cell boundaries. These methods can delineate well-defined contours but are highly sensitive to noise and may fail when edges are weak or discontinuous.

Another widely applied approach is the watershed transform, which treats the image as a topographic surface, with pixel intensities representing elevations. By simulating a flooding process, the watershed algorithm partitions the image into catchment basins, effectively separating touching or clustered cells. Although powerful, it is prone to over-segmentation if noise and spurious gradients are not suppressed through preprocessing.

Active contour models represent another class of segmentation techniques. These methods evolve a deformable curve or surface under the influence of internal smoothness constraints and external image forces until it converges on cell boundaries. Active contours can capture irregular and dynamic shapes but often require careful initialization and significant computational resources.

Finally, morphological operations (e.g., dilation, erosion, opening, and closing) are frequently used to refine segmentation masks, eliminate small artifacts, and split joined regions.

Overall, traditional approaches form the foundation of cell segmentation, offering interpretable and computationally efficient solutions. Yet, their performance is limited in real-world datasets characterized by low contrast, high variability, and dense cellular arrangements—conditions where modern machine learning and deep learning techniques now dominate.

In contrast to traditional methods, deep learning approaches have transformed cell segmentation by automatically learning features directly from raw image data, eliminating the need for manual feature engineering.

One of the most influential architectures is the U-Net, an encoder–decoder network with skip connections that allow the model to capture both contextual information and fine-grained spatial details. U-Net and its numerous variants have become the standard for biomedical image segmentation due to their high accuracy and ability to generalize across diverse datasets.

The problem of distinguishing neighboring cells in dense images is tackled by instance segmentation models. Notable examples include Mask R-CNN [He+17], which extends object detection frameworks to generate precise segmentation masks, and StarDist [Sch+18], which represents cells as star-convex polygons to improve boundary delineation.

Recent advances also focus on weakly supervised and self-supervised approaches, which reduce the need for labor intensive annotations. These methods leverage sparse labels or exploit large amounts of unlabeled data for pretraining.

Despite their success, deep learning methods require substantial datasets, high computational resources, and careful tuning to achieve robust performance.Nevertheless, deep learning represents the current state of the art in cell segmentation, consistently outperforming traditional methods and improving biomedical imaging.

1.4. Spermatogenesis

> "The reproduction of mankind is a great marvel and mystery. Had God consulted me in the matter, I should have advised him to continue the generation of the species by fashioning them out of clay."
>
> — Martin Luther

Spermatogenesis is the process by which sperm cells are produced. This process takes place in seminiferous tubules, which are located in the testes. Spermatogenesis involves the transformation of primordial germ cells into mature sperm cells, a process that takes approximately 64-75 days in humans.

The process of spermatogenesis begins when primordial germ cells migrate from the yolk sac to the developing testes. Once in the testes, these cells undergo a series of changes that result in the production of sperm cells.

Immature germ cells (spermatogonia) form concentric layers next to the wall of a seminiferous tubule cross-section. The germ cells then undergo a process to produce immature sperm in the innermost part of the seminiferous tubule. Three significant steps characterize this process.

The first step in spermatogenesis is the formation of more spermatogonia. Germ cells undergo a series of divisions, known as mitosis, to produce more spermatogonia.

In the second step spermatogonia undergo a process known as meiosis, which divides the chromosome number by two. It starts with DNA replication and division. The resulting cells are now called primary spermatocytes (diploid).

Primary spermatocytes then undergo a process known as meiosis, a type of cell division that produces four daughter cells, each with half the number of chromosomes as the parent cell. Meiosis is a crucial step in spermatogenesis, as it ensures that the resulting sperm cells have the correct number of chromosomes.

The resulting daughter cells are known as secondary spermatocytes (haploids). These cells then undergo another round of meiosis, resulting in the production of four round spermatids. Spermatids are the precursors to sperm cells, undergoing a series of changes to become mature sperm cells.

The final step in spermatogenesis is the transformation of spermatids into sperm cells, a process known as spermiogenesis. During spermiogenesis, spermatids undergo a series of morphological changes that result in the formation of a head, midpiece, and tail. The head of the sperm cell contains the nucleus, which contains the genetic material, while the midpiece and tail provide the necessary motility for the sperm cell to swim.

Given current trends of low fertility rates in highly industrialized countries, research in reproductive physiology should be prioritized[Ska+16]. Fertility problems might arise during spermatogenesis or the generation of spermatozoa from male germ cells.

Spermatogenesis is a complex process. Due to the similarity of mammalian testis structure and sperm development, human sperm production is usually modeled on mouse testes. Inside the testes in a given seminiferous tubule, around 4-5 generations of germ cells are in some stage of the spermatogenesis process. The process is synchronous and tightly controlled, spermatids at a given development step are always found with particular types of spermatocytes and spermatogonia. Thus, a given cell composition is known as a stage of the seminiferous epithelial cycle. In most mammalian species, including the mouse and human, twelve (I-XII) epithelial stages (7) can be identified [Oak56; Muc+13].

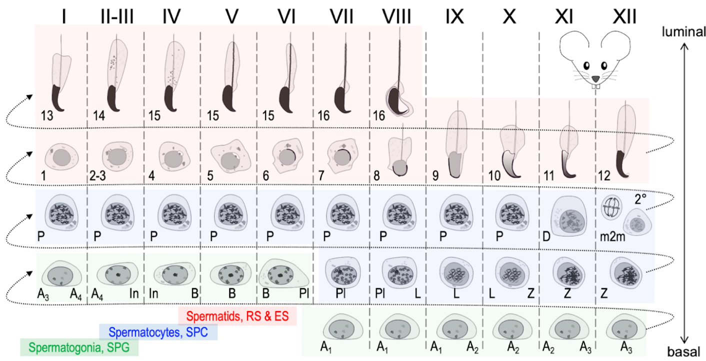

Figure 7: Spermatogenic cell differentiation and stages of the mouse seminiferous epithelial cycle. A spermatogonium that enters spermatogenesis has to pass through all the depicted developmental phases (from left to right, bottom to top) plus six mitotic and two meiotic divisions to be released as mature elongated spermatids. As spermatogenesis proceeds, spermatogenic cells move from the basal towards the luminal compartment. The progress of 4-5 generations of spermatogenic cells (in rows) is synchronized, and certain spermatids (highlighted in red) always associate with particular types of spermatocytes (blue) and spermatogonia (green). These cell associations are known as stages of the seminiferous epithelial cycle, and 12 (I-XII) different stages (columns) can be identified in the mouse. Any mouse seminiferous tubule cross-section presents one of these stages, and as spermatogenesis proceeds, it gradually develops to the following stage in numerical order. Spermatogonia can be further classified into type A1-A4, Intermediate (In) and type B, which present subsequent cell generations. Primary spermatocytes exist as preleptotene (Pl), leptotene (L), zygotene (Z), pachytene (P) and diplotene (D) spermatocytes. Meiotic divisions (m2m) and secondary spermatocytes (2o) can be found solely in stage XII. Postmeiotic germ cell differentiation, i.e. spermiogenesis, can be further divided into 16 steps (1-16).

In Section 3: Stagetool, we will introduce a deep learning method that characterizes the different types of cells and the stage of seminiferous tubules.

2. PRACTICAL SEGMENTATION

> "So let the pixels dance, in segmentation's embrace,
> An ode to algorithms, in the vast digital space."
>
> — chatGPT 3.5, AI language model

This chapter describes the article [Fis+21]: "Practical segmentation of nuclei in brightfield cell images with neural networks trained on fluorescently labelled samples". This article arose from a collaboration between the University of Tartu and the company then named PerkinElmer. PerkinElmer split into two in 2023, the life sciences and diagnostics business became Revvity, and the remaining business kept the PerkinElmer name.

Revvity is a big multinational company operating in 190 countries, and it is at the forefront of medical imaging device production and related software. Revvity collaborates closely with several pharmaceutical companies and knows which practical problems are more important for medical research.

The primary objective of the collaboration was to combine the university's scientific expertise in deep learning with Revvity's expertise in medical imaging. From the university side, Leopold Parts (main supervisor of this thesis) led the project, while from the Revvity side, Kaupo Palo was the main lead. Dmytro Fishman is the first author of the article, and at the time, he was the master supervisor of Sten-Oliver Salumaa, the second author of the article. My role in this article was to give a helping hand, and my main contribution was to evaluate how adding more focal planes improves segmentation.

2.1. Research Aim and Motivation

To determine the nucleus of cells, the typical approach is to use fluorescence microscopy by staining the sample with a dye that attaches preferentially to the cell nucleus. Light outside the visual range allows high-resolution images with a clearly segmented nucleus. However, the staining process requires substantial time and resources.

In contrast, bright-field microscopy is the simplest of all the microscopy techniques. The observation is done with white light, without the use of any dye, which considerably shortens the time and resources required. But the contrast between the nucleus and the rest of the cell is low, and it is very difficult even for humans to discern the nucleus well.

Previous work [Chr+18] had shown that bright field images contain enough information to transform them into fluorescence images with a reasonable amount of error. However, the segmentation of nuclei from brightfield cell images was not yet a standard part of an imaging workflow. Thus, to facilitate the adoption of it during our research, we wanted to answer several practical questions:

2.2. Goals

- Evaluate which neural architectures work best to segment cell nuclei from brightfield microscopy images.
- Evaluate how much data is required for cell segmentation, and whether adding data from multiple focal planes improves results.

2.3. Data

The data correspond to seven different cell lines: mouse fibroblasts (NIH/3T3), canine kidney epithelial cells (MDCK), human cervical adenocarcinoma (HeLa), human breast adenocarcinoma (MCF7), human lung carcinoma (A549), human hepatocellular carcinoma (HepG2), and human fibrosarcoma (HT1080). For all cell lines, both a fluorescent readout of a DNA-binding dye and a brightfield measurement were acquired. Figure 8 illustrates the different lines.

Ground truth pixel assignments were created semi-automatically from the dyed images by Harmony, Revvity's proprietary image analysis software.

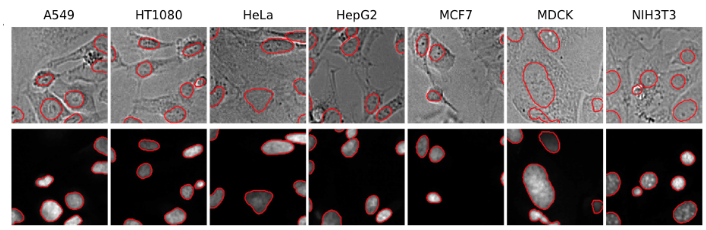

Figure 8: Brightfield Dataset. The top row is the brightfield image for each one of the cell lines, with the corresponding fluorescent image on the bottom row. Example image for each one of the cell lines. Ground truth segmentation in red

2.4. The best model

The first question that the paper addressed was which segmentation architectures were better to use on brightfield images. Three architectures commonly used at the time to segment nuclei in fluorescent microscopy images were evaluated:

- DeepCell [Van+16]
- U-Net architecture [RFB15]
- Mask R-CNN [He+17]

Since DeepCell is very slow (hundreds of seconds per image), we compared the methods only in the A549 cell line. Some qualitative results are shown in Figure 9A.

Quantitative results for the A549 cell line are shown in Figure 9B. The U-Net model demonstrated the highest performance on the A549 cell line, achieving an area under the receiver operating characteristic curve (AUROC) of 0.98 and an accuracy of 96%. In comparison, Mask R-CNN obtained a lower AUROC of 0.89 but maintained a comparable accuracy of 96%, whereas DeepCell achieved lower performance with an accuracy of 90% and an AUROC of 0.89. From these results, we concluded that U-Net was the best of the compared models.

Later, we check how well U-Net performs for all the cell lines. For reference, we trained the same architecture to segment fluorescence images. Figure 9C illustrates the results. On Brightfield images, model performance varied across cell lines, with accuracy ranging from 0.89 to 0.97 and F1-scores between 0.76 and 0.86. In contrast, performance on fluorescent images was both higher and more regular, with accuracy values between 0.98 and 1.00 and F1-scores between 0.95 and 0.99.

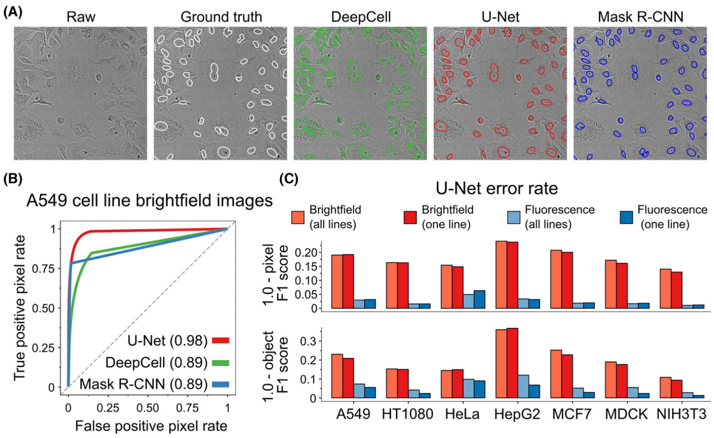

Figure 9: Neural networks can identify nuclei in brightfield images. (A) Example segmentation. A patch of a brightfield image from the A549 cell line (first image) annotated with ground truth nucleus boundaries (white, second image) determined from fluorescence data, as well as with predicted object boundaries for the three considered network architectures of DeepCell (green, third image), U-Net (red, fourth image), and Mask R-CNN (blue, fifth image). (B) Neural networks successfully segment nuclei from brightfield images of the A549 cell line. Pixel-level true positive rate (y-axis) and false positive rate (x-axis) at different score cutoffs for U-Net (red), DeepCell (green), and Mask R-CNN (blue). Grey dashed line: y = x; area under the curve for each method provided in brackets. (C) U-Net segmentation error (1.0 – F1 score; y-axis) for different cell lines (x-axis) using pixelwise (top) and objectwise (bottom) measures for models trained and applied on brightfield (red) and fluorescence (blue) channels, and models trained on one cell line at a time (light) as well as on all data (dark) of the corresponding acquisition type, using fluorescence-derived ground truth

2.5. Data requirements

In this section, we address the question of how much data is required to obtain a good performance for the U-Net model on brightfield images. Manually labeling a single image takes a considerable amount of time, since a typical image of 1080 × 1080 pixels consists of hundreds of cell nuclei. Thus, it is important to assess the amount of data needed to train the U-Net model on brightfield images.

We first tested models trained on data from a single cell line at a time. On average, 32 images are needed to achieve performance within 6% of the overall optimum, see Figures 10A and 10B. Thus, extensive manual labelling is required to train a new brightfield segmentation model.

Next, we tested how adding data from other cell lines affected performance. The performance on any single cell line was improved by including the same number of images from other cell lines during training, see Figures 10B and 10C, even when hundreds of training images were available for the target line.

Finally, leaving out the target cell line from the training set considerably increased the error on that cell line, see Figure 10D. Thus, the trained networks do not generalise to previously unseen cell lines.

2.6. Multiple focal planes

In brightfield microscopy it is possible to perform multiple acquisitions with different focal planes, since each one is quick and the specimen is not damaged. However, it is not clear if adding more focal planes will improve segmentation.

We evaluate this question in a dataset of prostate cancer cells with nine different focal planes in the brightfield channel (Figure 11A).

First, we trained a U-Net for each one of the planes. As illustrated by Figure 11B and 11C, the central ones (planes 4-5) had a higher error compared to the rest.

Next, we augmented the input layer of U-Net to accept a higher-dimensional input, leaving the rest of the architecture unchanged. Two planes improved the performance over a single one (Figure 11D, 11E), while including information from additional ones gave diminishing returns.

2.7. Conclusion

In this chapter, we have discussed how to employ deep learning to segment nuclei in brightfield images. When comparing three different methods (DeepCell, Mask R-CNN, U-Net), we conclude that, although nuclei can be segmented with the other models, U-Net results are superior (Figure 9B).

U-Net is a faster and simpler method than the other alternatives. However, a moderate number of brightfield images, annotated with ground truth objects obtained from fluorescence signal, is necessary to train a U-Net. Specifically, at

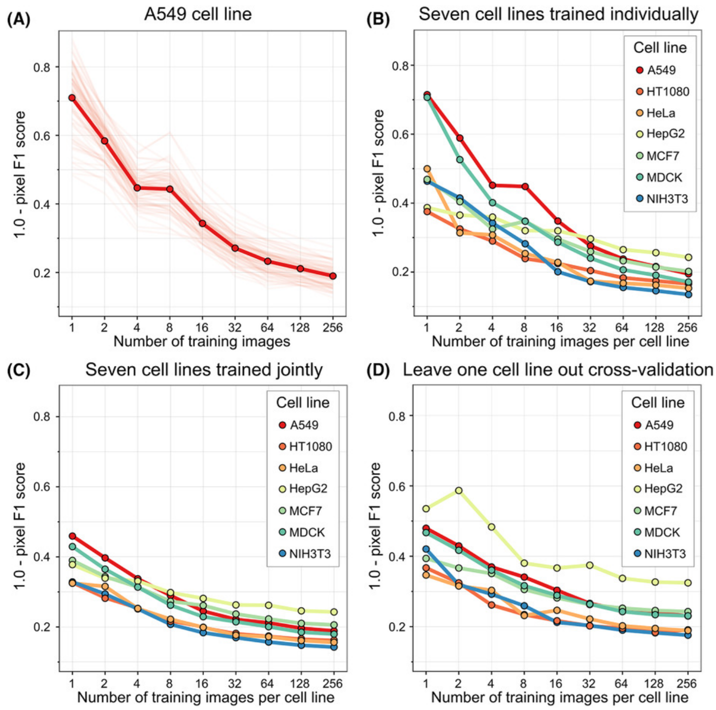

Figure 10: Segmentation performance versus training set size. Pixel-level segmentation error (1.0 – pixel F1 score; y-axis) for increasing number of training images per cell line (x-axis). (A) U-Net trained on A549 cell line only, performance measured on individual test images (light lines), and their average (dark line and markers). (B) As (A), but averages only for models trained separately on the seven different cell lines. (C) As B, but for models trained on a training set that includes the same fixed number of images from each cell line. (D) As C, without including the tested line in the training set. The total number of training images is seven times larger than the x-axis value in (C), and six times larger than the x-axis value in (D)

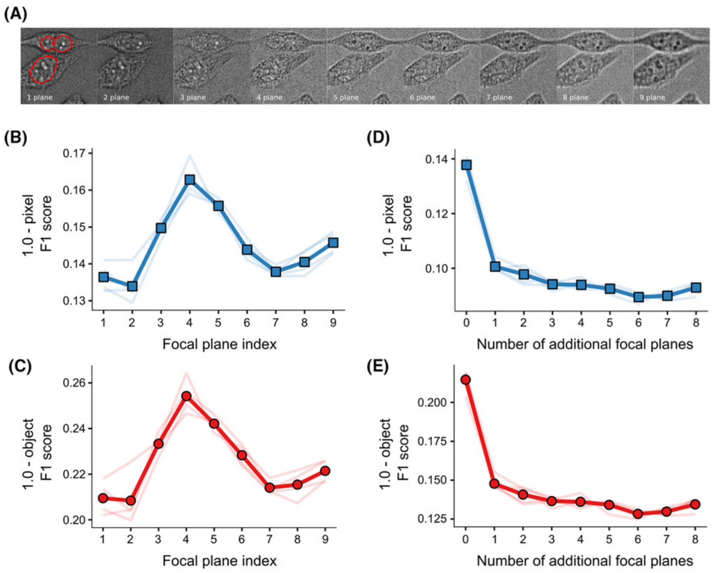

Figure 11: Multiple brightfield planes. (A) Example brightfield acquisitions of the same specimen from nine focal planes, ranging from top (left) to bottom (right) in 1 micrometre steps. Red line: contour of the nucleus, as segmented from the fluorescence channel. (B-C) Pixel and object errors depend on the focal plane. Error of five random training restarts (thin lines, y-axis) and their average (solid line) for U-Net models trained on data from different focal planes (x-axis) for pixels (blue) and objects (red). (D-E) Two focal planes are sufficient according to both pixel and object-level performance. Error of five random training restarts (thin lines, y-axis) and their average (solid line) for U-Net models trained on data with different number of input planes (x-axis) for pixels (blue) and objects (red).

least 32 images are needed to achieve performance within 6% of overall optimum (Figures 10A and 10B)

Finally, the single plane segmentation can be greatly improved upon by additional acquisitions, and two planes might be the optimal number, since additional planes increase the results minimally. There are multiple possible reasons for this. One possibility is that since the different focal planes capture different distortions of light, observing a variety of them could help to identify true structure in the signal. Another explanation is that two acquisitions theoretically capture the light phase, which can be informative for accurate segmentation.

Overall, we believe that microscopy pipelines can benefit from the use of brightfield images with lightweight deep learning models like U-Net, given the ease of acquisition of the microscopy images, the low computational resources of some deep learning models and, more importantly, the promising results shown here.

3. STAGETOOL

> "It is a well-documented fact that guys will not ask for directions. This is a biological thing. This is why it takes several million sperm cells... to locate a female egg, despite the fact that the egg is, relative to them, the size of Wisconsin."
>
> — Dave Barry

This chapter describes the article: "STAGETOOL, a novel automated approach for mouse testis histological analysis". The article was carried out in cooperation with Oliver Meikar from the Institute of Biomedicine and Translational Medicine (University of Tartu). He has a deep knowledge of spermatogenesis, the process of generation of spermatozoa from male germ cells, and of bioinformatics. Besides Oliver, this article was possible thanks to the knowledge of the coauthors from the Institute of Biomedicine, University of Turku, Finland and their invaluable contribution to the project.

My personal contributions to this article were to develop the cell classification network, model evaluations, and the first version of the scientific manuscript.

3.1. Research Aim and Motivation

Spermatogenesis is the process by which sperm cells are produced from male germ cells (for a summary see the background section 1.4). Spermatogenesis is a complex process at the cellular and molecular levels, involving numerous tightly regulated steps. Advances in microscopy and imaging technologies have made it possible to generate vast amounts of experimental data, capturing this process in great detail. However, transforming these data into meaningful biological insights is hindered by a critical shortage of specialists with the histological and analytical expertise required to interpret subtle morphological patterns within testicular tissue. Consequently, although data availability is no longer a limiting factor, the lack of trained experts capable of performing reliable and consistent histological assessments has become a major bottleneck in understanding the mechanisms underlying male infertility.

Previous literature [Xu+19; Xu+21] pointed out that it was possible to classify tubule stages in Hematoxylin and eosin (H&E) stained microscopy images (colors cell nuclei blue-purple with hematoxylin and cytoplasmic and extracellular components pink with eosin) and differentiate between the classification of three cell types in stage VI-III tubules (See 1.4 for stages description) .

The main question behind the Stagetool article was to see if it were possible to classify the different cell types and tubule stages with a Deep learning model for DAPI-stained microscopy images.

Fluorescent DAPI staining allows visualization of small differences in chromatin density. Thus, DAPI might facilitate highly accurate cell identification by morphology, and tubule stage identification is also determined by cell composition. Thus, it was possible that combining DAPI and deep learning might allow automated testis histological analysis.

However, cell identification by experts not only relies on morphology, since it also depends on cell position in the seminiferous tubule. Thus, before this research, it was not clear if it was possible to classify well the different cell types and tubule stages with deep learning.

3.2. Data

The image data was generated at the University of Turku in Finland. Four whole-testis cross-sections, plus areas from three additional testis cross-sections with rare developmental stages, were selected to generate training and validation data for convolutional neural network models.

Ground truth segmentations were generated for the seminiferous tubules. The developmental stages of seminiferous tubules were divided into 5 categories: I-V, VI-VIII, IX, X-XI, XII. Nine cell types were selected for classification: Sertoli cells (Sertoli), spermatogonia (Spg), prepachytene spermatocytes (Spc prePach; including preleptotene, leptotene, and zygotene spermatocytes), pachytene and diplotene spermatocytes (Spc Pach), meiotic divisions (m2m), secondary spermatocytes (Spc sec), round spermatids (Round spt), early elongating spermatids (El spt early) and late elongating spermatids (El spt late).

3.3. Stagetool models

Stagetool combines two models: a tubule model and a cell model. The tubule model segments and classifies tubules into five developmental stages. The cell model detects and classifies cells into nine different types. The output of both models is combined to obtain a histology analysis of a given image. Figure 12 provides an overview of Stagetool.

To implement and evaluate the models, we adopted FAIR's (Facebook AI Research) Detectron 2 object detection and segmentation platform version 0.2.1. Table 1 presents an overview of both models. For a detailed enumeration of all the parameters, we refer the reader to the original article.

Model	Architecture	Backbone	pre-training
Tubule	Mask R-CNN	ResNet-50	ImageNet
Cell	Faster-RCNN	ResNeXt-101	No

Table 1: Summary of the architectures that compose Stagetool

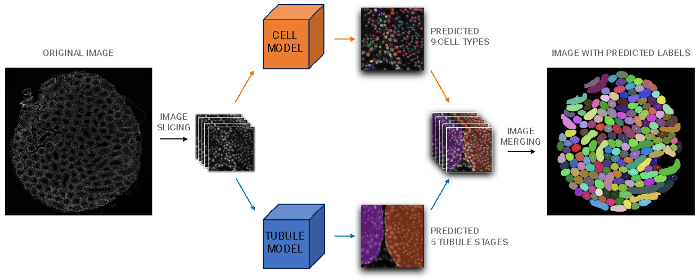

Figure 12: Overview of the Stagetool pipeline. The whole-testis DAPI-stained cross-section is partitioned into 1024×1024 subimages, which are analyzed using two neural network models: a cell model and a tubule model. The cell model identifies and classifies nine mouse cell types, while the tubule model detects, segments, and classifies tubules into five developmental stages. The full testis image is then reconstructed to include both tubule and cell annotations.

3.4. Results

First, we evaluate the Stagetool performance individually for both models; later we evaluate the performance of the model on a whole cross-testis section. Figure 13 presents an example of Stagetool individual models output.

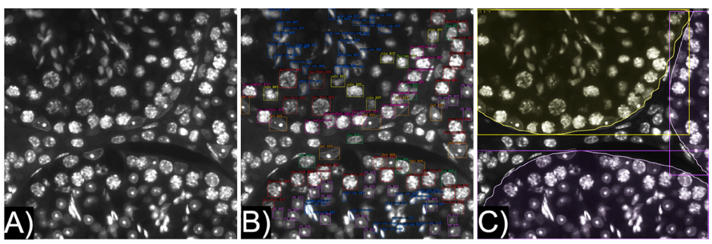

Figure 13: A) Input image, B) Cell model output, C) Tubule model output

3.4.1. Tubule model evaluation

This evaluation employs the same four evaluation metrics: pixel accuracy, mean accuracy, mean IoU, and frequency weight IoU, as described in [LSD15]. The Tubule model quantitative results are shown in Table 2. Although the model only receives as input a small percentage of any given tubule, it obtains a mean accuracy of 0.8551.

3.4.2. Cell model evaluation

Mean average precision Mean average precision (AP), as defined in the COCO object detection challenge [PND20], was calculated. The model obtained an AP50

Pixel accuracy	0.9180
Mean accuracy	0.8551
Mean IOU	0.7423
Frequency weight IOU	0.8569

Table 2: Tubule model quantitative results

of 78.99 and AP75 of 50.00 [LSD15].

AP reflects how well the ground truth boxes correspond to the predicted ones. However, beyond exact box coordinates, it was important to assess how well the model detected and classified cells for each category. Thus, for a fixed intersection over union Intersection Over Union (IOU) 0.5, the performance was evaluated. Table 3 presents the results obtained for each cell type.

When averaged across classes, the positive predictive value is 0.9607, pointing out that the model could differentiate appropriately between classes. Sensitivity was, in general, high with an average across classes of 0.8402, but lower than the positive predictive value. Late elongating spermatids stood out due to their lower sensitivity, with around 68% of the ground labels detected. This lower percentage of detection might have been due to their shape, which was ill-suited to fit the box, and their small size.

	Precision	Sensitivity	F1-score
Round spermatids	0.9947	0.9033	0.9468
Early elongated spermatids	0.9243	0.7680	0.8389
Late elongated spermatids	0.9611	0.6829	0.7985
Pre-pachytene spermatids	0.9390	0.8964	0.9173
Pachytene and diplotene spermatocytes	0.9782	0.9352	0.9562
Metaphase plates	0.9500	0.8261	0.8837
Secondary spermatocytes	0.9906	0.8203	0.8974
Spermatogonia	0.9152	0.7593	0.8300
Sertoli cells	0.9934	0.9700	0.9815

Table 3: Metrics for 0.5 Intersection Over Union (IOU)

Figure 14 provides the confusion matrix, which shows the incidence of erroneous cell labels for any of the 9 particular seminiferous epithelial cell types. The most frequent errors occurred between (1) Spc-prePach and Spg, which can be difficult to distinguish without antibody staining, and (2) elSpt-early and elSpt-late, where the former transitions gradually into the latter.

3.4.3. Whole-testis cross-section evaluation

We tested Stagetool over a whole-testis cross-section, excluded from the training and validation sets. The cross-section image was divided into 1024x1024 pixels, and 400x magnified mouse testis images. Every image was input to both the cell and tubule models.

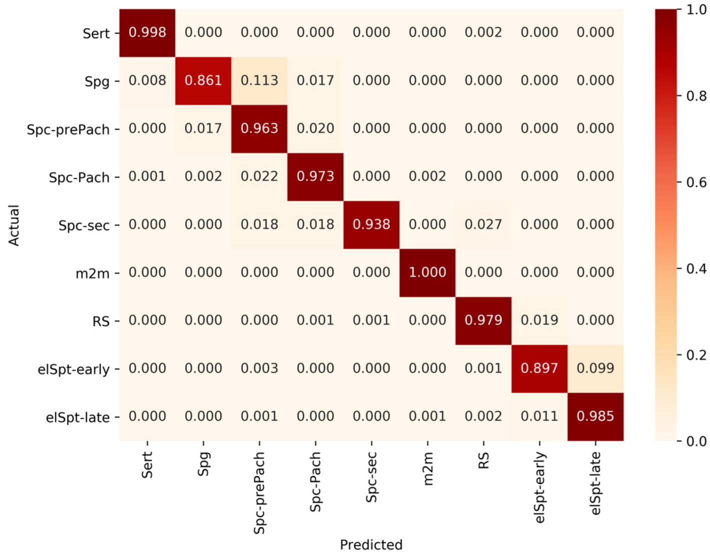

Figure 14: Confusion matrix for the 9 seminiferous epithelial cell types.

For the Tubule classification, since each tubule class is predicted across several images, as illustrated in Figure 15, a weighted majority algorithm merges the predictions for each tubule. For each mask, the prediction weight was calculated as the product of the predicted class confidence and the mask's proportion of the tubule area. This decreased the prediction weight of small masks.

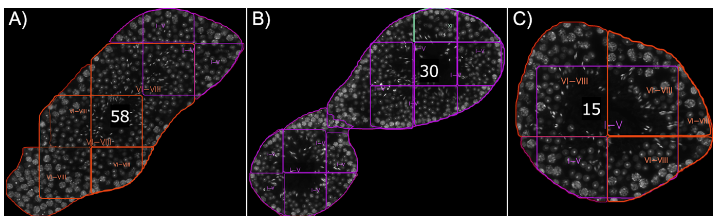

Figure 15: Visualisation of sub-predictions for tubule cross-sections for three tubules. Each tubule is represented by several overlapping images, which have individually predicted tubule borders and classes.

The results for stage tubule prediction were compared with human annotations by a group of experts. Results are presented in Table 4. Stage classification was quite similar. A few examples of disagreements between predictions and human annotations are provided in Figure 16.

Total number of tubules	In agreement	In disagreement
112	107 (95.6%)	5 (4.5%)

Table 4: Human annotations vs. Computer predictions

For the cell model, each image was analyzed 4 times, at every 90 degrees rotation angle of the image, and the results were merged by majority vote. To

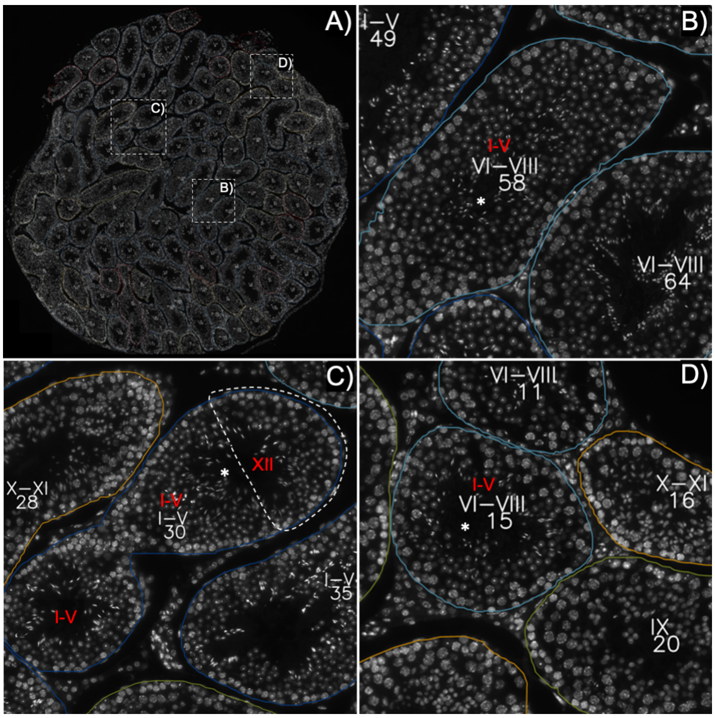

Figure 16: . A) A whole-testis cross-section with Stagetool-derived staging data; white dotted rectangles are shown as higher magnification insets in B-D). When human annotation is in conflict with the computer prediction, it is shown in red.

assess the performance of Stagetol, we studied the distribution of cell labels in a given stage. The results are provided in Figure 17. The distributions obtained were biologically correct and expected.

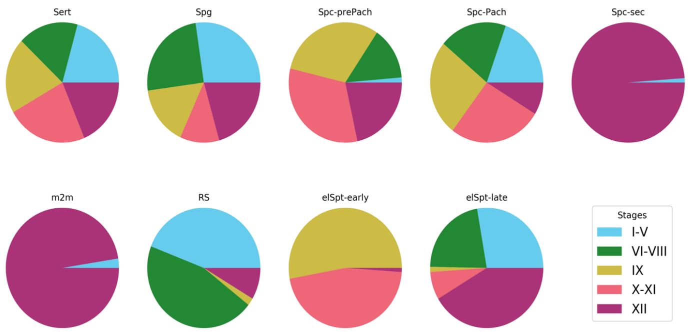

Figure 17: Sertoli cells were evenly distributed in stages, whereas secondary spermatocytes (Spc-sec) and metaphase plates (m2m) were seen in stage XII, and occasionally in stages I-V

3.4.4. Stagetool robustness and KO testis analysis

Stagetool performance was tested on several seminiferous tubule images from Knockout mice with a spermatogenic defect. The images originated from different laboratories and were generated by different instruments, and different mouse ages. Image quality was disparate, and the resolution differed from the training data in a few cases. All images were converted to gray-scale since it is required by Stagetool.

Stagetool was applied to several images from Miwi KO mice, which are characterized by spermatogenic arrest at the round spermatid stage and a lack of elongating spermatids [DL02]. Lack of elongating spermatids makes staging for Miwi AP testis challenging, since the presence and distribution of elongating spermatids that define the stages for a human histologist. Figure 18A-B illustrates representative examples of Stagetool output for Miwi AP mice. From the 856 cells predicted, only seven were incorrectly categorized as elongating spermatids. Furthermore, tubule staging was correct in Miwi AP mouse line, indicating that the model does not require elongating spermatids to determine seminiferous epithelial stages.

Stagetool was also applied to several low-resolution and partially overexposed images of paraffin-embedded seminiferous epithelium cross-sections from Spef2 AP mice. As illustrated by Figure 18C Stagetool was able to characterize cell types and tubule stages highly precisely.

3.4.5. Antibody expression profiling

Stagetool could be used for automated profiling of antigen expression by combining DAPI staining with antibody labeling. Many antigens have a restricted

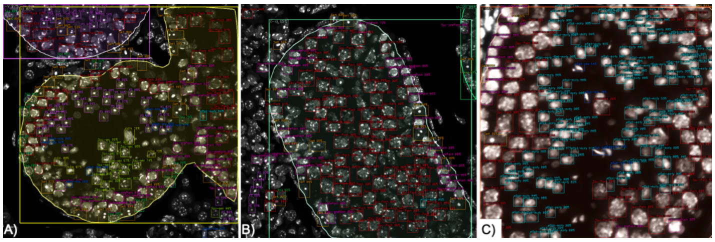

Figure 18: Two representative examples of Stagetool output for Miwi KO seminiferous tubules at stages A) I-V and XII, and B) X-XI. Despite the lack of elongating spermatids Stagetool correctly recognizes the stages and cell types therein. C) A representative example of Stagetool output for low-resolution and over-exposed Spef2 KO seminiferous tubules at stage IX.

expression in spermatogenic cells, and their expression is often limited to particular cell types and specific stages of the seminiferous epithelial cycle.

We stained the mouse testis cross-section with DAPI and the following antibodies:

1. SOX9 (SRY-Box Transcription Factor 9). Expressed in Sertoli cells independently of the epithelial stage [Da +96].
2. SALL4 (Spalt Like Transcription Factor 4), pan-spermatogonial marker [Cha+17].
3. SCP3 (synaptonemal complex protein 3). Expressed in spermatocytes [Yua+00; Par+16].
4. CREM (cAMP responsive element modulator). Expressed in round spermatids [Ble+96; Nan+96; BW01].
5. PNA (peanut agglutinin). Expressed in round and elongating spermatids [Mäk+20].
6. AR (androgen receptor). Expressed in Sertoli cells. High level expression in early-to-mid stages and lower in late stages (IX-XII).
7. USF1 (Upstream stimulatory factor 1). Expressed in Sertoli cells independently of the epithelial stage. Also expressed in undifferentiated and early differentiating spermatogonia [Fai+19].

To obtain the expression profile for each antigen, Stagetool was applied to the cross-section image containing only DAPI information. Then, Stagetool predictions were overlaid upon each one of the antibody-stained images. A luminosity threshold was fixed empirically, and predicted cells with a mean luminosity above the threshold, and 65% of the pixels above half the threshold were considered positive for antibody expression.

Figure 19 provides the expression profile obtained for the studied antigens. Stagetool derived antigen expression data confirmed the expected cell profile: Sertoli cell expression pattern for SOX9, AR, and USF1; SALL4 spermatogonia expression, SCP3 for spermatocytes; CREM round spermatids; and PNA round and elongating spermatids. Furthermore, for Sertoli cells, AR expression was high in early-to-mid stages and lower in late stages (IX-XII). In contrast, USF1 expression was uniform across all stages in Sertoli cells and was also expressed in a small subset of spermatogonia. The expression profiles obtained were in agreement with spermatogenesis literature.

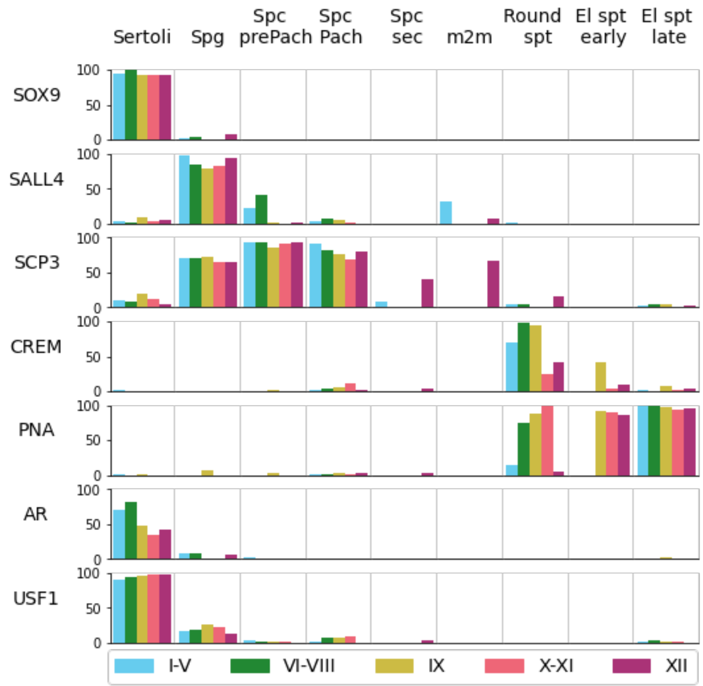

Figure 19: Percentage (bar height) of predicted cells type (columns) of a given tubular stage (color) labeled by the corresponding antibody (row). Cell categories from left to right correspond to Sertoli cells, spermatogonia (Spg), pre-Pachytene spermatocytes (Spc prePach), Pachytene and diplotene spermatocytes (Spc Pach), Secondary spermatocytes (Spc sec), Metaphase plates (m2m), Round spermatids (Round spt), Early (El spt early) and Late elongating spermatids (El spt late).

3.5. Discussion

3.5.1. Summary and Interpretation

This work aimed to develop a new automated method to analyze testis histology images. Previous work had classified tubule images in different stages in H&E stained images. However, our method, called Stagetool was developed to classify both tubule stages and individual cells in DAPI-stained testis cross sections.

For seminiferous tubule classification, Stagetool achieved 95.6% accuracy on a whole testis cross section. Thus, Stagetool seems a very reliable method for the recognition of five stages (I-V, VI-VIII, X, X-XI, and XII).

Beyond tubule classification, Stagetool is also able to detect nine cell types. Remarkably, when averaged across nine classes, the positive precision is 0.9607. Before Stagetool, analyzing the individual cells in a whole testis cross-section was a daunting manual task, since a cross-section might contain upwards of fifty thousand cells. Testis histology analysis is typically performed at the tubule level, but our method might facilitate more cell-level research of male infertility.

We also tested Stagetool reliability for AP mice studies. Despite the lack of elongating spermatids for Miwi AP mice, Stagetool correctly recognized the stages and cell types. Stagetool also worked correctly for low quality images of Spef2 AP mice.

Finally, we wanted to assess if Stagetool might facilitate antibody expression analysis for spermatogenesis. Therefore, we proceeded to test seven known antigens whose expression is limited to particular cell types and stages of spermatogenesis. For all of them, the expression profiles obtained were in agreement with the corresponding literature.

Overall, we believe that Stagetool can be a useful tool for basic research in male fertility, and the results obtained corroborate it.

3.5.2. Limitations

Any method can always be improved, and some of the current limitations of Stagetool are:

1. In the mouse, 12 different stages (I-XII) of the seminiferous tubules can be identified. However, Stagetool groups similar stages together and predicts only five categories (I-V, VI-VIII, IX, X-XI, XII).
2. The whole-testis cross-section image analysis requires some manual curation to remove unsuitable seminiferous tubules from the segmentation output image. This is because the tubular arrangement of the whole testis often contains artifacts from sample preparation or longitudinally cut tubules, which may interfere with downstream analyses.
3. The cell model takes as input the original image plus three rotated images, to obtain better results. This increases computational overhead, which might be avoided in the future with a model equivariant to rotation.
4. Although the cell model was very precise in separating the different classes, the model has low recall in some classes. I believe this is a general limitation of this type of deep learning models, and was the inspiration for the work described in section 4.

3.5.3. Future work

Future work might explore what criterion Stagetool employs to determine the seminiferous tubule stages. Human experts employ a fixed set of rules, and any difference might shed light on new ways to characterize the seminiferous epithelial cycle.

Of paramount importance might be to extend Stagetool to tackle human spermatogenesis. Infertility affects 10-15% of couples, with half the cases caused by male infertility [TRR18]. Therefore, there is a need for better diagnostics in male fertility. An automated analysis tool for characterizing spermatogenic failure in humans would facilitate counseling and treatment strategies for male infertility.

An approach to improve Stagetool is to change the object detection models, i.e. by employing Swin Transformer [Liu+21] as the backbone for the models. The transformer architecture [Vas+17] has shown impressive results in a variety of tasks. However, it is unclear if Stagetool ground truth training data will suffice to train a transformer architecture, since it has a comparatively larger number of parameters than convolutional models.

Changing the backbone architecture will probably not address entirely the lack of recall of some cells. Computer vision methods currently employed tend to omit some objects, which humans can easily spot. Thus, there should exist some method that avoids missing any object, and following this idea, we present an alternative approach to object detection in the next section.

4. KAIZEN

> “The purpose of todays training is to defeat yesterday’s understanding”
>
> — Miyamoto musashi

Questioning established assumptions within a research domain can lead to valuable insights. Although conventional wisdom is typically well substantiated, it also sets the conceptual boundaries that guide most researchers. Consequently, departing from these usual premises may uncover novel perspectives.

After several years of experience applying deep learning techniques to microscopy image analysis, the thesis author had acquired expertise regarding the prevailing best practices in the field. At that juncture, the thesis author sought to challenge these established methodologies. This endeavor culminated in the work presented in this chapter: “Kaizen: Decomposing Cellular Images with VQ-VAE.”

4.1. Research directions

> “Generative models for sensory inputs are doomed to failure”
>
> — Yann LeCunn

Several loosely connected ideas were the points of departure for the work exposed in this chapter:

- **Generative models in the brain**. This research was particularly inspired by the mechanisms underlying human perception. After years of working extensively with computers, I developed mild visual impairment in one eye, which occasionally led me to misperceive far away objects—for instance, mistaking a tree for a person. These experiences raised questions about the generative nature of perception: the notion that the brain constructs, rather than merely receives, our sensory experience of reality.
- **Sequential processing**. The common perception is that parallelizing the detection of objects is superior, as it enables the use of additional computing power in a more efficient manner. Are there potential benefits to conducting object detection in a serial fashion? This question was already explored in the thesis of Marharyta Domnich’s master’s thesis[Dek19], which I had the opportunity to supervise.
- **Intelligent Inference**. In deep learning, considerable computation is invested in the training stage. In contrast, inference presents reduced computational requirements. Yet, for a computer to perform "intelligent" thinking and exhibit generalization, it likely needs to execute sophisticated algorithms during the inference stage. As a result, it is logical to explore options that increase the computational load in the inference stage.

- **Beyond Segmentation and bounding boxes**. Why use bounding boxes to detect objects in an image? Why classify pixels in an image to identify and separate objects in an image? These tasks are suited to the capabilities of methods from years ago. A bounding box does not accurately represent object shape, although it is well-suited for classifier models. Regarding segmentation, it is possible to have overlapping objects or partially occluded objects, making the task differ from an ideal computer vision algorithm.
- **Machine Oversights**. Why can current computer vision models not detect all objects in an image? It is generally agreed upon that there is a trade-off between precision and recall; therefore, any method will miss some objects. But for a person, it is obvious when there is a missing object prediction in an image. What prevents current methods from recognizing instances where they failed to predict an object?

4.2. Goals

- Propose a fundamentally new way of analyzing microscopy images by decomposing them into individual cells rather than segmenting pixels or drawing bounding boxes.
- Investigate whether a generative approach could overcome some limitations of existing segmentation methods (e.g., errors with occlusions, overlapping cells, or missed detections).

4.3. Methods

Following all the ideas introduced in the previous section, a new method, Kaizen was created. Kaizen employs a generative model (VQ-VAE described below). The generative model is trained to encode individual cells. During inference, the generative model is applied sequentially to generate individual cell predictions on an image. As shown in Figure 20 the predictions merge to form a global internal image. New predictions are only accepted if they improve the similarity between external input image and the internal image. Kaizen avoids the oversight of objects (high recall) by halting only when the internal image matches the external one. Kaizen, instead of segmenting or using bounding boxes decomposes the image into individual objects.

4.3.1. VQ-VAE

To implement Kaizen, we tried diverse models of autoencoders. We settled for a VQ-VAE since we obtained good qualitative results in our initial assessment compared to other autoencoders. A vector quantised-variational autoencoder, or a VQ-VAE, is an autoencoder that encodes samples in a discrete latent space. The discrete latent space is used to recover embeddings from an internal memory table.

Figure 20: (a) Distinctly colored squares denote separate regions extracted as inputs to the VQ-VAE, each used to reconstruct the central cell within its respective patch. The reconstructed cells are combined to form an internal image (b), which represents the cumulative set of predicted cells so far. Subtracting this internal image from the original microscopy image generates an error image (c). New prediction points, shown as crosses, correspond to local maxima of the error image, indicating regions requiring further refinement. The prediction highlighted with a red cross is rejected, as adding a cell at that position increases the total reconstruction error.

These embeddings are then fed to the decoder, which reconstructs the original samples. For a more detailed description of a VQ-VAE, we refer the reader to the background section 1.2.8.

We wanted the VQ-VAE to encode only single cells. Therefore, the VQ-VAE was trained with an output different from the input. First we selected small image patches that contained a cell in the patch center (selecting points at random form the ground truth). The input consisted of a small patch of a microscopy image that contains a cell in the patch center but might contain also a few other cells. In contrast, the output consists of a single cell present in the patch, the one touching the central pixel (multiplying the segmentation ground truth with the input patch).

Our objective was to ensure that the VQ-VAE encoded only individual cells. To this end, the model was trained using intentionally distinct inputs and outputs. We began by randomly selecting small image patches from the dataset, each containing a cell centered within the patch (any part of the cell touching the central pixel according to the ground truth). Thus, the input to the network was a microscopy image patch centered on a cell, which might also include other adjacent cells. The target output, however, consisted solely of the single cell that overlapped the central pixel. This was produced by multiplying the central cell segmentation ground truth with the corresponding input patch. As shown in Figure 21, after training, the VQ-VAE encodes a single central cell from small patches containing multiple cells while disregarding other cells and noise present on the input.

4.3.2. Algorithm

Using the VQ-VAE that encodes individual cells described above as a predictor and taking inspiration from predictive coding in the brain, the core algorithm of

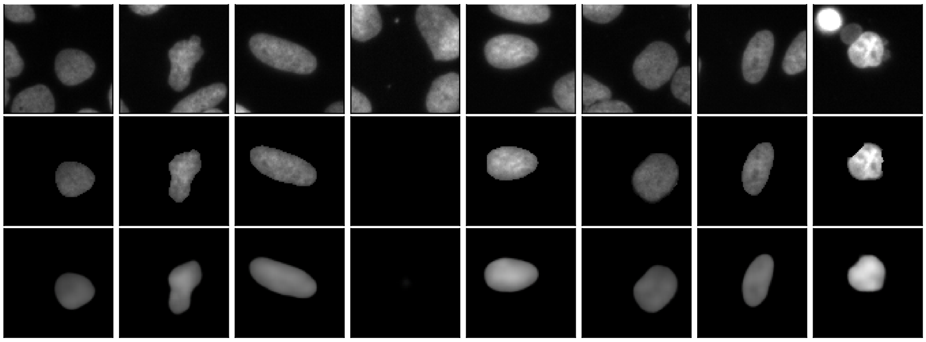

Figure 21: Samples of the VQ-VAE encoding individual cells for the U2OS dataset.The first row corresponds to training image patches employed as input to the VQ-VAE. In the second row, under each input, the corresponding ground truth is shown. Finally, the third row shows the corresponding VQ-VAE output after training.

Kaizen was implemented. Algorithm 2 is a simple evolutionary algorithm where the fitness function is the loss between the input and a global internal image (L1 loss). At each iteration, several points are proposed where the loss is maximal and spatially distant from each other. Then, at each proposed point in the image, an object prediction is made. Finally, only predictions that diminish the loss when added to the internal prediction are kept.

In practice, the selection of spatially distant high-loss points is performed in parallel by convolving the error image with a 7×7 kernel of ones using a stride of one, thereby emphasizing prominent error regions. The pixel corresponding to the maximum value in the resulting convolved image is selected as the first candidate point. To enforce spatial separation, a 32×32 pixel area surrounding this point is subsequently set to zero before the process is repeated to identify additional candidate locations.

Algorithm 2 Core algorithm

Initialize:

$internal \leftarrow zeroes$

$loss_{inter} \leftarrow Loss(internal, image)$

$error \leftarrow image$

repeat

$\{p_1, p_2, ..., p_N\} = distant\ points\ of\ max(error)$

$pred_p$ = $predict\ at\ each\ point\ p\ in\ image$

$loss_p = Loss(internal + pred_p, image)$

for $any\ pred_p\ where\ loss_p < loss_{inter}$ **do**

$internal \leftarrow internal + pred_p$

end for

$loss_{inter} \leftarrow Loss(internal, image)$

$error \leftarrow image$ - $internal$

until $all\ loss_p \geq loss_{inter}$

4.3.3. Predicting on the error

Sometimes it is difficult to separate an individual cell on a microscopy image due to the high density of neighboring cells. To tackle this issue, generating an internal global prediction of the external image is helpful. At a given point in time, we can subtract the internal prediction from the external image, obtaining an error image of what remains to be predicted. When predicting on this error image, we ignore the cells already predicted, simplifying new predictions.

Nonetheless, there is an inherent drawback to this approach. A prior erroneous prediction could lead to mistakes in subsequent predictions. Consequently, the process of predicting on the error image may result in compounding inaccuracies. To mitigate error compounding, we use algorithm 3. Kaizen predictions are made on the external image with the core algorithm until no further predictions are possible; at that point, an error image is generated. Then, predictions are made on the error image until no further predictions are possible, and a new error image is generated. The algorithm continues predicting on new error images until there is an error image for which no predictions are possible.

4.4. Datasets

U2OS dataset: Fluorescent microscopy images from U2OS cell lines. Image set BBBC039 version 1, available from the Broad Bioimage Benchmark Collection [LSC12]. The ground truth consists of individual nucleus instances. We designed our network on this dataset. Although the ground truth might contain some errors, we did not alter the ground truth. We randomly selected 100 images for the training set, 50 for the validation set, and 50 for the test set.

Neuroblastoma dataset: Fluorescent images of cultured neuroblastoma cells,

Algorithm 3 Predicting on error

Initialize:

$internal \leftarrow zeroes$

$error \leftarrow image$

$allpreds \leftarrow EMPTY$

repeat

$preds = CoreAlgorithm(error)$

$internal \leftarrow internal + preds$

$error \leftarrow image$ - $internal$

$allpreds \leftarrow allpreds + preds$

until $preds = EMPTY$

available from the Cell Image Library[Yu+19]. The ground truth consists of manually annotated cell boundaries. Although the ground truth might contain some errors, we did not alter the ground truth. However, since the ground truth size was half the weight and height of the images, we shrunk the images with cubic interpolation to match the ground truth size. We randomly selected 71 images for the training set, 12 for the validation set, and 17 for the test set.

4.5. Results

Kaizen decomposes an image into object representations, including superpositions of objects or occlusions. However, to the best of our knowledge, no comparable methodology for decomposing microscopy images has been proposed in the literature. Thus, to test our methodology, we were obliged to modify our method to perform instance segmentation. Specifically, the predicted objects are converted to binary masks by setting a minimum brightness threshold (ten percent of the input image average), such that all the pixels above it are one and below it zero. Such an approach might understate Kaizen results, but it obtains a reasonable comparison to the current methodology.

First, we evaluate Kaizen on a U2OS dataset of cell nuclei fluorescent microscopy images. To train the VQ-VAE, image patches of 40x40 pixels were generated, with eighty percent of them containing a cell touching the patch center. The VQ-VAE was then trained to code representations of individual cell nuclei as illustrated in Figure 21 and described in 4.3.1. Then Kaizen was applied to 50 left-out images from the dataset. Figure 22 illustrates qualitative results for several test image are illustrated in Figure 22.

Regarding numerical results, we compare Kaizen to the Stardist model [Sch+18], because it was specifically designed to predict cell nuclei in the same data type, and it is superior to more general approaches like U-Net [RFB15] or Mask R-CNN [He+17]. The numerical evaluation in Table 5 shows that our method obtains superior average precision across all the thresholds.

Next, to assess Kaizen in more complex images, we evaluate it on a Neuroblas-

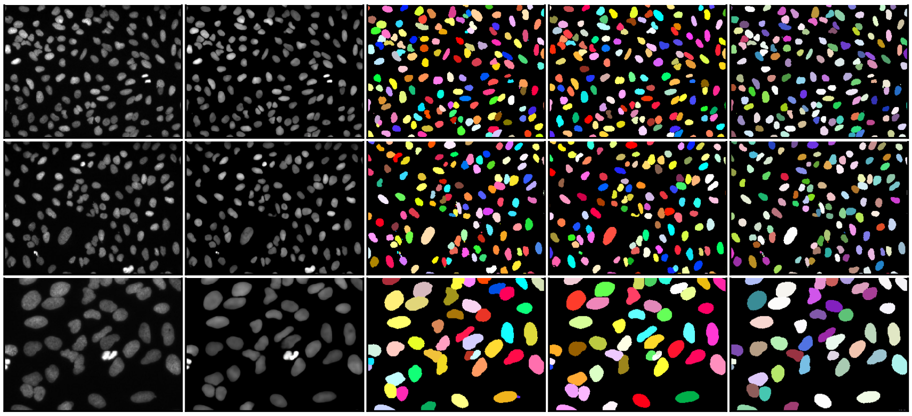

Figure 22: Examples of Kaizen segmentation for the U2OS dataset. The first two rows correspond to two dataset images, while the last row magnifies an image region. Panel b) shows the internal image reconstruction created by merging the individual nuclei generated by the VQ-VAE. Panels d) and e) show Kaizen and Stardist corresponding image segmentation.

toma dataset of fluorescent microscopy images. In contrast to the U2OS dataset, the entire cell is predicted, including the cytoplasm. To account for the increased individual prediction size, we increase the input size of the VQ-VAE to 120x120 pixels. Kaizen was applied to 17 left-out images from the dataset. Qualitative results are illustrated in Figure 23.

For the numerical results on the Neuroblastoma dataset, we compare Kaizen to Cellpose [Str+21], since it was designed with this specific dataset. Numerical evaluation is presented in Table 5. Both models obtain worse results than expected; this might be the result of chance, since the 17 left-out images seem difficult compared to the typical image in the dataset. Kaizen obtains superior average precision for the three lowest thresholds. However, it falls behind for the 0.8 and 0.9 thresholds. Upon inspection, average precision decay for higher thresholds might be related to the conversion from object images to binary masks.

4.6. Discussion

In this work, we have employed a generative model to decompose microscopy images into individual cells. The combination of the generated individual cells in an internal global image provides useful error feedback. The error feedback offers multiple possibilities. First, the error provides a signal to halt, making the computing time proportional to the number of cells in the input. Empty images are processed instantly, while more complex images are allowed as much computation as they require. Second, the error provides information about the location on the

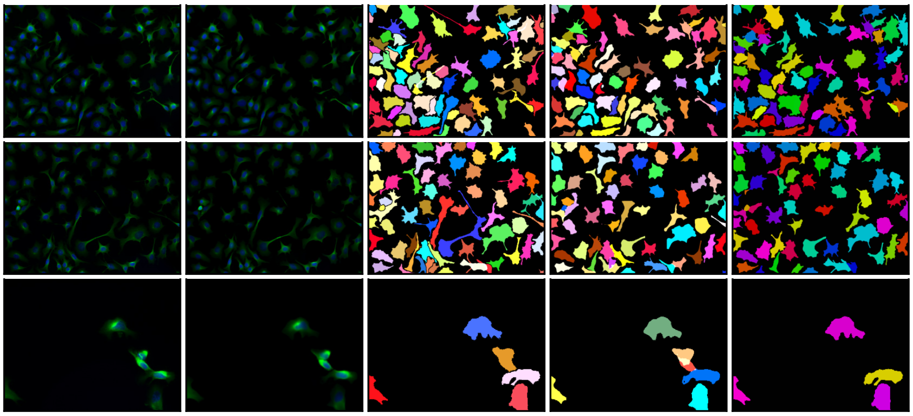

Figure 23: Example of Kaizen segmentation for neuroblastoma dataset. The first two rows correspond to two dataset images, while the last row magnifies an image region. Panel b) shows the internal image reconstruction created by merging the individual cells generated by the VQ-VAE. Panels d) and e) show Kaizen and Cellpose corresponding image segmentation

Threshold	0.5	0.6	0.7	0.8	0.9
			U2OS nuclei		
Stardist	0.908	0.889	0.864	0.812	0.576
Ours	**0.931**	**0.916**	**0.893**	**0.845**	**0.584**
			Neuroblastoma dataset		
Cellpose	0.657	0.612	0.574	**0.516**	**0.364**
Ours	**0.850**	**0.802**	**0.685**	0.476	0.190

Table 5: Results for the average precision(AP) for several intersection over union (IoU) thresholds for the U2OS nucli and Neuroblastoma datasets.

image where cells are not yet predicted. It allows to make multiple predictions to avoid missing cells. Third, the feedback error allows prediction on itself, ignoring previous predictions, thereby facilitating new ones.

Although Kaizen was not designed to perform segmentation, an evaluation against instance segmentation algorithms shows promising results. For example, surpassing stardist, a commonly used method, in average average precision. Overall, we propose an alternative method that presents several advantages over current methodologies.

4.6.1. Future work

A lot of work remains that requires further exploration. Instead of a VQ-VAE, other architectures can be used, like transformers or diffusion models [Cro+23]. Diffusion models can be an interesting approach since the different individual cells can be generated over several small steps, allowing each cell generation to take into account the surrounding cells.

Furthermore, although Kaizen relies on one generative model to make predictions, the core algorithm of Kaizen allows several models at the same time. It is possible to select the most fitting predictions proposed by a disparate set of models. Another interesting approach to generate a more diverse set of proposals might be to use a more sophisticated evolutionary algorithm.

5. CONCLUSION

In this thesis, we presented a set of contributions at the boundary of deep learning and microscopy image analysis, addressing challenges in biological image processing. Each chapter focused on a distinct but complementary problem and a distinct but related methodological approach.

In the first chapter, we explored the potential of deep learning to work directly with brightfield microscopy images. Traditionally, fluorescence staining has been considered indispensable for highlighting structures of interest, yet this process can be time-consuming and, in some cases, biologically disruptive. By demonstrating that neural networks can extract meaningful representations directly from brightfield data, we showed that staining-free imaging analysis is a viable alternative, offering significant savings in experimental preparation while preserving the integrity of living samples.

The second chapter introduced Stagetool, a neural network-based system designed to classify developmental stages in tubules and sperm cells. This work illustrates how supervised deep learning can be harnessed to capture subtle morphological and structural cues, enabling precise stage classification in complex biological processes. The application to reproductive biology is particularly relevant in the study of infertility, where quantitative, automated tools are urgently needed to complement expert assessments. More broadly, Stagetool exemplifies how deep learning can address domain-specific problems in biomedicine.

In the third chapter, we proposed a novel approach for object-level decomposition of microscopy images using a Vector Quantized Variational Autoencoder (VQ-VAE). Unlike classical segmentation methods, which often struggle in crowded images, this generative approach enables the image to be represented as a structured composition of individual objects. Such a representation not only supports tasks such as segmentation but might also contribute to the development of interpretable latent spaces that may facilitate new avenues of biological research. This work positions generative modeling as an alternative paradigm in microscopy image analysis, complementing other approaches and opening the door to expanded forms of computational analysis.

Looking forward, several directions for future research can be identified:

Multi-Modal Integration. Microscopy images represent only one layer of biological information. Combining imaging data with spatial transcriptomics, proteomics, or clinical metadata will be critical for building holistic models of cellular and tissue function. Deep learning methods capable of multi-modal fusion will play a central role in this integration.

Interpretability and Trustworthiness. As deep learning systems become increasingly embedded in biomedical workflows, it is essential to develop models whose decisions are transparent and interpretable. Methods such as error awareness, explainable AI frameworks, and biologically grounded latent representations will be central to ensuring that computational outputs can earn the trust of biologists and clinicians.

To conclude, this thesis has presented a series of contributions that advance deep learning for microscopy image analysis. By addressing tasks as diverse as staining-free imaging, biological stage classification, and generative objects for image decomposition, we have demonstrated the capability of deep learning in extracting knowledge from biological images. While challenges remain, the work presented here contributes to the scientific research on how artificial intelligence can reshape the study of cellular structures and biological processes.

BIBLIOGRAPHY

[Ble+96] Julie A Blendy et al. "Severe impairment of permatogenesis in mice lacking the CREM gene". In: *Nature* 380.6570 (1996), pp. 162–165.

[BW01] R Behr and GF Weinbauer. "cAMP response element modulator (CREM): an essential factor for spermatogenesis in primates?" In: *International Journal of Andrology* 24.3 (2001), pp. 126–135.

[Can09] John Canny. "A computational approach to edge detection". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (2009), pp. 679–698.

[Cha+17] Ai-Leen Chan et al. "Germline stem cell activity is sustained by SALL4-dependent silencing of distinct tumor suppressor genes". In: *Stem Cell Reports* 9.3 (2017), pp. 956–971.

[Chr+18] Eric M Christiansen et al. "In silico labeling: predicting fluorescent labels in unlabeled images". In: *Cell* 173.3 (2018), pp. 792–803.

[Cro+23] Florinel-Alin Croitoru et al. "Diffusion models in vision: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[Da +96] Sara Morais Da Silva et al. "Sox9 expression during gonadal development implies a conserved role for the gene in testis differentiation in mammals and birds". In: *Nature genetics* 14.1 (1996), pp. 62–68.

[Dek19] Marharyta Dekret. *SpiralNet: Two-stage recursive-CNN for microscopy image segmentation*. Master's thesis. Available at `https://dspace.ut.ee/server/api/core/bitstreams/13edf2fa-3b61-400e-89f4-9a203a8c5bb5/content`. June 2019.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of machine learning research* 12.7 (2011).

[DHS15] Jifeng Dai, Kaiming He, and Jian Sun. "Convolutional feature masking for joint object and stuff segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3992–4000.

[DL02] Wei Deng and Haifan Lin. "Miwi, a murine homolog of piwi, encodes a cytoplasmic protein essential for spermatogenesis". In: *Developmental cell* 2.6 (2002), pp. 819–830.

[Dru12] Gregor PC Drummen. "Fluorescent probes and fluorescence (microscopy) techniques—illuminating biological and biomedical research". In: *Molecules* 17.12 (2012), pp. 14067–14090.

[Fai+19] Imrul Faisal et al. "Transcription factor USF1 is required for maintenance of germline stem cells in male mice". In: *Endocrinology* 160.5 (2019), pp. 1119–1136.

[Fis+21] Dmytro Fishman et al. “Practical segmentation of nuclei in brightfield cell images with neural networks trained on fluorescently labelled samples”. In: *Journal of Microscopy* 284.1 (2021), pp. 12–24.

[FM82] Kunihiko Fukushima and Sei Miyake. “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”. In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[Gir15] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[He+17] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[Kra91] Mark A Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE journal* 37.2 (1991), pp. 233–243.

[KW13] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings* (Dec. 2013). DOI: `10.48550/arxiv.1312.6114`. URL: `https://arxiv.org/abs/1312.6114v10`.

[LA20] Zhiyuan Li and Sanjeev Arora. “AN EXPONENTIAL LEARNING RATE SCHEDULE FOR DEEP LEARNING”. In: *8th International Conference on Learning Representations, ICLR 2020*. 2020.

[Lab+23] Aqeel Labash et al. “Emergence of adaptive circadian rhythms in deep reinforcement learning”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 18153–18170.

[LC05] Jeff W Lichtman and José-Angel Conchello. “Fluorescence microscopy”. In: *Nature methods* 2.12 (2005), pp. 910–919.

[LeC+98] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[Lev+17] Hagai Levine et al. “Temporal trends in sperm count: a systematic review and meta-regression analysis”. In: *Human reproduction update* 23.6 (2017), pp. 646–659.

[Lev+22] Hagai Levine et al. “Temporal trends in sperm count: a systematic review and meta-regression analysis of samples collected globally in the 20th and 21st centuries”. In: *Human reproduction update* (2022).

[Liu+21] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[LSC12] Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. "Annotated high-throughput microscopy image sets for validation." In: *Nature methods* 9.7 (2012), pp. 637–637.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[Mäk+20] Juho-Antti Mäkelä et al. "Transillumination-assisted dissection of specific stages of the mouse seminiferous epithelial cycle for downstream immunostaining analyses". In: *JoVE (Journal of Visualized Experiments)* 164 (2020), e61800.

[Mat+22] Tambet Matiisen et al. "Do Deep Reinforcement Learning Agents Model Intentions?" In: *Stats* 6.1 (2022), pp. 50–66.

[MD24] Daniel Majoral and Marharyta Domnich. "Kaizen: Decomposing cellular images with VQ-VAE". In: *bioRxiv* (2024), pp. 2024–10.

[Mei+23] Oliver Meikar et al. "STAGETOOL, a novel automated approach for mouse testis histological analysis". In: *Endocrinology* 164.2 (2023), bqac202.

[MRD12] John L Mokili, Forest Rohwer, and Bas E Dutilh. "Metagenomics and future perspectives in virus discovery". In: *Current opinion in virology* 2.1 (2012), pp. 63–77.

[Muc+13] Barbara Muciaccia et al. "Novel stage classification of human spermatogenesis based on acrosome development". In: *Biology of reproduction* 89.3 (2013), pp. 60–1.

[MZV20] Daniel Majoral, Ajmal Zemmar, and Raul Vicente. "A model for time interval learning in the Purkinje cell". In: *PLoS computational biology* 16.2 (2020), e1007601.

[Nan+96] François Nantel et al. "Spermiogenesis deficiency and germ-cell apoptosis in CREM-mutant mice". In: *Nature* 380.6570 (1996), pp. 159–162.

[Oak56] Eugene F Oakberg. "Duration of spermatogenesis in the mouse and timing of stages of the cycle of the seminiferous epithelium". In: *American Journal of Anatomy* 99.3 (1956), pp. 507–516.

[Ots+75] Nobuyuki Otsu et al. "A threshold selection method from gray-level histograms". In: *Automatica* 11.285-296 (1975), pp. 23–27.

[Par+16] Miree Park et al. "SOHLH2 is essential for synaptonemal complex formation during spermatogenesis in early postnatal mouse testes". In: *Scientific reports* 6.1 (2016), pp. 1–12.

[PND20] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. "A survey on performance metrics for object-detection algorithms". In: *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE. 2020, pp. 237–242.

[QK20] Xin Qian and Diego Klabjan. "The impact of the mini-batch size on the variance of gradients in stochastic gradient descent". In: *arXiv preprint arXiv:2004.13146* (2020).

[Ren+15] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[RVV19] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. "Generating diverse high-fidelity images with vq-vae-2". In: *Advances in neural information processing systems* 32 (2019).

[Sch+18] Uwe Schmidt et al. "Cell detection with star-convex polygons". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 265–273.

[SF+68] Irwin Sobel, Gary Feldman, et al. "A 3x3 isotropic gradient operator for image processing". In: *a talk at the Stanford Artificial Project in* 1968 (1968), pp. 271–272.

[SFM16] Ron Sender, Shai Fuchs, and Ron Milo. "Revised estimates for the number of human and bacteria cells in the body". In: *PLoS biology* 14.8 (2016), e1002533.

[Sir+11] Anu Sironen et al. "Loss of SPEF2 function in mice results in spermatogenesis defects and primary ciliary dyskinesia". In: *Biology of reproduction* 85.4 (2011), pp. 690–701.

[Ska+16] Niels E Skakkebaek et al. "Male reproductive disorders and fertility trends: influences of environment and genetic susceptibility". In: *Physiological reviews* 96.1 (2016), pp. 55–97.

[Str+21] Carsen Stringer et al. "Cellpose: a generalist algorithm for cellular segmentation". In: *Nature methods* 18.1 (2021), pp. 100–106.

[Sut+13] Ilya Sutskever et al. "On the importance of initialization and momentum in deep learning". In: *International conference on machine learning*. pmlr. 2013, pp. 1139–1147.

[TRR18] Frank Tüttelmann, Christian Ruckert, and Albrecht Röpke. "Disorders of spermatogenesis". In: *medizinische genetik* 30.1 (2018), pp. 12–20.

[Van+16] David A Van Valen et al. "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments". In: *PLoS computational biology* 12.11 (2016), e1005177.

[Vas+17] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[VJT17] Helena E Virtanen, Niels Jørgensen, and Jorma Toppari. “Semen quality in the 21st century”. In: *Nature Reviews Urology* 14.2 (2017), pp. 120–130.

[VV+17] Aaron Van Den Oord, Oriol Vinyals, et al. “Neural discrete representation learning”. In: *Advances in neural information processing systems* 30 (2017).

[Wan+24] Shuai Wang et al. “MSR-UNet: enhancing multi-scale and long-range dependencies in medical image segmentation”. In: *PeerJ Computer Science* 10 (2024), e2563.

[Xu+19] Jun Xu et al. “Histopathological image analysis on mouse testes for automated staging of mouse seminiferous tubule”. In: *European Congress on Digital Pathology*. Springer. 2019, pp. 117–124.

[Xu+21] Jun Xu et al. “Computerized spermatogenesis staging (CSS) of mouse testis sections via quantitative histomorphological analysis”. In: *Medical image analysis* 70 (2021), p. 101835.

[Yu+19] Weimiao Yu et al. *CCDB: 6843, mus musculus, Neuroblastoma. CIL. Dataset*. `https://doi.org/doi:10.7295/W9CCDB6843`. Cell Image Library, 2019.

[Yua+00] Li Yuan et al. “The murine SCP3 gene is required for synaptonemal complex assembly, chromosome synapsis, and male fertility”. In: *Molecular cell* 5.1 (2000), pp. 73–83.

6. ACKNOWLEDGEMENTS

I would like to thank the past generations that worked to make the world a better place. On a smaller scale special thanks to all former members of the neuroscience lab, from whom I learned the obscure secrets of deep learning alchemy.

I would like to thank all my colleagues at the Computer Science Institute of the University of Tartu during my PhD who made this challenge lighter. Special thanks to Aqeel Labash for all the time shared during our doctoral life.

I am deeply indebted to my supervisors Leopold Parts and Raul Vicente for dealing with me for a long time. Last but not least, I am grateful to my family for their amazing support during all this time.

SISUKOKKUVÕTE

Sügavad närvivõrgud mikroskoopia piltide jaoks

Käesolevas väitekirjas käsitletakse süvaõppe meetodeid mikroskoopiapiltide analüüsimiseks. Töös tutvustatakse kolme erinevat teadusartiklit:

Esimene artikkel [Fis+21] kannab pealkirja: "Praktiline tuumade segmenteerimine heledate rakkude piltidel fluorestseeruvate märgistusega proovide põhjal treenitud närvivõrkude abil". Selles töös uurime, kuidas süvaõppe meetoditega saab tuvastada tuumasid heledavälja piltidel. Heledusvälja pilte on lihtsam saada kui fluorestseeruvaid pilte, kuid nende töötlemine traditsiooniliste meetoditega ja isegi inimeste jaoks on väga keeruline. Võrreldakse kolme erinevat meetodit Brightfield-piltide segmenteerimiseks: Deep cell, Mask R-CNN ja U-Net. Kuigi tuumasid saab segmenteerida ka teiste mudelitega, on U-Neti tulemused paremad. Lisaks sellele vajab U-Net vajalike piltide arvu hindamisel ainult 32 pilti, et saavutada tulemuslikkus, mis jääb 6% piires üldisest optimaalsest. Heledate väljadega mikroskoopias on võimalik teha mitu pildistamist erinevate fokaaltasanditega. Erinevate fokaaltasanditega katsetamisel leiavad autorid, et keskmised tasandid on paremad ja et kaks tasandit parandasid tulemuslikkust võrreldes ühe tasandiga, samas kui täiendavate tasandite teabe kaasamine andis kahanevat tulu.

Teine artikkel[Mei+23] on: "STAGETOOL, uudne automatiseeritud lähenemisviis hiire munandite histoloogiliseks analüüsiks". Meeste spermatosoidide arv on aastatel 1973-2018 vähenenud umbes 50-60% [Lev+17; Lev+22]. Meeste viljatuse diagnoosimiseks on hädasti vaja paremaid meetodeid. Selle probleemi lahendamiseks töötati välja uus meetod nimega Stagetool. Stagetool klassifitseerib DAPI-ga värvitud munandite ristlõikudel nii tubulaarsed staadiumid kui ka üksikud rakud. Tubulite klassifitseerimisel tunneb Stagetool ära viis staadiumi (I-V, VI-VIII, X, X-XI ja XII) ning kogu testise ristlõike puhul saavutab Stagetool 95,6% täpsuse. Rakkude klassifitseerimisel suudab Stagetool samuti tuvastada üheksa rakutüüpi. Üheksast klassist keskmistatuna on positiivne täpsus 0,9607. Stagetool töötab ka spermatogeensete defektidega KO-hiire mudelite keeruliste piltide puhul. Lõpuks saab Stagetooli rakendada antigeenide ekspressiooni automatiseeritud profileerimiseks.

Viimane artikkel [MD24] kannab pealkirja: „Kaizen: VQ-VAE". Artiklis pakutakse välja uus meetod arvutinägemise jaoks, Kaizen. Kaizen koosneb generatiivsest mudelist (VQ-VAE). Geneerivat mudelit treenitakse üksikute rakkude kodeerimiseks. Järeldamise ajal rakendatakse generatiivset mudelit järjestikku, et genereerida üksikute rakkude ennustused pildil. Prognoosid liidetakse kokku, et moodustada üldine sisemine kujutis. Uued ennustused võetakse vastu ainult siis, kui need parandavad välise sisendi ja sisemise kujutise sarnasust. Kaizen väldib objektide üle vaatamist, peatudes ainult siis, kui sisemine kujutis vastab välisele kujutisele. Seega lagundab Kaizen, selle asemel et segmenteerida või kasutada piiritletud ruutu, kujutise üksikuteks objektideks. Rakkude tuumade fluorestseeruvate mikroskoopiakujutiste andmekogumi puhul saavutab Kaizen kõikide künniste puhul parema täpsuse kui Stardist (üldkasutatav segmenteerimismeetod). Järgnevalt katsetatakse Kaizenit neuroblastoomi fluorestseeruvate mikroskoopiakujutiste andmestikul. Selles andmestikus ennustatakse kogu rakk, sealhulgas tsütoplasma. Võrreldes Cellpose'iga (tavaliselt kasutatav segmenteerimismeetod) saavutab Kaizen parema keskmise täpsuse mitme iou-künnise puhul (0.5 0.6 0.7).

Need kolm artiklit näitavad, kuidas tehisintellekt võib parandada mikroskoopiakujutiste analüüsi. Loodetavasti on tehisintellekt lähiaastatel eesrindlik läbimurdeid rakustruktuuride ja haiguste mehhanismide mõistmisel, et parandada inimeste olukorda.

7. CURRICULUM VITAE

Personal data

Name:	Daniel Majoral Lopez
Birth:	25/09/1981
Nationality:	Spain
Contact:	danielmajoral@gmail.com

Education

2017-2026	University of Tartu, Institut of computer science, Doctoral studies.
2015–2016	University of the Balearic Islands, Master in physics of complex systems.
2007–2008	Autonomous University of Barcelona, Master in mathematics for financial instruments.
1999–2005	Autonomous University of Barcelona, Physics degree.

Employment

2023-	Data scientist, Aleasoft
2016–2017	Junior researcher, University of Tartu
2012–2013	Software Developer, Generali
2008–2012	Middle Office specialist, La Caixa
2005–2007	Software Developer, T-systems

Scientific work

Main fields of interest:

- Deep learning
- Computational biology
- Neuroscience

ELULOOKIRJELDUS

Isikuandmed

Nimi:	Daniel Majoral Lopez
Sünnikuupäev:	25/09/1981
Kodakondsus:	Hispaania
Kontakt:	danielmajoral@gmail.com

Haridus

2017–2026	Tartu Ülikool, arvutiteaduse instituut, doktoriõpe.
2015–2016	Baleaari saarte ülikool, Komplekssete süsteemide füüsika magister.
2007–2008	Barcelona autonoomne ülikool, Finantsinstrumendi matemaatika magister.
1999–2005	Barcelona autonoomne ülikool,Füüsika kraad.

Teenistuskäik

2023-	Andmeteadlane, Aleasoft
2016–2017	Nooremteadur, Tartu Ülikool
2012–2013	Tarkvaraarendaja, Generali
2008–2012	Keskbüroo spetsialist, La Caixa
2005–2007	Tarkvaraarendaja, T-systems

Teadustegevus

Peamised uurimisvaldkonnad:

- Sügav õppimine
- Arvutusbioloogia
- Neuroteadus

DISSERTATIONES INFORMATICAE PREVIOUSLY PUBLISHED IN DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Šor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multi-party computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh**. Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas**. Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi**. Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich**. Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka**. Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinemaa**. Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto**. Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.
24. **Arnis Parsovs.** Estonian Electronic Identity Card and its Security Challenges. Tartu 2021, 214 p.
25. **Kaido Lepik.** Inferring causality between transcriptome and complex traits. Tartu 2021, 224 p.
26. **Tauno Palts.** A Model for Assessing Computational Thinking Skills. Tartu 2021, 134 p.
27. **Liis Kolberg.** Developing and applying bioinformatics tools for gene expression data interpretation. Tartu 2021, 195 p.
28. **Dmytro Fishman.** Developing a data analysis pipeline for automated protein profiling in immunology. Tartu 2021, 155 p.
29. **Ivo Kubjas.** Algebraic Approaches to Problems Arising in Decentralized Systems. Tartu 2021, 120 p.
30. **Hina Anwar.** Towards Greener Software Engineering Using Software Analytics. Tartu 2021, 186 p.
31. **Veronika Plotnikova.** FIN-DM: A Data Mining Process for the Financial Services. Tartu 2021, 197 p.
32. **Manuel Camargo.** Automated Discovery of Business Process Simulation Models From Event Logs: A Hybrid Process Mining and Deep Learning Approach. Tartu 2021, 130 p.
33. **Volodymyr Leno.** Robotic Process Mining: Accelerating the Adoption of Robotic Process Automation. Tartu 2021, 119 p.
34. **Kristjan Krips.** Privacy and Coercion-Resistance in Voting. Tartu 2022, 173 p.
35. **Elizaveta Yankovskaya.** Quality Estimation through Attention. Tartu 2022, 115 p.
36. **Mubashar Iqbal.** Reference Framework for Managing Security Risks Using Blockchain. Tartu 2022, 203 p.
37. **Jakob Mass.** Process Management for Internet of Mobile Things. Tartu 2022, 151 p.
38. **Gamal Elkoumy.** Privacy-Enhancing Technologies for Business Process Mining. Tartu 2022, 135 p.
39. **Lidia Feklistova.** Learners of an Introductory Programming MOOC: Background Variables, Engagement Patterns and Performance. Tartu 2022, 151 p.
40. **Mohamed Ragab.** Bench-Ranking: A Prescriptive Analysis Approach for Large Knowledge Graphs Query Workloads. Tartu 2022, 158 p.
41. **Mohammad Anagreh.** Privacy-Preserving Parallel Computations for Graph Problems. Tartu 2023, 181 p.
42. **Rahul Goel.** Mining Social Well-being Using Mobile Data. Tartu 2023, 104 p.

43. **Anti Ingel.** Algorithms using information theory: classification in brain-computer interfaces and characterising reinforcement-learning agents. Tartu 2023, 142 p.
44. **Shakshi Sharma.** Fighting Misinformation in the Digital Age: A Comprehensive Strategy for Characterizing, Identifying, and Mitigating Misinformation on Online Social Media Platforms. Tartu 2023, 158 p.
45. **Kristiina Rahkema.** Quality Analysis of iOS Applications with Focus on Maintainability and Security Aspects. Tartu 2023, 182 p.
46. **Ivan Slobozhan.** Studying Online Social Media Engagement in CIS Countries during Protests, Mass Demonstrations and War. Tartu 2023, 81 p.
47. **Nurlan Kerimov.** Building a catalogue of molecular quantitative trait loci to interpret complex trait associations. Tartu 2023, 248 p.
48. **Pavlo Tertychnyi.** Machine Learning Methods for Anti-Money Laundering Monitoring. Tartu 2023, 117 p.
49. **Abasi-amefon Obot Affia.** A Framework and Teaching Approach for IoT Security Risk Management. Tartu 2023, 180 p.
50. **Raimond-Hendrik Tunnel.** Video Game Design and Development Bachelor's Curriculum for Estonia. Tartu 2024, 137 p.
51. **Ahto Salumets.** Bioinformatics analysis of various aspects in immunology. Tartu 2024, 198 p.
52. **Mohammed Abdulhameed Shaif Ali.** Deep Learning Methods for Cell Microscopy Image Analysis. Tartu 2024, 143 p.
53. **Pille Pullonen-Raudvere**. Foundations of Efficient and Secure Algorithm Development for Secure Multiparty Computation. Tartu 2024, 265 p.
54. **Marili Rõõm.** Multiple approaches to learners' success and factors affecting it in computer programming MOOCs. Tartu 2024, 170 p.
55. **Shivananda Rangappa Poojara.** Design and Orchestration of Scalable, Event-Driven Serverless Data Pipelines for Internet of Things (IoT) Applications. Tartu 2024, 172 p.
56. **Hassan Abdulgaleel Hassan Salim Eldeeb.** Empowering Machine Learning Pipelines with Automated Feature Engineering. Tartu 2024, 121 p.
57. **Muhammad Uzair.** Soft decision making for agri-food 4.0. Tartu 2024, 158 p.
58. **Kirill Milintsevich.** Estimation of Depression Level from Text: Symptom-Based Approach, External Knowledge, Dataset Validity. Tartu 2024, 130 p.
59. **Maksym Del.** Multilingual and Multi-Domain Representational Patterns Across Trpansformer-Based Models. Tartu 2024, 131 p.
60. **Kristo Raun.** Adaptive Out-of-order Handling in Streaming Conformance Checking. Tartu 2024, 118 p.
61. **Toivo Vajakas.** Towards integration of mobile network data into analyzing human mobility. Tartu 2024, 103 p.
62. **Katsiaryna Lashkevich.** Data-Driven Analysis and Optimization of Waiting Times in Business Processes. Tartu 2024, 169 p.
63. **Alejandra Duque-Torres.** Classifying, Constraining and Ranking Metamorphic Relations. Tartu 2025, 159 p.

64. **Mariia Bakhtina.** A Method for Information Security and Privacy Management in Smart Solutions. Tartu 2025, 199 p.
65. **Andre Tättar.** Multilingual Machine Translation for Under-Resourced Languages. Tartu 2025, 170 p.
66. **Mahmoud Shoush.** Prescriptive Process Monitoring Under Uncertainty and Resource Constraints. Tartu 2025, 178 p.
67. **Alireza Akhavi Zadegan.** A Multimodal approach for refining Mapping and Localization by Integrating Generative AI and Pedestrian-Centric Data. Tartu 2025, 147 p.
68. **Eerik Muuli.** Automating the assessment and feedback processes in IT teaching – improving creation and maintenance from the teaching staff perspective. Tartu 2025, 196 p.
69. **Kateryna Kubrak.** Towards User-Centered Prescriptive Process Monitoring Systems. Tartu 2025, 151 p.
70. **Zhigang Yin.** Computing and Sensing in a Smart Ring. Tartu 2025, 251 p.
71. **Abdul-Rasheed Olatunji Ottun.** Practical Trustworthy Artificial Intelligence with Human Oversight. Tartu 2025, 239 p.
72. **Sander Mikelsaar.** Analysis and Optimization of Iteratively Decodable Codes. Tartu 2025, 146 p.
73. **Marharyta Domnich.** Advancing Human-Centric Counterfactual Explanations in Explainable AI. Tartu 2025, 210 p.
74. **Viacheslav Komisarenko.** Aligning Training Loss to Evaluation Metrics in Deep Learning. Tartu 2026, 165 p.
75. **Heidi Taveter.** Using Programming-Process Data of Introductory Programming Courses: Finding Solver Types, Giving Feedback, and Detecting Plagiarism. Tartu 2026, 184 p.