Learning-based prediction, representation, and multimodal registration for bioimage processing

NICOLAS PIELAWSKI
Abstract


Microscopy and imaging are essential to understanding and exploring biology. Modern staining and imaging techniques generate large amounts of data resulting in the need for automated analysis approaches. Many earlier approaches relied on handcrafted feature extractors, while today's deep-learning-based methods open up new ways to analyze data automatically.

Deep learning has become popular in bioimage processing as it can extract high-level features describing image content (Paper III). The work in this thesis explores various aspects and limitations of machine learning and deep learning with applications in biology. Learning-based methods have generalization issues on out-of-distribution data points, and methods such as uncertainty estimation (Paper II) and visual quality control (Paper V) can provide ways to mitigate those issues. Furthermore, deep learning methods often require large amounts of data during training. Here the focus is on optimizing deep learning methods to meet current computational capabilities and handle the increasing volume and size of data (Paper I). Model uncertainty and data augmentation techniques are also explored (Papers II and III).

This thesis is split into chapters describing the main components of cell biology, microscopy imaging, and the mathematical and machine-learning theories to give readers an introduction to biomedical image processing. The main contributions of this thesis are deep-learning methods for reconstructing patch-based segmentation (Paper I) and pixel regression of traction force images (Paper II), followed by methods for aligning images from different sensors in a common coordinate system (named multimodal image registration) using representation learning (Paper III) and Bayesian optimization (Paper IV). Finally, the thesis introduces TissUUmaps 3, a tool for visualizing multiplexed spatial transcriptomics data (Paper V). These contributions provide methods and tools detailing how to apply mathematical frameworks and machine-learning theory to biology, giving us concrete tools to improve our understanding of complex biological processes.

Keywords: Deep learning, Multimodal image registration, bayesian optimization, Bioimage processing

Nicolas Pielawski, Department of Information Technology, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.

© Nicolas Pielawski 2023

ISSN 1651-6214
ISBN 978-91-513-1725-0
URN urn:nbn:se:uu:diva-497143 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-497143)
To Li and Émile
List of papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.


Reprints were made with permission from the publishers.
List of related work

In addition to the papers included in this thesis, the author has also contributed to the following publications/published datasets:

Summary of Contributions

The roman numerals correspond to the numbers in the list of papers.

I N. Pielawski and C. Wählby are the main contributors to the design of the method, and analysis and interpretation of the results. N. Pielawski implemented and conducted the experiments. All authors participated in the writing of the manuscript.

II N. Pielawski is the main contributor to the method’s design and implementation. J. Hu collected the dataset and computed the traction forces. N. Pielawski and C. Wählby are the main contributors to the analysis of the results. All authors participated in the interpretation of the results and the writing of the manuscript.

III N. Pielawski and E. Wetzer equally contributed to the method’s conception, design, analysis, and interpretation of the results. N. Pielawski implemented the code for representation learning and generating the figures. E. Wetzer and J. Öfverstedt implemented the code and prepared the dataset for testing and analysis of image registration. E. Wetzer and J. Lu implemented and compared alternative multimodal registration techniques against the proposed method. All authors participated in the writing of the manuscript.

IV N. Pielawski is the main contributor to the design and implementation of the bayesian optimization approach to image registration. N. Pielawski and E. Wetzer contributed to the theoretical framework of the method. J. Öfverstedt and N. Pielawski implemented the code for the evaluation and analysis of the experiments. N. Pielawski and E. Wetzer wrote the paper with input from the co-authors.

V N. Pielawski, A. Andersson, and C. Wählby are the main contributors to the design and orchestration of the paper. C. Avenel, L. Solorzano, and F. Nysjö implemented the central, critical components of the code. C. Avenel, N. Pielawski, A. Andersson, E. Chelebian, and A. Behanova implemented TissUUmaps plugins. C. Avenel and F. Nysjö benchmarked TissUUmaps; N. Pielawski analyzed and generated the figures of the results. All authors contributed to the writing of the manuscript.
Contents

1 Introduction .............................................................................................................. 11

2 Biology and quantitative microscopy ..................................................................... 13
  2.1 Cell biology ........................................................................................................... 13
    2.1.1 What is a cell? ................................................................................................. 13
    2.1.2 Transcription ................................................................................................. 14
    2.1.3 Cell motility ..................................................................................................... 14
  2.2 Imaging modalities ............................................................................................... 15
    2.2.1 Brightfield microscopy .................................................................................. 15
    2.2.2 Fluorescence microscopy .............................................................................. 16
    2.2.3 Phase contrast microscopy .......................................................................... 17
    2.2.4 Electron microscopy ...................................................................................... 17
    2.2.5 Second-harmonic imaging microscopy ....................................................... 17
    2.2.6 Traction force microscopy ........................................................................... 19
  2.3 Biomedical image analysis .................................................................................. 19

3 Mathematics and computer vision theory ................................................................ 22
  3.1 Hypercomplex number systems ......................................................................... 22
    3.1.1 Complex numbers ......................................................................................... 22
    3.1.2 Quaternions .................................................................................................... 25
    3.1.3 Dual numbers ................................................................................................ 26
  3.2 Fourier transform ............................................................................................... 28
    3.2.1 Discrete Fourier transform ......................................................................... 28
    3.2.2 Fast-Fourier transform ................................................................................ 31
    3.2.3 Fourier transform of images ....................................................................... 32
  3.3 Probabilities and statistics .................................................................................. 32
    3.3.1 Lognormal distribution of microscopy images ........................................... 37

4 Machine learning in digital image processing ......................................................... 39
  4.1 Inductive biases and model architectures .......................................................... 40
    4.1.1 Model fitting ................................................................................................... 41
    4.1.2 Loss functions ................................................................................................ 42
  4.2 Kernel methods and bayesian optimization ......................................................... 45
    4.2.1 Kernel methods .............................................................................................. 46
    4.2.2 Common kernels ............................................................................................ 47
    4.2.3 Gaussian processes ....................................................................................... 51
1. Introduction

Artificial neural networks are a collection of algorithms inspired by the human brain. Like the human brain, these networks learn to recognize and interpret different patterns automatically after training. More complex networks with several layers of neurons are called deep neural networks and started to become computationally feasible and possible to use at scale around 2012, leading to an increasing number of published papers on artificial intelligence. Furthermore, these methods have been shown to perform significantly better than many classical methods in prestigious classification challenges organized in connection with different image analysis conferences [1, 2].

When I started my Ph.D. journey at the beginning of 2018, many research areas were already permeated by these new deep-learning methods, and an interest in using them in biomedical research had emerged. Since then, the methods have kept evolving, and a proper theory has started to crystallize. This thesis is my attempt at providing an introduction to the background of this interdisciplinary field. More specifically, the thesis aims at giving a summary of the tools needed for working at the crossing between biology, computer science, and mathematics. Furthermore, I hope to provide an interesting overview of the interdisciplinary research I have been conducting over the past five years. During this time, I have learned that intuition is an essential trait of a researcher, and throughout this work, I have tried to communicate my own. In addition, I have attempted to contribute with a new point of view, sometimes by bringing different ideas under the same umbrella to provide a different perspective from the established reading material out there, hoping to provide the reader with new, engaging, and fun insights.

This thesis contains five publications: Papers I and II focus on deep-learning methods applied to image classification and regression. Paper III and IV contain techniques for rigid registration and alignment of images from a common coordinate system of images from different sensors. Finally, paper V describes a tool – TissUUmaps 3 – that we have developed to visualize large amounts of data and describe the spatial locations of different active genes in a tissue. The visualization is essential to formulate any hypotheses necessary for continued quantitative analysis.

As for the organization of the thesis, it begins with an introduction to biology and the many types of microscopes used for imaging cells and tissues in chapter 2. Then follows an introduction to mathematics in chapter 3, which is necessary to understand the theory behind the many machine-learning methods described in chapter 4. These introductory chapters provide the background
needed to motivate the choices of methods and help explain the description of the papers presented in chapter 5. Finally, the conclusion and future research are summarized in chapter 6.
2. Biology and quantitative microscopy

Biology is the study of the building blocks of life. The field focuses on answering fundamental questions, helping us understand our origins, and fighting challenges and diseases humanity faces. The study of the infinitely small and the infinitely big\(^1\) are both paved with challenging questions: how do we see what is going on? How do we create and validate models that fit reality when we cannot directly observe what is happening with our eyes? Can we even find helpful models? There are no set answers, but we discovered many methods and technologies over time to uncover the underlying working of the atoms, cells, tissues, and organs in great detail and provide valuable insights. One such technological invention, perhaps the most important, is the microscope.

This chapter will describe how biological cells function and locomote – which is essential when trying to understand diseases such as cancer – and how different microscopes allow the imaging of various aspects of cells and tissues while having inherent strengths and weaknesses.

2.1 Cell biology

Cells are the essential structural elements of life forms, combining to make tissues, organs, and living organisms. They fulfill a vast range of functions, for instance, allowing for motility (muscle cells called myocytes), gathering energy from the environment (epithelial cells for breathing, intestinal epithelial cells), and thinking (neurons).

In this section, we briefly describe human cells, transcription, and migration, which will provide some ground to understand the biological components of the papers in this thesis.

2.1.1 What is a cell?

Animals (humans included) are made of eukariotic\(^2\) cells, which are compartmentalized structures held together with a membrane. Aside from the following brief description, we will not dive into the details of the constituent elements of cells. The cell cytoplasm – the interior of the cell – contains a well-structured soup rich in proteins and provides different functions, for instance,

\(^1\)Coincidentally, many challenges faced to explore biology and the universe are similar in technology, from microscopes to telescopes, and more recently, in computer vision, e.g., counting individual cells and stars.

\(^2\)From the Greek, \(e\upmu\) meaning "well, good" and \(karyon\) meaning "nut, kernel".
giving structure to the cell, allowing the cells to interact with their immediate environment, and the ability to replicate. The deoxyribonucleic acid (DNA) contains all the information needed for maintaining integrity and cell mitosis – reproducing by splitting a mother cell into two daughters. The DNA is protected by the nucleus, a cell compartment with its own membrane. The cell maintains its shape with the cytoskeleton, a network of proteins that provides mechanical resistance to deformation. It can also retract and expand, allowing cells to deform and migrate as further described in 2.1.3.

2.1.2 Transcription

Cells are full of proteins responsible for different functions, but they also decay rapidly as their half-lives may range from minutes to days [3]. Therefore, cells must constantly produce new proteins through protein synthesis to maintain their function. First, cells use their DNA – containing the blueprint or templates for all the needed proteins – to perform transcription, the process responsible for reading the DNA and producing orders for the cell to synthesize a specific protein. Next, enzymes open the strand of DNA and read the strands of nucleotides sequentially, producing messenger RNA (mRNA) simultaneously. mRNA is a short sequence of nitrogenous bases that escapes the nucleus and arrives in the cytoplasm facilitated by nuclear pore complexes [4]. Diverse processes will change the amount of mRNA in the cytoplasm, changing how much different genes are expressed. Then, the mRNA is used to synthesize the proteins in a process named translation, where a large molecular machine called a ribosome reads the mRNA in the cytoplasm and starts producing the protein. At this stage, the protein is a chain of amino acids that will start folding onto itself and finally be able to fulfill its function. Unfortunately, proteins can misfold, which is often harmless but is also believed to be responsible for certain diseases, such as Alzheimer’s and Parkinson’s disease [5].

Padlock probing [6] is a method where a synthetic RNA sequence detects a specific mRNA molecule, amplifies it, and reads the nucleotide sequence using repeated hybridization with fluorescent molecules. This allows for the visualization of the spatial transcriptomes of the cells, that is, understanding the presence of expressed genes at a given time along with their spatial locations. The resulting data is very rich, calling for interactive visualization tools as discussed in Paper V.

2.1.3 Cell motility

Cell motility allows cells to migrate throughout a living organism, a crucial component for development (e.g., of embryos), organ formation, immunological response, and wound healing [7]. Cells can move in one dimension, following strands, two dimensions, on the surface of tissues, or in three dimen-
sions in 3D matrices. The cells act on their cytoskeleton, using actin proteins to deform, stretch, change shape, and protrude. Through the surface of the cytoplasm lie cell-matrix adhesion complexes (CMACs) that can connect and attach to the extra-cellular matrix (ECM) – a three-dimensional network of collagen and proteins. The cells can thus create protrusions, adhere to the ECM and pull (acting on the cytoskeleton) to achieve motility. Understanding cell motility is critical to understanding certain diseases, notably cell migration in the case of metastatic cancers [8]. Subsection 2.2.6 discusses traction force microscopy [9], a way to quantify the forces a cell exerts using this type of motility, which is the focus of Paper II.

2.2 Imaging modalities

A microscopy experiment’s manual assessment is time-consuming and very subjective. Since the end of the 16th century, it became possible to use increasingly complex combinations of lenses and explore the invisible world of microscopy. The technology, as well as the methods, evolved a great deal – from drawing sketches to taking pictures and filming. Still, in recent years it has also become possible to process the results of the experiments fully automatically. These methods are not necessarily novel, most dating from the second half of the 20th century. Still, today’s computer capabilities allow us to use computer vision and machine learning techniques to interpret and evaluate experiments to push medical research forward faster than ever before.

2.2.1 Brightfield microscopy

Brightfield microscopy is the most straightforward technique to analyze cells and subcellular structures. Light is emitted from below the sample and is transmitted through it. As a result, the specimen restricts the light, making its shapes discernible from the background. The transparent background and dark shadowy samples gave the name to this technique: brightfield microscopy. Most biological samples are transparent, which makes this method polyvalent. Researchers also often stain a specimen to get a precise color differentiation between the different parts of the sample. However, this damages the cells, heavily deteriorating or even killing them.

Microscopes, channeling the information between the samples and researchers, have many important parameters to set to use them properly. The magnification represents the number of times an object will look enlarged with respect to the geometry of the microscope. Numerical Apertures (NA) are the ability of a lens to capture light and are written on the lens. Using a specific medium between the sample and the lens, such as water or oil, increases the NA and, thus, the resolution.
The resolution of an image determines the minimal distance between which two individual points can be distinguished. The higher the resolution, the better the details and, hopefully, the quality. However, high-resolution images often contain single cells instead of multiple ones, hindering the use of statistics. Nevertheless, the resolution is essential as it allows us to convert a pixel to a distance, thus making more precise visual measurements. The limit of the resolution can be calculated as follows:

\[
D = \frac{\lambda}{\text{NA}_{\text{condenser}} + \text{NA}_{\text{objective}}} = \frac{\lambda}{2 \text{NA}}
\]

with \( D \) the minimum distance at which two points can be resolved, \( \lambda \) the vacuum wavelength of the light, and \( \text{NA}_{\text{condenser}}, \text{NA}_{\text{objective}} \) the numerical apertures of the condenser and the objective lenses respectively.

Equation 2.1 helps us determine the resolution for a given light frequency and the NA of our microscope. A smaller light wavelength or a higher microscope NA increases the resolution.

The minimal resolution possible on a brightfield microscope is around 0.2\( \mu \)m, which is a physical limitation due to two constraints: the visible light is limited to wavelengths ranging from 360 to 700, and it is challenging to get numerical apertures greater than 1.40. Yet, it is possible to overcome those limitations using super-resolution microscopy methods, such as STORM (Stochastic Optical Reconstruction Microscopy) [10].

2.2.2 Fluorescence microscopy

Fluorescence microscopy is a technique where a specimen is modified to contain fluorescent molecules called fluorophores. Under specific lighting conditions, the specimen will fluoresce, i.e., re-emit light, and the studied parts will be easily distinguished over the dark background. Fluorescent proteins may be attached to specific antibodies that bind to structures in fixed (dead) cells or can be used to observe processes in living cells. A commonly used fluorescent protein is GFP (Green Fluorescent Protein), a natural protein isolated from the \textit{Aequorea Victoria} jellyfish in the 1960s [11].

It is possible to make a cell contain fluorescent proteins by either administering them through lipids containing mRNA of the GFP — the cell line will be called transient — or by modifying the genome of the cell to add the GFP sequence in the DNA, in which case the cell line will be called stable. However, GFP expression decays over time because the fluorescent protein is likely to cause cellular damage and cell death and may lead to a misinterpretation of an experiment’s results [12].

When the microscope produces a light beam directed toward the sample, the fluorophores will deteriorate and eventually break down. This process is named photobleaching and causes the intensity of the image to fade over time.
If the cell is stable, the DNA will be translated to mRNA, which leads to the synthesis of new fluorescent proteins, and the sample will fluoresce once again.

Finally, the STORM technique, invented in 2006, permits the creation of super-resolution images. The fluorophores are stochastically switched on and off, and the detected points can be resolved individually as they will be further apart from one another on average. This yields a higher resolution image that can detect individual proteins [10].

2.2.3 Phase contrast microscopy
This type of optical microscopy uses subtle differences in refractive indices arising from the sample’s cytoplasmic components. More, light is an electromagnetic wave and thus possesses constructive and destructive interference patterns, such that when two light beams are in phase, they reinforce each other or vanish when they are out of phase. Phase contrast microscopy allows seeing structures that would not be visible when using brightfield techniques while not requiring staining. This allowed studies of live cells since most staining methods are toxic and may cause the cells to die.

2.2.4 Electron microscopy
Electron microscopes use an electron beam to construct an image [13]. The electrons are directed with electromagnets, and sensors receive the electrons scattered by or transmitted to the specimen. Using this method, the resolution can reach around 0.1nm. However, despite its excellent resolution, electrons get scattered whenever they hit an atom, so the experiments must be performed in a vacuum to avoid hitting air molecules. In this case, the electrons can go directly from the source to the samples that are either polymerized or frozen [14].

2.2.5 Second-harmonic imaging microscopy
Second-Harmonic Generation (SHG) is a nonlinear optical process that exploits a property of light where a photon can scatter and produce two photons at twice the original frequency. This method helps quantify specific structures prone to generating SHG signals, helping to understand better diseases such as particular types of cancers and connective tissue disorders [15]. A higher-intensity light source is required compared to other methods but doesn’t require staining, doesn’t degrade the sample’s health much, and can be used on live tissues.

When a light beam propagates through a transparent medium, the medium becomes electrically polarized (note that we are not talking about the polarization of the light field here). This polarization can be decomposed as the
following Taylor series [16]:

$$P(t) = \chi^{(1)} E(t) + \chi^{(2)} E^2(t) + \chi^{(3)} E^3(t) + \ldots, \quad (2.2)$$

with $P$ the total polarization, $\chi^{(n)}$ the $n$th order nonlinear susceptibility of the medium (usually measured experimentally), and $E$ the electric field of incident light. The first term $\chi^{(1)} E(t)$ describes the normal absorption, reflection, and scattering of light, while the second term $\chi^{(2)} E^2(t)$ describes the SHG. The SHG is highly susceptible to noncentrosymmetric molecules, which biologically correspond to collagen, microtubules, and myosin.

$\chi^{(2)}$ can be further decomposed into

$$\chi^{(2)} = N_s \mathbb{E}[\beta], \quad (2.3)$$

with $N_s$ the molecular density of the medium, and $\mathbb{E}[\beta]$ the average orientation of the molecules of interest. Figure 2.1 shows a tissue – a tissue microarray (TMA) core – imaged by a brightfield microscope and second harmonic generation, where specific molecules present in the tissue are susceptible to generating an SHG signal captured by the microscope. A dataset of TMA cores was used in Paper III where both modalities (brightfield and SHG) are imaged in different microscopes and need to be registered, that is, transformed to be in the same frame of coordinates with matching structures.
2.2.6 Traction force microscopy

Traction Force Microscopy (TFM) is a recently developed method to study cell motility. Cells are laid on an elastic gel containing fluorescent beads (shown in Figure 2.2 b.). The gel deforms when the cells grip the medium, and the beads are displaced (Figure 2.2 a.). It becomes possible to reconstruct the forces the cells exert on the gel by matching the beads before and after the displacement with a mathematical model, e.g., splines, describing the deformation properties of the gel [18]. The forces typically form a dense vector field with a magnitude and orientation. In Paper II, we created a deep learning model to predict the magnitude of the traction forces from fluorescent microscope images of the sample.

TFM has several downsides, such as requiring skilled and experienced biologists; cells are known to behave differently on non-elastic media, so one must exercise caution when generalizing the results of an experiment; and it is difficult to assess the quality of the reconstruction algorithm because no ground truth force vector field is available. That is, there is no knowledge about the cell’s fundamental forces on the gel, making the calibration and validation of force reconstruction algorithms challenging.

2.3 Biomedical image analysis

The first microscopes were created almost a half millennium ago, achieving a magnification of up to 20 or 30 times the sample size [20]. The door to seeing things beyond the capabilities of our eyes was open. However, it took until...
the very end of the 20th century to use computers to record images permanently and move on from qualitatively judging the result of an experiment to quantitative assessment, which led to an upsurge of publications in microscopy on PubMed (Figure 2.3, top plot). Today, the ability of computers to deal with immense amounts of data changed the paradigm, enabling data science techniques and artificial intelligence to further enhance our ability to research complex questions. Figure 2.3 (bottom plot) shows the exponential growth, starting around 2016, of publications on PubMed related to microscopy and artificial intelligence combined. This recent rise in publications of a developing theory of deep learning in biology reflects improvements in architecture, training methodologies, and learning-based methods where significant components could be updated monthly. Currently, the number of AI and machine learning preprints on arXiv approximately doubles every 23 months [21].

Defining an image consistently across scientific fields can be challenging when crossing the many areas of microscopy, digital image processing, and mathematics. Mathematically, an image is an n-dimensional tensor (a multi-dimensional extension to matrices) where the dimensions can be spatial (width, height, and depth), temporal, span multiple channels, and focus levels. When answering biological questions, the image’s content is of interest. It can vary broadly, where the focus can be a single object (like a cell) and its structure, groups of objects, or textures (e.g., in a tissue). Finally, we may be interested
in the multiple modalities of a sample, where various sensors or microscopes image the same object – resulting in numerous views – each giving a different type of information about the sample and providing some facet of the answer to multiple complex research questions. Small objects can nowadays be trivially captured by microscopes along with thousands of channels, rendering the visualization of such datasets challenging. Paper V describes TissUUmaps, a visualization tool suitable for such tasks. Generally, there is no set answer as to which tool or algorithm technique should be used on a given dataset. Therefore, review articles become very relevant tools in finding a suitable method, as done in publication R1.
3. Mathematics and computer vision theory

Mathematics is the basic component of this thesis, where the theories provide a sound ground for developing, improving, and testing the methods of interest. Moreover, mathematics provides a way to understand and manipulate otherwise obscure theories. Papers I, II, III and IV heavily rely on mathematical frameworks, from theories stemming from statistics, linear algebra, real and complex analysis, signal processing, and machine learning.

This chapter focuses on hypercomplex number systems, from complex numbers to quaternions. The dual representation of numbers is discussed as a way to fit learning-based methods to a set of data points. The focus shifts from there onto signal processing, notably discussing the Fourier transform in detail, and ends with probability and statistics. All those topics are important and relevant to grasp the next chapter about machine learning and the papers in this thesis.

3.1 Hypercomplex number systems

Hypercomplex number systems are an extension of the real number system. They create new tools to manipulate quantities useful for geometric transformation, computing derivatives, etc. For example, in Paper IV, we used complex numbers to improve the speed of an algorithm that can register images and quaternions for rotating three-dimensional images. In addition, we used methods stemming from dual numbers to train neural networks through automatic differentiation in Papers II, III, and IV.

3.1.1 Complex numbers

Complex numbers are a number system [22, p. 1] that extends real numbers to the two-dimensional plane, with a real horizontal line and an imaginary vertical line. Complex numbers take the form $a + ib$ with $i^2 = -1$. Although a number squared being negative seems nonsensical initially, it becomes possible in two dimensions. This property was initially met with skepticism but is nowadays widely accepted in the mathematical community. An entertaining account of the invention of complex numbers and their usefulness has been covered by Veritasium [23].

Complex numbers can be equivalently separated into two distinct representations, which we will name point representation for $a + ib$, corresponding to a
two-dimensional point with coordinates \((a, b)\), and rotation representation for \(r \exp(i\theta)\), for a radius \(r\) and for an angle of \(\theta\) radians\(^1\). Point representation helps represent physical objects, such as two-dimensional points or images (a grid of points). On the other hand, rotation representation is used to quantify a rotation of the space when \(r = 1\). In mathematics, rotations are often dealt with through other means, such as a linear combination of sine and cosine functions or, more commonly, rotation matrices. However, the complex representation will become helpful when we extend complex numbers to rotations in three-dimensional spaces in the next section.

Figure 3.1 displays a heart curve [24] in the complex plane; that is, the \(x\) and \(y\) coordinates are mapped to a point representation as \(x + iy\). The heart is rotated with two different rotation representations of complex numbers: \(\exp(i\pi/2)\) and \(\exp(-i\pi/3)\) (rotations by 90° and −60° respectively), which demonstrates that complex numbers indeed act as rotation operators. Note that a rotation is performed counter-clockwise by convention.

Euler’s formula states that

\[
\exp(i\theta) = \cos(\theta) + i\sin(\theta).
\]

For instance, \(\exp(i\pi/2) = \cos(\pi/2) + i\sin(\pi/2) = i\). This means multiplying by \(i\) is equivalent to a 90° rotation, and \(i^2\) a rotation of 180°. If we imagine that 1 is a horizontal vector starting at the origin and pointing at 1 on the real line, multiplying it by \(i\) makes the vector point upwards (from 0 to 1 unit in the imaginary axis), and multiplying by \(i\) a second time makes the vector pointing leftwards (from 0 to −1, back again on the real line). This is a geometric intuition explaining why \(i^2 = 1\).

Another use for complex numbers is the representation of periodic functions, which will be helpful with the Fourier decomposition. The simplest periodic complex signal is a complex sinusoid, which can be represented with two variables: \(f\) for the frequency and \(r\) for the magnitude: \(r \exp(2\pi if t)\) with \(t\) the time. The frequency represents how many oscillations occur every second; the unit is called Hertz. The convention \(\omega = 2\pi f\) is often used instead, and \(\omega\) is named the angular frequency. This formula can be used to model physical systems such as a vibrating string or a rotating propeller over time. For example, Figure 3.2 displays the complex sinusoid \(\exp(2\pi it)\) – which has a magnitude of 1 and frequency of 1 Hertz – and both in 2D (left) and in 3D (right). In section 3.2, we will see how any signal can be represented as a sum of complex sinusoids. In this case, the complex sinusoids will be called frequency components.

---

\(^1\)For completeness, a radian is a unit describing angles and is in the range \([0, 2\pi]\), where \(2\pi\) represents 360°. Converting \(\theta\) degrees to radians can be done by multiplying \(\theta\) with \(\pi/180\).
Figure 3.1. The original heart (blue, solid line) is rotated $90^\circ$ counter-clockwise when multiplied by $i$ (orange, dashed line) and rotated by $-60^\circ$ when multiplied by $\exp(-i\pi/3)$ (green, dotted line). The same can be done to other two-dimensional objects, such as cloud points or images.

Figure 3.2. A complex sinusoid has a period and a magnitude that can be represented as a complex number. The left figure displays the sinusoid as a combination of a cosine (blue, solid line) and a sine (orange, dashed line). The right figure displays the same sinusoid as a three-dimensional spiral.
3.1.2 Quaternions

Quaternions extend complex numbers further to achieve rotations in three dimensions [25, p. 103]. Like complex numbers, quaternions were met with skepticism and considered unpleasing to many mathematicians. Lord Kelvin wrote in 1892: “Quaternions came from Hamilton after his really good work had been done; and, though beautifully ingenious, have been an unmixed evil to those who have touched them in any way, including Clerk Maxwell.” Sir William Rowan Hamilton worked many years on extending complex numbers to achieve three-dimensional rotations. Complex numbers have two parts, allowing rotation in two dimensions, leading him to conjecture that an extension to three dimensions should likewise have three parts, adding an extra imaginary number. Hamilton wrote a letter to his son about this attempt [26]:

“Every morning in the early part of the above-cited month\(^2\), on my coming down to breakfast, your brother William Edwin and yourself used to ask me, ‘Well Papa, can you multiply triplets?’ Whereeto I was always obliged to reply, with a sad shake of the head, ‘No, I can only add and subtract them.’”

In October 1843, Hamilton had an epiphany while walking along the Royal Canal of Dublin with his wife on their way to the Royal Irish Academy. He realized that the object he was trying to invent – the quaternion – required four parts and carved the following equation in the stone of the Broom Bridge [27, p. 5]:

\[
i^2 = j^2 = k^2 = ijk = -1.
\]  

Quaternions take the form \(a + ib + jc + kd\) where \(a, b, c, d\) are real numbers, and, unlike complex numbers, are non-commutative, that is, \(pq \neq qp\) for two quaternions \(p\) and \(q\). For our purpose, just like for complex numbers, we will separate quaternions into two distinct categories: point quaternions (also named pure quaternions) which help represent physical objects in three dimensions – such as cloud points and images (called volumes) – and rotation quaternions (also named unit quaternions). Point quaternions are named pure quaternions because they contain three imaginary parts: \(ix + jy + kz\) for a point with coordinates \((x, y, z)\). On the other hand, rotation quaternions can rotate pure quaternions around the origin – the point at \((0, 0, 0)\). They are better understood under the angle-axis representation; that is, a rotation can be decomposed into an axis of rotation and how much to rotate around this axis, namely, the angle. Figure 3.3 (left) describes the rotation of the minute hand around the axis that originates at the clock’s center and is perpendicular to the plane of the clock. Each minute, the minute hand rotates by an angle of 6\(^\circ\), or \(\pi/30\) radians. Rotation quaternions can take the form \(\cos(\theta/2) + \sin(\theta/2)v\) with \(v\) the axis and \(\theta\) the angle. We will denote rotation quaternions as \((\theta, v)\) for simplicity. Notice how the angle is divided by two in the formula; this is

\(^2\)October 1843.
Figure 3.3. The left plot displays a clock with the minute’s hand rotated by the quaternion \((\pi/30, \vec{x})\), that is, a rotation of 6 degrees around the x-axis. The right plot displays how two quaternions, namely \((\pi/30, \vec{x})\) and \((-\pi/30, -\vec{x})\), correspond to the same rotation, highlighting the double cover property. The wall clock was designed by Broccoletto and is licensed under Creative Commons Attribution.

no mistake and originates from how a rotation is performed. Let \(p\) be a point quaternion and \(q\) a rotation quaternion, the rotation of \(p\) by \(q\) is achieved with \(qpq^*\), where \(q^*\) is the conjugate\(^3\) of \(q\). Since the rotation quaternion is applied twice, the angle must be halved for the complete rotation to occur once.

Interestingly, quaternions double cover the space of rotations: two unit quaternions describing the same rotation always exist. For example, this is shown in Figure 3.3 (right) where rotating the second hand is achieved with both quaternions \((\pi/30, \vec{v})\) and \((-\pi/30, -\vec{v})\).

Although rotations can be achieved through other means, quaternions are easier to manipulate mathematically than matrices and do not suffer the issues that arise when using Euler angles, for instance. Euler angles are a simple way to perform rotations, where an object is rotated sequentially around the \(x\), \(y\), and \(z\) axis. Those rotations are called pitch (around \(x\)), yaw (around \(z\), pointing upwards), and roll (around \(y\)). However, a common issue with this representation is the gimbal lock [27, p. 133], which makes the system lose degrees of freedom with specific configurations of angles.

3.1.3 Dual numbers

Dual numbers are a number system introduced in the 19th century [28]. The numbers take the form \(a + b\epsilon\) with \(\epsilon^2 = 0\). It would be easy (and one would be mistaken) to believe that \(\epsilon = 0\). Let’s instead imagine that \(\epsilon\) is an extremely

\(^3\)For completeness, the conjugate of \(q = a + ib + jc + kd\) is defined as \(q^* = a - ib - jc - kd\)
small number (an infinitesimal), such that squaring makes it zero. Those dual numbers have interesting properties, providing an alternative way of computing derivatives, useful for automatic differentiation. subsection 4.1.1 describes how useful it is to compute the function and its derivative to train, e.g., neural networks and Gaussian processes.

Consider a function $f$ that can be decomposed in a Taylor series:

$$f(a + b\epsilon) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)\epsilon^n}{n!} = f(a) + f'(a)b\epsilon$$  \hspace{1cm} (3.3)

with $f^{(n)}$ the $n$th derivative of the function $f$. The higher order terms cancel when $n \geq 2$ because $\epsilon^n = 0$. This means that evaluating a function with dual numbers computes both the image of a function and its derivative at $a$.

As an example, consider the function $x^2$:

$$(a + b\epsilon)^2 = a^2 + 2ab\epsilon + \epsilon^2 = a^2 + 2ab\epsilon,$$  \hspace{1cm} (3.4)

that shows that the derivative of $x^2$ is $2x$. The chain-rule can be derived using Equation 3.3:

$$g(f(a + b\epsilon)) = g(f(a) + f'(a)b\epsilon) = g(f(a)) + f'(a)g'(f(a))b\epsilon,$$  \hspace{1cm} (3.5)

that is, the derivative of $g(f(x))$ equals $f'(x)g'(f(x))$. For instance:

$$\cos(x^2) = 2x(-\sin(x^2)).$$  \hspace{1cm} (3.6)

This is known as the chain rule, a fundamental component of learning-based methods trained with gradient descent.

A final example is given here for the exponential function’s derivative, which can be computed with

$$\exp(b\epsilon) = \sum_{n=0}^{\infty} \frac{(b\epsilon)^n}{n!} = 1 + b\epsilon,$$  \hspace{1cm} (3.7)

using the Taylor decomposition of the exponential function and because $\epsilon^0 = 1$.

Dual numbers and the exponential function can even be extended to complex numbers, as $\exp(i(\theta + \epsilon)) = \exp(i\theta) + i \exp(i\theta)\epsilon = \exp(i\theta) + \exp(i(\theta + \pi/2))\epsilon$ meaning that the derivative is a rotation by 90 degrees! The derivative is, in fact, tangent to the unit circle\(^4\). This concept will be helpful later when working with the Fourier transform, which helps compute the signal’s derivative when multiplied by $i$.

\(^4\)The YouTube channel 3Blue1Brown made an excellent video describing the concept visually [29].
3.2 Fourier transform

The Fourier transform decomposes a signal into its frequency components. In computer vision, the Fourier transform is used over images, quantifying the periodicity of the signal horizontally and vertically. Often, it is used for time series or signals with periodic components.

The Fourier transform of a signal is defined as an integral [30, p. 5]:

\[ \mathcal{F}\{f(t)\} = \hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} \, dt, \tag{3.8} \]

with \( \mathcal{F} \) the Fourier transform operator, \( t \) the time, \( \omega \) the angular frequency of interest. Intuitively, the Fourier transform computes the correlation between a signal and a complex sinusoid with a frequency of \( \frac{\omega}{2\pi} \) Hertz. When multiple frequencies of the signal are computed, the resulting object is called the spectrum.

Some signals’ Fourier transform can be computed analytically. For instance, let \( \Pi(x) \) be a rectangle function from \(-1\) to \(1\), the Fourier transform is

\[
\mathcal{F}\{\Pi(x)\} = \hat{\Pi}(\omega) = \int_{-\infty}^{\infty} \Pi(x) e^{-i\omega x} \, dx \\
= \int_{-1}^{1} e^{-i\omega x} \, dx = -\frac{1}{i\omega} e^{-i\omega x}\bigg|_{-1}^{1} \\
= -\frac{1}{i\omega} (e^{-i\omega} - e^{i\omega}) = \frac{2}{2i\omega} (e^{i\omega} - e^{-i\omega}) \\
= 2 \frac{\sin(\omega)}{\omega} = 2 \text{sinc}(\omega), \tag{3.9}
\]

which can be deduced using Euler’s formula\(^5\). The resulting function is the cardinal sine function, as shown in Figure 3.4.

We will now describe a few fundamental properties of the transform in Equation 3.9. First, the rectangle function (1 in the \([-1, 1]\) interval, 0 otherwise) is symmetric, making the Fourier transform a real function (as opposed to complex). The integral of the rectangle is its area and is equal to 2, which can also be calculated by evaluating the Fourier transform at \( \omega = 0 \), that is, \( \hat{\Pi}(0) = 2 \). Finally, summing infinitely many complex sinusoids parameterized by \( \Pi \) will reconstruct the rectangle: the transform is invertible.

3.2.1 Discrete Fourier transform

The Discrete Fourier Transform (DFT) is an extension of the Fourier transform where the signal is discretized into integer bins [30, p. 260]. This gives us an algorithm to apply to digitized signals. The result is a spectrum containing

\(^5\)Specifically, \( \sin(x) = \frac{e^{ix} - e^{-ix}}{2i} \).
\( n + 1 \) unique frequencies for an input length of \( n \). At \( \omega = 0 \), the complex sinusoid disappears, explaining why it evaluates the integral of the original signal – also named the area under the curve.

An example of DFT is shown in Figure 3.4, which displays a waveform created by a toy triangle musical instrument\(^6\) along with the signal’s spectrum, where the task is to find the maximum frequency component of the sound. The maximum frequency of the spectrum is 4140 Hertz, which is a relatively high-pitched sound, as expected.

An essential property of the Fourier transform is its ability to perform convolutions. When a signal is convolved with another signal – often called a filter –, the filter will be multiplied by the signal at different shifts. The result

\(^6\) A word to the wise: one may reconsider offering musical instruments to young children.
Figure 3.6. A true signal – a rectangle function at $t = 11$ seconds – is corrupted by heavy Gaussian noisy. The FFT of the noisy signal multiplied by the FFT of a rectangle function performs a convolution that recovers the true signal and allows us to estimate the position and even the shape of the original non-noisy rectangle.

is a new signal showing how correlated the filter is with respect to the signal at various locations. This property helps find patterns, denoising, compression, and computing moving averages. This computation is usually expensive but becomes a simple pointwise multiplication in the Fourier domain. As an example, Figure 3.6 displays a rectangle function placed at $t = 11$ seconds, heavily corrupted by Gaussian noise. However, when convolved with a rectangle function, it is possible to recover the location of our signal, where the cross-correlation is the highest, at $t = 11.466$ seconds. Another interpretation of this operation is that we calculated a moving average, smoothing the noisy signal. In the literature, this is also called a low-pass filter because the high frequencies are discarded, and the cross-correlated signal in Figure 3.6 tries reconstructing the original rectangle function, albeit poorly.

Using the Fourier transform to find better estimates of the signal’s derivative and images by extension is also possible. Let $\cos(t)\exp(-0.04t^2)$ be a function for which we are interested in the derivative

$$\hat{f}'(\omega) = \int_{-\infty}^{\infty} f'(t) \exp(-i\omega t) \, dt \quad (3.10a)$$

$$= f(t) \exp(-i\omega t)|\infty_{-\infty} - \int_{-\infty}^{\infty} -i\omega \exp(-i\omega t) t \, dt \quad (3.10b)$$

$$= i\omega \int_{-\infty}^{\infty} \exp(-i\omega t) t \, dt = i\omega \mathcal{F}\{f(t)\}, \quad (3.10c)$$

where we integrated by part in Equation 3.10b, and assumed the function $f$ to taper off to 0 in Equation 3.10c, that is $f(-\infty) = f(\infty) = 0$. This means that the Fourier transform of the derivative of a signal is equivalent to computing
the Fourier transform of the signal, rotating the frequency components by 90° and scaling them\(^7\).

Computing the derivative of a signal in the Fourier domain – named the spectral derivative – is very efficient, as we shall see in the following example. Figure 3.7 displays the function \( f(t) = \cos(t) \exp(-0.04t^2) \) with \( t \) the time. The function has been heavily subsampled; that is, the function was sampled once every second for 19 seconds (20 data points). The true derivative of the function can be computed mathematically: \( f'(t) = -\sin(x) \exp(-0.04t^2) - 0.08tf(t) \), estimated with the difference quotient formula \( \frac{1}{dt}(f(t+1)-f(t)) \), or estimated with the spectral derivative. The spectral derivative closely matches the true derivative.

### 3.2.2 Fast-Fourier transform

The Fast Fourier Transform (FFT) is an essential improvement to the computational complexity of the DFT algorithm while producing the same result [30, p. 275]. This algorithm was used to compute the Fourier transform of two- and three-dimensional images in Paper IV, as well as in Figures 3.5, 3.6, and 3.7.

The large runtime of various algorithms can be prohibitive in practice, and we may be interested in quantifying how efficient algorithms are. The big \( \mathcal{O} \) notation expresses how the run time of an algorithm grows as a function of its parameters, such as the input length. It is invariant to proportional changes, that is, \( \mathcal{O}(an+b) = \mathcal{O}(n) \) with \( n \) the input length, and \( a, b \) two constants. The Discrete Fourier Transform has a big \( \mathcal{O} \) complexity of \( n^2 \), meaning that

\(^7\)For the same reason, the derivative of \((\sin(fx))' = f \cos(fx) = f \sin(fx + \pi/2)\), where a complex rotation of \( \pi/2 \) becomes a shift in the argument of the original function.
the run time grows quadratically with the input length. A quadratic complexity often becomes prohibitively expensive when the input size is large. The Fast Fourier Transform was invented to circumvent this bound and achieved a big $O$ complexity of $n \log_2(n)$. For instance, for an input length of 1000, the DFT would have a big $O$ of $10^6$ while the FFT would be at $\times 10^4$. It is hard to understate how much of a shorter run time that is. The FFT is believed to be one of the most important algorithms ever developed. Proving that the complexity $O(n \log_2(n))$ is the best possible bound is left as an exercise to the reader.

3.2.3 Fourier transform of images

The Fourier transform can be extended to images by computing the DFT of each row followed by calculating the DFT of each column. The resulting object is a complex image containing all the image frequencies in two dimensions. All the properties cited above still apply in two dimensions (or any number of dimensions). For example, Figure 3.8 shows a synthetic image\(^8\) convolved with a Gaussian filter in the Fourier domain, producing a blurred image. The process removes high frequencies from the original image, with the Gaussian function in the Fourier space acting as a low-pass filter.

3.3 Probabilities and statistics

Probability is the study of how events are likely to occur. This is one of the most useful theories in the research toolbox of many scientific fields, including computer vision, artificial intelligence, and statistical analysis of experiments.

The Bernoulli distribution [31, p. 621] is one of the simplest distributions and describes binary events with a probability $p$ of happening. For instance, tossing a coin has two equally probable outcomes such that $p = 50\%$. The probability mass function (pmf) is

$$p_X(x) = p^x(1-p)^{1-x}$$

and describes the probability of an event $x$ happening. One interpretation of this probability is through the lens of uncertainty: before a toss, any outcome is possible, but once the toss is realized, the uncertainty vanishes, and we are left with the knowledge of the outcome.

Two important statistical summaries are the average – also named the arithmetic mean – and the variance. The mean is a single-point representative of the distribution and is written as

$$\mathbb{E}[x] = \mu = \int_{-\infty}^{\infty} p(x)x \, dx,$$

\(^8\)Code available at https://github.com/npielawski/raytracing.
Figure 3.8. A synthetic image from a raytracing rendering program is transformed in the Fourier domain (top row). A Gaussian filter is generated, and its Fourier transform is computed (middle row). The two Fourier domain images are multiplied by each other, resulting in a convolution in the original space (bottom row). This operation results in a Gaussian blur of the original image.
for a continuous random variable $x$. It is simple to compute the expectation – another term for the mean – of a linear transformation of the variable: $\mathbb{E}[ax + b] = a\mathbb{E}[x] + b$ for two constants $a$ and $b$. Meanwhile, the variance representing the spread around the mean is written as

$$\forall [x] = \sigma^2 = \mathbb{E}[(x - \mu)^2] = \int_{-\infty}^{\infty} p(x)(x - \mu)^2 \, dx.$$  \hspace{1cm} (3.13)

This time, the variance of a linear transformation of a random variable is more complex: $\forall [ax + b] = a^2 \mathbb{E}[x]$ for two constants $a$ and $b$. The variance can also be written with two expectations: $\forall [x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$.

Computing those quantities for discrete random variables can be achieved by replacing the integral sum operator with a sum over the domain. Note that the expectation and the variance are two essential summaries that do not necessarily exist for a given random variable (e.g., the Cauchy distribution).

**Characteristic function**

Summing random variables is important when dealing with statistics, machine learning, and computer vision tasks, and is, unfortunately, more complex than taking the sum of the probability density functions (pdf)$^9$ of the random variables (RVs). The pdf of the sum of two random variables will consist of the convolution of the two RVs

$$p_Z(x) = p_X(x) \ast p_Y(x) = \int_{-\infty}^{\infty} p_X(x)p_Y(x - y) \, dy,$$  \hspace{1cm} (3.14)

for two random variables $X$ and $Y$, and $\ast$ the convolution operator.

We saw in section 3.2 that convolutions could be performed efficiently in the Fourier space, which we can use here too. Let $\phi_X(t)$ be the characteristic function of the random variable $X$ and corresponds to the Fourier transform$^{10}$ of the pdf of $X$

$$\phi_X(t) = \mathbb{E}[e^{itX}] = \int_{-\infty}^{\infty} p_X(x)e^{itX} \, dx.$$  \hspace{1cm} (3.15)

It is possible to invert the characteristic function to find the pdf again with

$$p_X(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_X(x)e^{-itX} \, dx.$$  \hspace{1cm} (3.16)

For example, let us derive the pmf of the sum of $n$ independent and identically distributed (i.i.d.) Bernoulli distributions, for instance, how many heads

$^9$Probability mass functions correspond to the probability of events of discrete variables, while probability density functions correspond to continuous variables.

$^{10}$You may notice a change of sign in the exponential of the Fourier transform here, and it seems purely conventional [32] to use $e^{itX}$ instead of $e^{-itX}$. 

34
can be obtained after tossing a coin \( n \) times. The characteristic function of a Bernoulli random variable \( X \) is

\[
\phi_X(x) = \sum_{t=0}^{1} (1 - p)^{1-k} p^k e^{it} = (1 - p)(1 + p e^{it}),
\]

that we will multiply to each other \( n \) times leading to the sum of \( n \) Bernoulli variables – a new random variable called \( Y \):

\[
\phi_Y(x) = (1 - p + p e^{it})^n = \sum_{k=0}^{n} \binom{n}{k} (1 - p)^k p^k e^{itk}
\]

Finally, inverting the characteristic function of \( Y \) yields

\[
p_Y(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{k=0}^{n} \binom{n}{k} (1 - p)^k p^k e^{itk - it} \, dx
\]

\[
= \sum_{k=0}^{n} \binom{n}{k} (1 - p)^k p^k \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{it(k-1)} \, dx
\]

\[
= \sum_{k=0}^{n} \binom{n}{k} (1 - p)^k p^k \delta(k - 1)
\]

\[
= \binom{n}{k} (1 - p)^k p^k,
\]

proving that the sum of \( n \) Bernoulli variables corresponds to a Binomial distribution, and where we used the following definition of the Dirac delta function

\[
\delta(k - n) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{it(k-n)} \, dt = \begin{cases} 
1 & \text{if } k - n \text{ equals } 0 \\
0 & \text{otherwise}
\end{cases}
\]

**The Central Limit Theorem (CLT)**

The normal distribution is another type of distribution that describes a certain type of uncertainty we can hold about a real number. It is a widespread distribution and was named normal for this reason. The probability density function is [31, p. 625]

\[
\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right),
\]

\[
\phi_{\mathcal{N}}(\mu, \sigma^2) = e^{it\mu - \frac{1}{2}\sigma^2t^2},
\]

with \( \mu \) the mean and \( \sigma^2 \) the variance.
It is not by chance that this probability distribution is typical and is primarily due to the central limit theorem, which states that a sum of independent and identically distributed random variables will eventually converge to a normal distribution (in the limit of \( n \) approaching infinity) [33]. Hence, counts and averages are often assumed to be normally distributed.

Let \( Z_n = X_1 + X_2 + \ldots + X_n \), summing \( n \) iid random variables with mean \( \mathbb{E}[X_i] = \mu \) and variance \( \text{Var}[X_i] \). \( Z_n \) is a random variable with mean \( n\mu \) and variance \( n\sigma^2 \). We can now standardize\(^{11}\) \( Z_n \) such that it has a null mean and unit variance

\[
Z_n = \frac{X_1 + X_2 + \ldots + X_n - n\mu}{\sqrt{n\sigma^2}}
\]

\[
= \sum_{i=1}^{n} \frac{X_i - \mu}{\sigma} \sqrt{n}
\]

(3.23)

with \( Y_i \) the standardized version of \( X_i \).

The characteristic function of \( Z_n \) can be thus be written as

\[
\phi_{Z_n}(t) = \prod_{i=1}^{n} \phi_{Y_i}(\frac{t}{\sqrt{n}}) = \left[ \phi_{Y_1}(\frac{t}{\sqrt{n}}) \right]^n.
\]

(3.24)

Using a Taylor decomposition, the characteristic function of \( Y_i \) is

\[
\phi_{Y_i}(\frac{t}{\sqrt{n}}) = \mathbb{E}[e^{it\frac{Y_i}{\sqrt{n}}}] = \mathbb{E}[1 + it\frac{Y_i}{\sqrt{n}} + \frac{1}{2}(it\frac{Y_i}{\sqrt{n}})^2 + o(\frac{t^2}{n})]
\]

\[
= 1 + \frac{it}{\sqrt{n}}\mathbb{E}[Y_i] - \frac{t^2}{2n}\mathbb{E}[Y_i^2] + o(\frac{t^2}{n})
\]

(3.25)

because \( Y_i \) is standardized, that is, \( \mathbb{E}[Y_i] = 0 \) and \( \mathbb{E}[Y_i^2] = 1 \). The \( o(\frac{t^2}{n}) \) means that the remaining terms are upper bounded and decay at a rate greater than \( \frac{t^2}{n} \).

The characteristic function of the sum of infinitely many random variables then becomes

\[
\lim_{n \to \infty} \phi_{Z_n}(t) = \lim_{n \to \infty} \left( 1 - \frac{t^2}{2n} + o(\frac{t^2}{n}) \right)^n = \exp(-\frac{t^2}{2}),
\]

(3.26)

where the higher-order terms vanish to zero. This is the characteristic function of a normal distribution [34], proving that a sum of random variables eventually

\(^{11}\)A random variable transformed such that it has a zero mean and unit variance is said to be standardized. It is rather unfortunate that normalization refers to transforming a random variable such that the minimum value is 0 and the maximum value is 1.
converges to zero. This proof has been extended to random variables that are independent but not necessarily identically distributed (Lyapunov CLT).

### 3.3.1 Lognormal distribution of microscopy images

Microscopy images can often be much more complex than natural images. For instance, there can be many more color channels\(^\text{12}\), multiple focus planes and time frames, lens distortions, and a high bit-depth: pixel intensity is often recorded using 12 or 14 bits per color channel – compared to regular cameras having only 8 in general. Bit depth enables the collection of richer and higher-quality images and allows for fine post-processing, especially in low-light settings.

One issue when creating learning-based algorithms that take microscope images – as input or ground truth – is that the average amount of light that hits the sensor can be low. Still, there can be a few pixels with very high intensity, which can be problematic in practice, as those pixels will be overrepresented and break the fitting of the parameters. This problem arose in Paper II. One strategy is to preprocess the images by applying the logarithm to each pixel, but this choice is often unmotivated.

The intensities of the pixels in a microscopy image often follow a log-normal distribution; that is, they will follow a normal distribution after the logarithm is applied. This happens in systems with a sequence of transformations with independent, randomly distributed efficiencies. One example is an engine with gears; each gear will have an efficiency between 0 and 100%, which may vary due to wear and heating. In a microscope, the light emerging from a laser may vary over time, the light may scatter in the air in unexpected ways, and the sample and the lenses may absorb, refract and diffract the light differently, especially if the sample is motile. All those efficiencies are by nature multiplicative, and the total amount of light \( T \) absorbed by the sensor becomes

\[
T = \prod_{i=1}^{N} X_i,
\]

with \( N \) variables with random efficiencies distributed as \( X_i \). Taking the logarithm of \( T \) yields

\[
\log T = \sum_{i=1}^{N} \log X_i \xrightarrow{d} \mathcal{N}(\mu, \sigma^2) \quad \text{as} \quad n \to \infty
\]

which converges in distribution to a normal distribution with parameters \( \mu \) and \( \sigma^2 \) due to the central limit theorem. Thus, \( T \) converges to a log-normal distribution.

\(^{12}\)Color often represents the narrow strip of the wavelength of visible light to the human eye, but specific microscopes can measure \( \alpha \) and \( \gamma \) rays, and electron microscopes use electrons instead of light, so the color channel is used as a loose term, here.
Figure 3.9. The probability density functions of a normal distribution $X$ (solid blue line) and its lognormal distributed counterpart $Y = e^X$ (dashed orange line). The lognormal distribution is right-skewed, making random samples potentially large.

This proof motivates transforming the images with the logarithm before using learning-based methods sensitive to outliers since the log-normal distribution is heavy-tailed (i.e., prone to generate rare, high-intensity values) as shown in Figure 3.9 or using loss functions accounting for this type of distribution as done in Paper II.
4. Machine learning in digital image processing

Machine learning aims at comprehending real-world systems by matching them to mathematical models that we can better understand and manipulate algebraically. Modelization involves finding suitable or likely models to emulate those complex systems. However, the models can and often require degrees of freedom – or parameters – to account for the variability of the real world. Furthermore, measurements are inherently noisy and often too complex to fully account for in a simple mathematical representation, limiting the search for informative models to reasonable approximations.

Probabilistic models are usually written as a fixed model (what can be explained) and some random noise (what can’t be explained by the model). A model fit on some data is called a hypothesis making the search for likely hypotheses at the heart of machine learning.

A model maps a set of features – usually denoted as $x$ – about a sample to a quantity we are interested in estimating – denoted as $y$. For instance, we may wonder how likely a cancerous cell is to start migrating (probability of migration $y \in [0, 1]$) by looking at $n$ morphological features extracted from bright-field microscopy images ($x \in \mathbb{R}^n$).

For a real-world model $y = f(x) + \epsilon$ corrupted with additive random noise $\epsilon$, we will mainly focus on learning-based models of the form

$$\hat{y} = f_\theta(x)$$  \hspace{1cm} (4.1)

with $f_\theta$ a model with a set of parameters $\theta$ which estimates $\hat{y}$. The goal is to find the set of parameters $\theta$ which maximizes a loss function $\mathcal{L}(y, \hat{y})$, minimizing the error between $y$ and $\hat{y}$. Note that the noise of a system is not always additive, but this choice is widespread in practice.

This chapter focuses on modelization, loss functions, and fitting in the following sections. First, linear methods can be expanded in various ways, for instance, kernelization and non-linear optimization (e.g., gradient descent for deep neural networks). The kernel method enables the application of Gaussian processes to a wide range of tasks, including bayesian optimization in section 4.2. Finally, learning-based methods often involve multiple sources of uncertainties, for instance, in the data, the model’s parameters, and its estimates, which will be the focus of section 4.3.
4.1 Inductive biases and model architectures

The field of computer vision (CV) has dramatically evolved in the past decade, changing how we think and how new methods are developed. While CV specialists used to hand-craft relevant features, requiring expert knowledge about the topic and data, the field moved to learned features with convolutional neural networks winning several contests in 2011 with much higher accuracies than the previous classical approaches [35]. The neural networks’ architectures were essential in reaching those impressive improvements, surpassing human-level accuracy in a traffic sign classification benchmark. Still, little was known about which architectural choices performed best, and many decisions were purely empirical. Later, methods were developed to optimize neural network architectures using random search, evolutionary methods, and reinforcement learning. There are still many open questions today, but we will start by discussing how interpreting the different layer types as inductive biases provide helpful insight for choosing them. Using previously published, proven architectures is common since they often perform well on different datasets and data modalities. Paper R1 reviews previously published deep learning methods applied to image cytometry and can provide a starting point when exploring new image modalities and computer vision tasks.

Inductive biases are a set of assumptions that the model will use (as opposed to learning) to perform a task. Understanding the data let us decide which priors we want to induce on the model, which increases the generalization capabilities, accuracy, and, generally, the performance of the task. For instance, linear regression may become suitable if we know a linear relationship exists between the features $x$ and the predicted variable $y$. Occam’s razor is an empirical rule often used when choosing model architectures, where simpler models are privileged over other models that perform as well but are more complex [36]. Weight decay is an example of this type of prior, where the parameters of a model are encouraged to be small or sparse during training [37, p. 144].

Neural networks are often composed of stacked layers, which makes the assumption that features are hierarchical. For example, in image classification, the first layers will detect low-level features, such as edges, colors, and high-frequency intensity changes; the successive layers may be sensitive to textures and angles, while the deeper layers may focus on semantic information, such as people, animals, vehicles, etc.

Common inductive biases used in neural networks are as follows:

- **Linear layers** assume a linear relationship between input and output features.
- **Non-linear activation functions** add the assumption that the features are not linearly separable.
- **Convolutional layers** which assume that features are either spatially or temporally correlated and provide translation-invariance (up to integer shifts).
- **Recurrent layers or long short-term memory cells** [38] assume that temporal information gives a context when certain past features are temporally salient and vital.

- **Batch-normalization** forces that the feature’s intensities are consistent across input samples, i.e., a zero mean and unit standard deviation.

- **Attention layers** assume that some features are more important than others and build a relationship graph between them.

- **Pooling layers** discard high-frequency features, often trading spatial resolution for feature richness.

- **Dropout layers** prevent the model from relying on single – or a specific subset of – features.

Neural network architectures can also encode some prior information about the data; for instance, a typical architecture named U-Net [39] – due to its shape reminiscent of the form of the letter U – encodes a hierarchical representation of features across different resolutions. This helps the neural network to relate low-level features, such as pixel intensities, to high-level features, such as semantic information. As a result, this model is highly performant on tasks such as segmentation or depth estimation.

### 4.1.1 Model fitting

When a model is chosen, the parameters of the model need to be tuned carefully to fit the data. How fitness is determined is an important question we will develop in subsection 4.1.2. Note that the parameters are not necessarily deterministic but can belong to a distribution as often done in Bayesian statistics.

Linear regression is an interesting fitting example because it can be solved analytically, on top of its widespread usage. This formulation will be useful when expanding the method to kernels, as discussed in subsection 4.2.2. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a dataset with $n$ rows and $d$ dimensions, representing $n$ samples with $d$ features each, and $\mathbf{y} \in \mathbb{R}^n$ the ground-truth containing $n$ variables that we are interested in predicting. Let us assume that $\mathbf{y}$ is corrupted with i.i.d. Gaussian noise with zero mean and unknown variance $\sigma^2$. We are interested in finding some linear parameters $\mathbf{w} \in \mathbb{R}^d$ that help us produce an estimate of $\mathbf{y}$, which can be achieved using the maximum likelihood estimation (MLE) method, or equivalently, minimizing the negative log-likelihood\(^1\) as following

$$\arg\min_{\mathbf{w}} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \arg\min_{\mathbf{w}} - \log \mathcal{N}(\mathbf{X}\mathbf{w} - \mathbf{y}, \sigma^2 \mathbf{I}_n)$$

$$= \arg\min_{\mathbf{w}} \log \frac{n}{2} \log(2\pi) + \frac{n}{2} \sigma^2 + \frac{1}{2\sigma^2} (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y}),$$

(4.2)

\(^1\)Although there is an information-theoretical interpretation for using the negative log-likelihood, we perform this operator for simplifying the mathematical derivation of $\mathbf{w}$. 

41
Since the function is convex, which can easily be proven by showing that the second derivative is non-negative, the most likely point at the maximum will have a null derivative which we can exploit to find the best parameters $\hat{w}$:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\partial}{\partial \mathbf{w}} \frac{1}{2\sigma^2} (\mathbf{Xw} - \mathbf{y})^\top (\mathbf{Xw} - \mathbf{y}) = 0 \quad (4.3)$$

$$\frac{1}{\sigma^2} \mathbf{X}^\top (\mathbf{Xw} - \mathbf{y}) = (\mathbf{X}^\top \mathbf{Xw} - \mathbf{X}^\top \mathbf{y}) = 0 \quad (4.4)$$

$$\mathbf{X}^\top \mathbf{Xw} = \mathbf{X}^\top \mathbf{y} \quad (4.5)$$

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \mathbf{X}^+ \mathbf{y} \quad (4.6)$$

This proves that the best estimate of the parameters $\mathbf{w}$ is the Moore-Penrose pseudoinverse [40] of $\mathbf{X}$ (written $\mathbf{X}^+$) multiplied by $\mathbf{y}$. In bayesian statistics, for instance, we are often interested in evaluating a population of likely parameters instead of a single-point estimate $\hat{\mathbf{w}}$ as done above by MLE.

For non-linear methods, such as neural networks, the likelihood will not be convex, and closed-form solutions do not exist. In this setting, parameters are fitted using gradient descent, iteratively updating the parameters toward a better solution. However, finding the derivative of complex functions can be challenging, and current frameworks provide automatic differentiation to compute the likelihood’s derivative with respect to the parameters.

One issue with powerful methods such as neural networks and kernel methods is their tendency to overfit; that is, they will perfectly estimate the points in the dataset but perform poorly when given another dataset. This is often addressed by using regularization methods such as weight decay, where a Gaussian prior is added to the learned parameters:

$$\argmin_{\mathbf{w}} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \argmin_{\mathbf{w}} - \log \mathcal{N}(\mathbf{Xw} - \mathbf{y}; 0, \sigma^2 \mathbf{I}_n), \mathcal{N}(\mathbf{w}; 0, \sigma_0^2 \mathbf{I}_d)$$

$$= \argmin_{\mathbf{w}} \frac{1}{2\sigma^2} (\mathbf{Xw} - \mathbf{y})^\top (\mathbf{Xw} - \mathbf{y}) + \frac{1}{2\sigma_0^2} \mathbf{w}^\top \mathbf{w}, \quad (4.7)$$

Proceeding with MLE yields the new estimator of the regularized parameters

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_d)^{-1} \mathbf{X}^\top \mathbf{y}, \quad (4.8)$$

with $\lambda = \sigma_0^{-2}$. Note that the matrix inverse is in $\mathbb{R}^{d \times d}$, which is computationally inexpensive for small dimensional spaces. This procedure is called maximum a posteriori (MAP) instead of MLE since we added a prior distribution to the weights we estimate. Another name is ridge regression because of the diagonal elements – creating a ridge – added to the matrix to invert.

4.1.2 Loss functions

Loss functions assess the fitness of a model for a set of predictions. In a supervised setting, just like teachers help students learn by providing them with
relevant feedback, loss functions help models quantify how far off predictions are by comparing the estimates of a model with the ground truth. Loss functions for supervised training typically take the form $\mathcal{L}(f(x), y)$ for a model $f$, input data $x$, and ground truth $y$.

**Regression loss functions**
Regression predicts values in a set $S \subseteq \mathbb{R}^d$. Examples of domains are $[0, 4000]$, or $\mathbb{R}^2$. A common default choice for regression is the mean squared error. Let $\hat{y} \in \mathbb{R}^{n \times d}$ be $n$ $d$-dimensional vectors predicted by a model, $y \in \mathbb{R}^{n \times d}$ $n$ $d$-dimensional ground truth vectors, the mean-squared error loss function is

$$\text{MSE}(y, \hat{y}) = \mathbb{E}[(\hat{y} - y)^2] = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2,$$

which corresponds to the MLE of a normal distribution. The expectation is taken over the dataset, meaning we attempt to minimize the mean of the prediction error squared during training.

Another common loss is the mean absolute error

$$\text{MAE}(y, \hat{y}) = \mathbb{E}[|\hat{y} - y|] = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|,$$

which corresponds to the MLE of a Laplace distribution. The MSE leads to the regression function $f(x) = \mathbb{E}[Y|X = x]$ while the MAE leads to the conditional median $f(x) = \text{median}(Y|X = x)$, two different statistics. Often, the MAE is chosen over the MSE due to its resilience in the presence of outliers.

Finally, we used the following loss function in Paper II because the ground truth noise was assumed to be log-normally distributed (see subsection 3.3.1 for the motivation)

$$\text{LMSE}(y, \hat{y}) = \mathbb{E}[(\hat{y} - \log y)^2] = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - \log y_i)^2,$$

corresponding to the MLE of a log-normal distribution.

**Classification loss functions**
For classification, the model’s output is a probability prediction $\hat{p} \in [0, 1]^d$. A non-linear activation such as a sigmoid (for binary classification) or a softmax (for multi-class) squash the model’s output, corresponding to a probability for each class. It is hoped that the predicted probability corresponds to the model’s accuracy; for example, if a model predicts that a cell is 20% likely to be cancerous, the accuracy of a model on such instance should also be 20%. However, there is evidence that recent (post-2016) deep learning models only have this ability before a calibration step (see section 4.3).
For classifying two independent classes, the most common loss function used is the binary cross-entropy

$$
BCE(y, \hat{y}) = \mathbb{E}[(y \log(\sigma(\hat{y})) + (1 - y) \log(1 - \sigma(\hat{y}))],
$$

(4.12)

with \( \sigma(x) = \frac{1}{1 + \exp(-x)} \) the sigmoid function. This loss corresponds to the MLE of a Bernoulli probability distribution.

Generalizing to multiple classes, the multi-class cross entropy for \( K \) classes is

$$
MCE(y, \hat{y}) = \mathbb{E}[\sum_{k=1}^{K} y_k \log(\text{softmax}(\hat{y}_k))],
$$

(4.13)

corresponding to the MLE of a categorical distribution.

The softmax ensures that the predicted probabilities for the \( K \) classes sum to 1. Just like the sigmoid function, this loss makes sure the probabilities of each class lie in the interval \([0, 1]\), but also normalizes the probabilities such that the sum is 1:

$$
\text{softmax}(x) = \frac{\exp(x)}{\sum_{i=1}^{K} \exp(x_i)} = \frac{\sigma(x)}{\sum_{i=1}^{K} \sigma(x_i)}
$$

(4.14)

**Loss functions for representation learning**

The triplet loss function allows training deep learning models to project high-dimensional data points into low-dimensional embeddings. The points become part of a feature space where distances become meaningful, which is useful for dimensionality reduction or metric learning. For instance, two images of a cell might be very different by looking at the pixel intensities, but they might correspond to the same cell and thus be close in the feature space. This representation learning can be achieved with the triplet loss [41]

$$
L_{\text{triplet}}(x, x^+, x^-) = \max(0, ||x - f(x^+)|| - ||x - f(x^-)|| + m),
$$

(4.15)

with \( x \) an anchor, \( x^+ \) a positive example, \( x^- \) a negative example and a margin \( m \). The loss attempts to minimize the distance between similar points (the anchor and the positive example) in the feature space while keeping dissimilar points far apart (the anchor and the negative example). Figure 4.1 shows an example of a triplet loss used on a set of points. The points are randomly scattered at initialization and form clusters after training. Distances in the feature space become meaningful, representing similarity.

Representation learning has been used in Paper III, where a loss similar to the triplet loss generates a feature space common to images originating from different sensors (e.g., microscopes or cameras).

**Other loss functions**

Many other loss functions are available for various purposes, and not all correspond to the MLE of some probability distribution. Notably, the min-max
loss allows training generative adversarial networks [42] (GANs) by creating a game between two models: one to generate synthetic images and another to discriminate between fake and real images. During training, the generative network will learn to improve the quality of the generated images to deceive the other model. In contrast, the other model learns to discriminate the generated images better.

The default loss functions used in practice often include MSE for regression and MCE for classification. Choosing the proper loss function can be performed by optimizing a metric (e.g., accuracy) on a validation set, making the loss an additional hyperparameter.

4.2 Kernel methods and bayesian optimization

When trying to find parameters that maximize a black-box function that is too expensive to evaluate, using standard strategies such as random search can be too slow for any practical purpose. Bayesian optimization is a method for sequential global optimization that offers the promise of faster and cheaper convergence to the best set of parameters. Often, the method is characterized using Gaussian processes, relying heavily on the theory of kernels.

This section provides a way to extend ridge regression to nonlinear manifolds using kernels and how this theory can be used for Gaussian processes and Bayesian optimization. This theory is at the core of Paper IV, where we
used Bayesian optimization to find the best rotation parameters in an image registration task.

4.2.1 Kernel methods
The MAP estimator for linear regression using a Gaussian parameter described in Equation 4.8 can be reformulated in the following way

\[ \hat{w} = (X^T X + \lambda I_d)^{-1} X^T y \]
\[ = X^T (XX^T + \lambda I_N)^{-1} y \]
\[ = X^T \hat{a}, \]  
(4.16)

where \( \hat{a} = (XX^T + \lambda I_N)^{-1} y \). This formula can be derived using Searle’s identity [43] \( (A + BB^T)^{-1}B = A^{-1}B(I + B^T A^{-1}B)^{-1} \) and is known as the dual representation of ridge regression. For a dataset of \( N \) entries with \( d \) dimensions, we are seemingly worse off with the dual representation, as the matrix to invert is of size \((N, N)\) instead of \((d, d)\) if \( d \ll N \) — which is often the case.

Whereas before, we were attempting to minimize the loss function in Equation 4.7, the problem can be reformulated as a maximization over the space of \( a \) (and discarding the \( \sigma^2 \) parameter)

\[ \arg\max_a \mathcal{L}(\hat{y}, y) = \arg\max_a -\frac{1}{2} a^T (XX^T + \lambda I_N)^{-1} a + a^T y. \]  
(4.17)

Both formulas from Equation 4.7 and Equation 4.17 return the same global optimum and provide a direct, one-to-one mapping between \( w \) and \( a \) [44]. The advantage of this formulation is that we are now working \( XX^T \) which gives us a base to apply the kernel trick.

The kernel trick
It is possible to extend ridge regression by transforming the input variable to the model into other nonlinear spaces, where \( X \) becomes \( \varphi(X) \). For instance, one can use a geometric progression to project \( X \) (assuming that \( X \) is a \((N, 1)\) matrix) into an \( M - 1 \) degree polynomial, where each row \( X_{x,i} = \begin{pmatrix} 1 & X_{1,j}^2 & X_{1,j}^3 & \cdots & X_{1,j}^{M-1} \end{pmatrix} \). The result is a polynomial regression of the fourth degree as shown in Figure 4.2 (upper right), upgrading the linear regression (upper left).

It is now possible to improve on the original model for linear regression by replacing the parameters with the alternative formulation from Equation 4.16

\[ f(x)^T = w^T \varphi(x) = y(\Phi^T \Phi + \lambda I_N)^{-1} \Phi^T \varphi(x) \]
\[ = y(K + \lambda I_N)^{-1} \kappa(x), \]  
(4.18)
with $\Phi$ the projected $X$, $K(x_i, x_j) = \varphi(x_i, x_j)$ a kernel function, and $\kappa(x) = K(x, x)$. This new formulation is called the kernelized ridge regression, where $K$ is a kernel defining an implicit relationship between points. The kernel defines an inner product between two points, potentially in a different space. For instance, the projection space can be infinite-dimensional and still form a valid kernel if a closed-form solution exists.

Figure 4.2 (bottom row) shows a polynomial kernel $K(x_i, x_j) = (x_i^T x_j + 1)^M$ enabling the nonlinear fitting of the points with a polynomial expansion. This reproduces the standard polynomial fit in the top right figure when $M = 4$. These methods allow using a wide range of kernels, describing relations between pairs of objects, and extending the regression framework to other types of structures, for instance, graphs, text, and non-euclidean spaces.

### 4.2.2 Common kernels

Kernels are at the core of methods such as kernel ridge regression and Gaussian processes (see next section), allowing the nonlinear fitting of data points. In addition, kernels generalize the concept of positive-definite functions and can expand the capacities of linear fitting methods by projecting the data points in potentially infinitely-dimensional space.

Figure 4.3 displays how kernels influence the mean, standard deviation, and samples of Gaussian processes. We sampled the function $f(x) = \cos(x) \exp(-0.04x^2)$ at eight random locations and fit a GP with different kernels. Some kernels better fit the actual function than others because they act as more accurate priors. In other words, they better represent how the underlying
model behaves overall. For instance, if it is known that sales increase at the end of the year, a periodic kernel will be particularly suitable for predicting next year’s December sales based on previous end-of-year sales.

A non-exhaustive list of kernels is described below, starting with the squared exponential kernel

$$K_{SE}(x, x') = \exp\left(-\frac{1}{2l^2} \|x - x'\|_2^2\right).$$

(4.19)

This kernel produces smooth curves that are infinitely derivable and is commonly used in practice.

The exponential kernel is defined as

$$K_E(x, x') = \exp\left(-\frac{1}{l} \|x - x'\|_1\right),$$

(4.20)

producing rough curves that are nowhere differentiable.

Notably, the Matérn kernel has been invented to produce curves that are more realistic than the Gaussian kernel for many tasks and is defined as

$$K_{\text{Matérn}}(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{l} \|x - x'\|_2\right)^\nu \text{K}_\nu \left(\frac{\sqrt{2\nu}}{l} \|x - x'\|_2\right),$$

(4.21)

where the $\nu$ parameter controls the smoothness of the kernel, and the curves are $\lceil \nu - 1 \rceil$ times continuously derivable. The exponential kernel is a special case of this kernel when setting $\nu$ to $\frac{1}{2}$. Computing the limit as $\nu \rightarrow \infty$ recovers the squared exponential kernel, although it doesn’t belong to the Matérn family, which only models functions with finite smoothness [45, p. 51].

**Kernel parameters**

Kernels have parameters such as the lengthscale controlling how much the Gaussian process is allowed to vary, the level of noise for the input data or the covariance, or the mean. Those parameters are often chosen by maximizing the likelihood of the Gaussian distribution under a given dataset. Quasi-newton methods can be used or simple gradient descent as performed in Paper IV for the optimization. Sometimes, a multi-start optimization scheme is used as well to avoid local optima. Figure 4.4 shows the effect of different parameters on the resulting Gaussian process’ covariance function, mean, and variance. The parameters affect the behavior of the GPs, and thus may have a considerable impact when processing and analysis of the results.

**Wrapped manifolds**

If the space of the kernel is not euclidean, for instance, lying on a (hyper)sphere, the default kernels may become inappropriate. It is possible to transform the kernel – the operation is often called warping in the literature
Figure 4.3. A function is sampled at eight random locations. Different kernels generate different priors, which result in different fit Gaussian processes (shown in the left column). The right column displays random functions sampled from the Gaussian processes, highlighting the differences in variations as a function of the uncertainty.
Figure 4.4. The lengthscale $\lambda$ of the kernels and the noise $\sigma^2$ are two parameters affecting the mean, covariance, and fit of the Gaussian process. The lengthscale controls the rate of variation of the GP, and a high value will cause smooth lines to emerge by controlling how much nearby points are correlated. The noise controls the breadth of the variance when there is no information. Here, the data points are assumed to be noiseless, and the GP mean is set to zero.
Figure 4.5. A Gaussian process with a Matérn kernel is fit to 8 data points (top row). When the function of interest is cyclic, such a kernel cannot account for the wrapping of the space around the unit circle. For this reason, periodic kernels need to be used to achieve proper wrapping, making the mean and standard deviation of the Gaussian process continuous around the circle (bottom row).

– to account for these new constraints [45, p. 58]. For example, Figure 4.5 shows the same Gaussian process fitted to some data points lying on a two-dimensional circle. Unlike the warped one, the default kernel does not account for the periodicity of the space. In this case, the transformation was achieved by mapping the one-dimensional points to a two-dimensional circle:

\[
x' \mapsto \begin{bmatrix} r \cos(x) \\ r \sin(x) \end{bmatrix},
\]

with \( r = \frac{p}{2\pi} \), and \( p \) the periodicity of the function. In this example, \( r = 1 \) because the period of the function is \( 2\pi \).

4.2.3 Gaussian processes

Gaussian processes (GPs) are stochastic processes extending multivariate Gaussian distributions to model functions on infinite domains. They inherit the pleasant mathematical properties of the Gaussian distribution allowing multiple quantities to be derived analytically.

One can specify a Gaussian process in the following way

\[
p(f) = \mathcal{GP}(f; \mu, K),
\]
with an objective function over an arbitrary infinite domain \( f : \mathcal{X} \to \mathbb{R} \), a mean function \( \mu : \mathcal{X} \to \mathbb{R} \) and a semidefinite covariance function (or kernel) \( K : \mathcal{X} \times \mathcal{X} \to \mathcal{R} \). The GP thus corresponds to a probability distribution over functions. The mean function determines the expected value of the GP, while the covariance function determines how the deviations from the mean occur [45, p. 15]. When the domain is finite, the GP corresponds to a multivariate Gaussian distribution, as shown in Figure 4.6 (three upper rows). Extending to infinite domains yields the case we are interested in (bottom row).

When a dataset is available, it is possible to condition the GP on the sampled points and integrate them into the process. The result is a fitted GP, where the mean function follows the sampled data points, and the covariance decreases due to the newly integrated information that reduces the uncertainty about what is happening in those regions.

The joint distribution is defined as

\[
p(f, y) = \mathcal{GP} \left( \begin{bmatrix} f \\ y \\ \mu \\ K \\ \kappa^\top \\ C \end{bmatrix} \right),
\]

(4.24)

with \( \mu \) the mean function of \( f \), \( m \) the mean function of \( y \), \( K \) the kernel of \( f \), \( C \) the kernel of \( y \) and \( \kappa \) the kernel between \( f \) and \( y \).

The posterior distribution conditioned on a dataset \( D \) becomes

\[
p(f|D) = \mathcal{GP}(f; \mu_D, K_D),
\]

(4.25)

for which we can compute the updated mean function and kernel

\[
\mu_D = \mu(x) + \kappa(x)^\top C^{-1}(y - m);
\]

\[
K_D(x, x') = K(x, x') - \kappa(x)^T C^{-1} \kappa(x').
\]

(4.26)

Figure 4.7 shows a GP before and after updating the mean and kernel with the formula above, where the data points are considered noise-free. After fitting the GP to eight data points, the GP follows the curve nicely.

Gaussian processes are interesting when working with statistical modeling as they provide a way to approximate functions and their uncertainty around poorly explored regions, which is helpful for Bayesian optimization, as shown in Paper IV.

4.2.4 Acquisition functions

Black box optimization is an active area of research where we are interested in finding the parameters that maximize a function. Unfortunately, the black box function is often considered slow to evaluate, rendering any classical search for the best parameters slow by extension. Bayesian Optimization (BO) can thus be used to find the maximum of the function efficiently. Acquisition functions determine which candidate point is worth exploring at each step. A general
\textbf{Figure 4.6.} Different covariance matrices (kernels) create different Gaussian processes. The left column displays the covariance of a point decreasing as the distance increases from the neighbors. The second column displays the covariance matrix, which describes the covariance between pairs of points. The last column displays a sample from the Gaussian process, where different covariance matrices produce different functions of different smoothness. The last row shows that extending this idea to a continuous domain is possible.
Figure 4.7. Fitting a Gaussian process on some points requires updating the mean and covariance matrix. The top row shows a GP before fitting the sampled data points, whereas the bottom row shows the fitted GP. The covariance matrix after fitting has a low variance in the region where the points are sampled because the points are assumed to be noise-free.
A common acquisition function is the expected improvement (EI), which can be computed analytically. This acquisition function gives high importance to points potentially greater than the maximum of the points $\phi^*$ that have already been sampled.

In the case where the points are noise-free, the acquisition function can be computed as

$$
\alpha_{EI}(x; D) = (\mu - \phi^*)\Phi\left(\frac{\mu - \phi^*}{\sigma}\right) + \sigma\phi\left(\frac{\mu - \phi^*}{\sigma}\right),
$$

(4.27)

with $\mu$ the mean of the GP, $\sigma$ the square root of the diagonal elements of the kernel, $\phi^*$ the maximum value of the sampled points, $\Phi(x)$ and $\sigma(x)$ the cumulative distribution function and probability distribution function of a normal distribution, respectively. A derivation of the EI acquisition function can be found in [45, p. 159].
**Figure 4.8.** A Gaussian process is first fitted to a set of points (top row). The expected improvement acquisition function (bottom row) can then be analytically computed and convey how interesting each point in the domain is to be explored or exploited. The maximum of the acquisition function $x_{\text{candidate}} = -0.033$ is chosen to be explored next, which is very near the maximum of the original function ($x^* = 0$).

Figure 4.8 shows the EI acquisition function on a set of points and is used to find the best next point to explore. The process provides a good candidate near the actual global maximum of the function.

### 4.3 Uncertainty estimation

Uncertainty estimation is concerned with expanding the capacities of a model to not only give an estimate of a quantity but also complement it with uncertainty. This feature is a prominent area of research in frequentist and bayesian statistics and has grown in recent years for deep learning models. The focus started as a way to increase the robustness of the prediction in settings where the data is noisy and improve the interpretability of the models and their predictions. Uncertainty estimation methods were used in Paper II.

When fitting a model to some data, it is possible that the model is not able to perfectly represent the actual, underlying model, leading to model uncertainty. At the same time, the training procedure might not find the best possible model, leading to approximation uncertainty. Finally, a model’s prediction can carry a sense of uncertainty, which leads to predictive uncertainty. Model uncertainty and approximation uncertainty are often combined under the notion of epistemic uncertainty, where the model is uncertain due to a lack of knowledge and can be reduced through additional information. When the uncertainty originates from a lack of information in the data inputted to the model, the uncertainty is categorized as aleatoric and can’t be reduced.
**Figure 4.9.** The parasol mushroom (left, a.) is considered a choice edible mushroom and can be differentiated from the inedible shaggy parasol mushroom (center, b.) by looking at the texture of the stems – a snakeskin pattern or a plain white color. Epistemic uncertainty refers to the required knowledge to recognize the mushrooms and discriminate them from one another. Aleatoric uncertainty refers to the lack of information in the input image, as shown in the last image (right, c.), where the stem is not visible, rendering the identification difficult.

**Aleatoric and epistemic uncertainties**

Figure 4.9 depicts a situation where epistemic and aleatoric uncertainties can be differentiated and explained. We will assume that the reader is not knowledgeable in mushroom picking. Determining the species of a mushroom is essential when considering whether it is edible or poisonous – and potentially deadly. Two mushroom species are shown in a. and b. The parasol mushroom is a choice edible mushroom, while the shaggy parasol mushroom can cause stomach upset. A beginner might be unable to discern which is edible, leading to epistemic uncertainty. After being taught that the edible parasol mushroom has a snakeskin pattern on the stem, unlike the bare white-colored stem of the shaggy parasol, the uncertainty is reduced, and the epistemic uncertainty is reduced (at least for differentiating those two mushrooms). However, in figure c., determining the mushroom’s species is difficult, not because of a lack of knowledge, but because of a lack of information in the image: the stem is not visible. This leads to aleatoric uncertainty.

There are multiple ways to implement aleatoric and epistemic uncertainties in practice; in Paper II, we focused on the implementation by Gal et al. [46] where the aleatoric uncertainty is obtained by modifying the model to have

---

2For legal reasons we are required to disclose to the readers not to pick the mushrooms in the nature after reading this thesis, and use a proper manual written by a certified mycologist.
two prediction heads: one for the estimation of the ground-truth and the other for estimating the uncertainty. A specific loss function propagates both the error and uncertainty to the weights. This technique often yields uncalibrated uncertainties, however (see next section). Finally, the epistemic uncertainty is obtained by averaging the model’s output with dropout enabled such that the network’s output is probabilistic.

**Model calibration**

Current neural networks have lost their capacity to provide calibrated predictions; that is, the probability of the estimates should be equal to the accuracy of the model on a test dataset [47]:

$$p_\theta(y \mid X, p) = p, \forall p \in [0, 1],$$

(4.28)

for a model parameterized with weights $\theta$ and a target probability $p$. For instance, in a binary prediction task, a model should have an accuracy of 40% on the set of images predicted to contain class 1 with 40% of probability.

Multiple methods are used for calibration, notably Platt’s scaling [47], and conformal prediction [48], where the model’s output is corrected to obtain calibrated, meaningful probabilities.
5. Contributions

This chapter dives into the details of the papers in this thesis. The theory laid out in Chapters 2, 3, and 4 is used to describe, explain and motivate the methods presented in Papers I-V. This chapter is divided into three sections, each discussing a distinct theme. The first theme is deep learning methods for biomedical applications, where artificial neural networks are used for dense (image) prediction through patch segmentation (Paper I) and pixel regression (Paper II). Next, the multimodal image registration theme is discussed and provides methods for multimodal image registration using deep learning (Paper III) and bayesian optimization (Paper IV). Finally, methods for visualization of spatial transcriptomics experiments are discussed in the last section, presenting TissUUmaps 3 (Paper V).

5.1 Deep learning methods for biomedical applications (Papers I and II)

Introducing Hann windows for reducing edge-effects in patch-based image segmentation

Deep learning consists of a set of powerful techniques allowing researchers to discover complex nonlinear relationships in datasets but at a high computational cost. One such issue occurs when performing segmentation – classifying each pixel as belonging to a specific semantic class – and pixel regression – predicting a continuous value for each pixel, e.g., depth estimation of real-world images. Deep learning methods trained on such datasets have a hierarchical architecture preserving the spatial relevance of each feature in feature maps, causing them to have a substantial memory footprint. At the same time, training is often accelerated with dedicated hardware such as graphic cards because they can perform parallelized computations. Unfortunately, this specific hardware often has limited built-in – i.e. not extensible – memory available, while biomedical images can often be very large in the order of gigapixels. One possible solution is to cut the images into patches and segment the said patches instead of the whole images. The segmented patches can be averaged to reconstruct the full-sized prediction image, but since the model only has access to the local information present in the patch, this may lead to edge artifacts when reconstructing the full-sized predicted image. We proposed in Paper I to use overlapping patches weighted by Hann windows to reconstruct the segmented
image and reduce edge effects. The contribution of each patch is maximized at the center of the patch, where the signal-to-noise ratio is highest and tapers off as a function of the distance from the center. Figure 5.1 depicts our method applied to three one-dimensional synthetic signals. The sum of the Hann windows is one, showing that the method preserves the energy of the original signal while reconstructing the original signal accurately. In the paper, a U-net neural network architecture was used to segment cells based on hematoxylin and eosin (H&E) stained tissues. We experimented with different window functions and showed an improvement in accuracy when using the windowing, and the best recorded accuracy was achieved when using Hann windows.

A boilerplate code is available at https://gist.github.com/npielawski/7e77d23209a5c415f55b95d4aba914f6, showcasing the method on a synthetic two-dimensional image under the open-source permissive license MIT.

**In Silico Traction Force Microscopy**

We collaborated with biologists to use deep learning methods to study traction forces. Traction Force Microscopy (TFM) is a recently developed method to describe better and study cell migration as described in subsection 2.2.6. This type of microscopy produces a dense vector field representing the forces with a magnitude and orientation. We used a deep learning model named dense U-Net to predict the magnitude of the traction forces based on a fluorescence microscopy image of the cell in Paper II. The model is based on the dense U-Net architecture presented in [49] and features a hierarchical representation of the spatial features in a U-Net combined with dense blocks [50]. This architecture has a higher performance while converging faster than the original U-Net [39].

The traction forces were first computed using the numerical approach presented in [18]. The distribution of the intensities was experimentally determined to follow a log-normally distribution (more details are discussed in subsection 3.3.1), and some high-intensity artifacts were also occurring, thus warranting the use of a logarithmic transformation of the force images. During training, a modified version of the mean-squared-error loss function was used by computing the maximum likelihood estimator of a log-normal distribution. We used the uncertainty estimation methods from [46] to complement the estimation of the forces with an aleatoric (data uncertainty) and epistemic uncertainty (knowledge uncertainty). Those uncertainties were useful for deciding whether faulty predictions were due to the small amount of imaged cells – which would cause a high epistemic uncertainty – or due to a lack of information in the fluorescence channel – resulting in high aleatoric uncertainty. Due to the assumption of log-normality, the confidence intervals described in the paper are non-symmetric due to the skewness of the log-normal distribution (see Figure 3.9).
Figure 5.1. Three overlapping sinusoidal functions (left column) are depicted as an example of signals corrupted with noise due to edge effects. Those effects occur due to the model not having information past the edges of the signal. It is possible to craft Hann windows (middle column) to taper the noise by changing each signal’s contribution and prioritizing the signal with the highest signal-to-noise ratio. The windows can be multiplied by the signals, reweighing each signal’s contribution (right column), and the sum provides a filtered signal. The edge effects remain on the boundaries due to a lack of information outside of the domain.
Figure 5.2. A schematic representation of the structure of the dense U-Net architecture. The transition-down blocks contain max-pooling layers, effectively reducing the spatial resolution of the feature maps and allowing the model to extract features at lower resolutions. The model can merge low- and high-level features across resolutions, leading to strong performances with fewer parameters. Reproduced from [49] with authorization from IEEE. © 2017 IEEE.
5.2 Multimodal image registration (Papers III and IV)

**CoMIR: Contrastive Multimodal Image Representation for Registration**

Rigid multimodal image registration is concerned with registering images originating from different sensors – e.g., brightfield with fluorescence microscopy images or MRI with PET images – so they are in the same coordinate system. The different images have matching structures, and their alignment increases how much information we can analyze – this process is named image fusion. Although classical algorithms – consisting of hand-crafted algorithms – perform well on images from the same modalities, they may struggle in a multimodal setting when the underlying structures have little in common.

The transformations applied to the images can be constrained to specific geometric transformations depending on the type of data. We focused on rigid registration, which includes rotation and translation. The transformations can be extended to include reflections and shearing in the case of affine registration. The field is often interested in deformable registration, where the transformation consists of a smooth vector field. More exotic transformations are also possible, for instance, when the lenses of a microscope produce chromatic aberrations and warp the images towards the edge, resulting in a circular deformation field.

Registration methods can be intensity-based, where the intensity of each pixel of the images is matched based on a distance function and uses all the information available. Conversely, point-based methods focus on sparse, discrete points of interest using a local feature detection method and aligning the resulting point clouds extracted from both images. Both of these methods have shortcomings, intensity-based methods being expensive, and the optimization method is likely to stop at a local maximum, while point-based methods – comparatively cheaper – can fail to register the images altogether when there are few noisy extracted points.

In Paper III, we hypothesized that deep learning methods would excel at generating a common representation of multimodal images that could be used by classical registration methods for aligning those representations. At the same time, neural networks have great difficulty dealing with geometric transformations and relating spatially distant features, leading to poor registration outcomes in an end-to-end learning-based method. In this context, we split the registration pipeline into two main components: first, a neural network generates a common representation from a set of multimodal images, then, proven classical methods perform the proper registration of the generated representations. The classical methods featured both intensity- and point-based

---

1 Many such papers assume that the images are already registered with a rigid transformation, often performed automatically with classical methods or manually, which makes multimodal rigid registration still relevant for research.

2 Nowadays, transformer-based methods could become viable contenders for end-to-end learning-based registration.
registration. Figure 5.3 is the graphical depiction of this idea as featured in Paper III, showing the full pipeline for multimodal image registration.

The method uses a contrastive loss named infoNCE [51] on two dense U-Net models. The models have different sets of weights as opposed to shared-weights, or a siamese network formulation as done usually done in contrastive [52]. The incentive originated from the belief that different modalities would require different parameterizations for extracting relevant features instead of a single model working with multiple modalities. The models were fed with augmented patches from the various modalities as anchors and positive examples. Meanwhile, negative examples were random patches originating from the rest of the dataset. In practice, as an optimization scheme, the negative examples were extracted from the batch necessary to perform gradient descent since each batch element is independent of the other. After training, the models start generating representations of the input images that can be registered. However, the models are not rotation-equivariant at this stage, such that a rotated image fed to the model will not result in the representation being rotated. This equivariance property is defined as

$$f_\theta(T(X)) = T(f_\theta(X)),$$

for a model $f$ parameterized with weights $\theta$, an input image $X$ and geometric transformation $T$, here a random $90^\circ$ rotation. Equation 5.1 was added to the contrastive loss where a random, $90^\circ$ rotation is chosen at every step, effectively enforcing rotation equivariance during inference. This correction was successful and provided rotation equivariance to the whole spectrum of possible angles instead of multiples of $90^\circ$. The video at https://youtu.be/iN5GlPWFZ_Q depicts the method with and without this correction, showing the variability – with the original contrastive loss – and stability – with the modified loss – of the features when rotated.

Global Parameter Optimization for Multimodal Biomedical Image Registration

While testing CoMIR against other classical methods, specifically mutual information (MI), Johan Öfverstedt (a co-author of Paper III) discovered a way to perform mutual-information-based registration efficiently in the Fourier domain using the fast Fourier transform [53]. This method finds the best translation parameters but necessitates a random search or evolutionary search to find the orientation. We proposed to improve on this method and use Bayesian optimization to find the orientation which leads to faster convergence, especially in 3D settings\(^3\), in Paper IV.

Classical methods for registration have two main components, one distance function between the images and an optimization algorithm. Poor registration\(^\footnote{For two-dimensional images, the rotation angle is a one-dimensional number in $[0, 2\pi]$, whereas the search space for the three-dimensional case is bigger, necessitating three- or four-dimensional objects, such as quaternions.}
Figure 5.3. Images originating from different microscopes (here, second-harmonic generation and bright-field microscopy) compose a challenging registration task. Two independent neural networks can be trained to project the images in a common representation called CoMIR. The representations can be registered with proven classical methods, and the final transformation can be applied to the original images, resulting in a successful multimodal registration.

outcomes can occur if one of those components fails; for instance, if the optimization fails to find the global maximum and converges to a local subpar maximum. Global optimization is a set of methods that will eventually find the global maximum [54], simplifying the registration task to finding distance functions. Admittedly, different global optimization techniques have different rates of convergence and computational costs, however, the success rate of registration becomes independent of these. We attempted to improve the original evolutionary search using bayesian optimization in this context.

A Gaussian process is fitted to a set of randomly sampled points along with their mutual-information-matching score. An acquisition function is then chosen – expected improvement in practice – to determine the next quaternion to explore. The kernel of the Gaussian process is based on the Matérn kernel that is warped to work on a noneuclidean geometry: the space of quaternions. We used the Matérn kernel on the d-dimensional sphere $S^d$ described in the appendix of [55]. However, computations of the Gaussian process and acquisition functions need to be performed on a regular grid, which is challenging. We used the homochoric representation of the quaternions described in [56] to project a regular three-dimensional cube into the space of quaternions, generating a regular grid of quaternions, approximately equally spaced. This projection is shown in Figure 5.4 where a cubochoric grid is projected to the angle-axis representation of the homochoric representation of the quaternions.
Figure 5.4. Unit quaternions are four-dimensional objects that can be used for representing the space of 3D rotation group SO(3). A regular grid representing the space of quaternions can be difficult to produce; one solution is to create a three-dimensional cube (the homochoric representation, left figure) and project it to the space of quaternions using a volume-preserving projection. The resulting unit quaternions can be displayed in three dimensions (right figure) using the angle-axis representation, where the distance of a quaternion to the center represents the angle, and the direction represents the rotation axis. This mapping is shown on a randomly chosen quaternion.
5.3 Methods for visualization of spatial transcriptomics experiments (Paper V)

Visualizing spatial transcriptomics data can be challenging due to the images being very large and spatially-resolved transcriptomics data containing a considerable amount of points to display simultaneously. TissUUmaps [57] is a web-based tool for visualizing and interacting with millions of markers overlayed on top of tissue images in real-time. TissUUmaps 3, the focus of Paper V is an improved version of the original TissUUmaps and is the fruit of the collaboration between many members of the Wählby lab. It features WebGL support for displaying large images and millions of markers in real time, which wouldn’t be possible with hardware acceleration along with new features, such as new types of markers (e.g., pie charts and graphs), plugin support and a more flexible, refreshed user interface. Figure 5.5 shows a screenshot of TissUUmaps running in the Firefox browser. As a webpage, the tool allows researchers to share the data produced in omics experiments easily. TissUUmaps uses GPU rendering to display images and markers, making navigation fluid on a wide range of computers. It also supports plugins to extend its capabilities and interaction with other software; I was involved in the development of one of them, napari-tissuuumaps, which allows TissUUmaps to import data originating from the image visualization and processing tool Napari [58].

Released as free, open-source software, a gallery, and tutorials are available at https://tissuuumaps.github.io/.
Figure 5.5. TissUUmaps is a web-based GPU-accelerated tool that helps researchers visualize tissues along with millions of spatial transcriptomics markers, assess the outcome of an experiment and perform quality control of image processing algorithms. The flexible sharing of datasets makes TissUUmaps an asset in the toolbox of researchers for disseminating the results of experiments.
6. Discussion and Future Work

In this thesis, we explored the themes of biology, math, and machine learning. These sections laid the basic components to understand the papers presented in chapter 5. Some of the methods have limitations in scope and additional research and new advances in machine learning could improve them. Here follow a few topics interesting to explore and potentially help build on the methods presented in the papers, particularly Papers II, III, and IV.

6.1 Unsupervised learning of periodic time-series

The dataset in Paper II contains time series of cells moving on an elastic substrate. The deep-learning model predicts the forces exerted by the cell by performing computations on unique frames, i.e., the temporal information is inexistent. This causes a lack of temporal stability, as shown in Figure 3 of the paper, where the predicted forces feature high-frequency noise over time. Thus, exploring methods accounting for the temporal dimensions is valuable for improving the accuracy and reducing the variance of the estimates of the method. Notably, [59] used a linear Gaussian state-space model to account for such temporal dependencies. The amount of data stemming from such analyses makes the creation of a ground truth prohibitively expensive, and focusing on unsupervised or semi-supervised learning methods becomes relevant. [60] proposed an unsupervised method when the targets have a periodic component and showed that unsupervised learning could be worth exploring even in supervised settings when a dataset has few samples, which is often the case in bioimage processing. The method also seems to produce better manifolds when interpolating and extrapolating to new samples (zero-shot generalization). Moreover, it would be relevant to incorporate uncertainty quantification in unsupervised learning so that points projected to the manifold (the feature space) become probabilistic. For example, if one wants to cluster the features of a dataset containing drugs, drug types that are rarely seen should have high uncertainty in the manifold.

6.2 Extensions to CoMIR

Rigid image registration (as opposed to deformable) using deep learning methods remains largely unexplored. The CoMIR method could benefit from being
extended to three-dimensional images for processing a broader range of image modalities, such as MRI/PET or 3D microscopy. Newer architecture, such as U-Net transformer [61] and newer contrastive loss functions, such as BYOL [62], could improve the method's overall performance in terms of convergence speed and better generalization performance to out-of-distribution images.

6.3 Decision-making algorithms

Paper IV improves the rate of convergence to the best orientation using Bayesian optimization. This is not always the most sensible option for black box estimation, especially when the function of interest is non-stationary. In this case, other methods from multi-armed bandit problems can be used when the reward entails temporal uncertainties [63]. Methods ranging from Bayesian statistics to reinforcement learning encompass varying complexity and computational costs, which could be better suited for faster convergence to the best parameters for the global registration method in Paper IV. For instance, tree-structured Parzen estimators [64] seem to be interesting candidates that could lead to faster convergence by reducing the amount of time spent the search of a next candidate and improve the overall registration speed.
7. Sammanfatning på svenska

Artificiella neurala nätverk är en samling algoritmer som inspirerats av den mänskliga hjärnan funktion. Liksom den mänskliga hjärnan lär sig dessa nätverk (genom någon form av träning) att känna igen och tolka olika mönster automatiskt. Artificiella neurala nätverk började bli beräkningsmässigt möjliga att använda i stor skala runt 2012, vilket också återspeglas i att allt fler artiklar om artificiell intelligens började publiceras då. Mer komplexa nätverk med många lager av så kallade neuroner kallas djupa neurala nätverk, och dessa metoder har visat sig kunna presterar betydligt bättre än många klassiska metoder i prestigebygda klassificeringsutmaningar ordnade i samband med till exempel bildanalyskonferenser.

När jag påbörjade min doktorandutbildning i början av år 2018 genomsyrade dessa djupinlärningsmetoder redan många olika områden, och intresset för användning inom biomedicinsk bildanalys hade just väckts. Metoderna har sedan dess fortsatt att utvecklas och en riktig teori har börjat formas. Denna avhandling är mitt försök att ge en introduktion till bakgrunden till detta tvärvetenskapliga område. Mer specifikt syftar den här avhandlingen till att ge en sammanfattning av de verktyg som behövs för att arbeta i gränslandet mellan biologi, datavetenskap och matematik. Jag hoppas kunna ge en intressant överblick av den tvärvetenskapliga forskning som jag har bedrivit under de senaste fem åren. Under den här tiden har jag lärt mig att intuition är ett viktigt drag hos en forskare, och jag har försökt att dela med mig av min egen genom hela det här arbetet. Dessutom har jag försökt jag bidra med en ny synvinkel, ibland genom att sätta olika idéer under samma paraply för att ge ett annat perspektiv på den forskning som redan publicerats, med förhoppning om att ge läsaren nya engagerande insikter.

bakgrund som behövs för att motivera valen av metoder och ge en beskrivning av strategierna som utgör grunden för artiklarna i kapitel 5. Slutligen sammanfattas resultaten och framtida forskning i kapitel 6.
8. Acknowledgements

I have really enjoyed my Ph.D. journey at Uppsala University, where I have had the opportunity to meet so many great people and friends. First and foremost, I would like to express my deepest appreciation and gratitude to my supervisor Carolina Wählby, for giving me the opportunity to pursue a Ph.D. Thank you for all the much-needed professional and personal advice, for always believing in me and guiding me so well throughout these five years that felt so short. Thank you for all your feedback and invaluable help with this thesis.

I am also extremely grateful to my co-supervisors Ida-Maria Sintorn and Anindya Gupta. Thank you, Ida-Maria, for all your support and cheerfulness and for your helpful comments on this thesis. And Anindya, thanks for the many fun and fruitful discussions we have had and for our collaboration on the Sysmic project.

A special thanks to my current colleagues and fellow Ph.D. students: Christophe Avenel, Eva Breznik, Axel Andersson, Ankit Gupta, Andrea Behanova, Eduard Chelebian Kocharyan, Fredrik Nysjö, Anna Klemm, Bedour Alshaigy, Johan Snider, Natalia Calvo Barajas, Deniz Bezek, Marc Fraile, Karl Bengtsson Bernander, Erik Hallström, Nadezhda Koriakina, Antti Ahola, Nikolaus Huber, and Tobias Mages. Thanks should also go to the colleagues that I had the good fortune to meet and work with, who graduated and moved on to new adventures: Håkan Wieslander, Leslie Solorzano, Teo Asplund, Jiahao Lu, Gabriele Partel, Giorgia Mili, Kimmo Kartasalo, Marielle Jansen, Philip Harrison, Manuel Haussmann, Amit Suveer, Kalyan Ram Ayyalasomayajula, Ektta Bats, and Thomas Wilkinson. Thanks to Ragnar Seton and Igor Tominec.

I am also thankful to Petter Ranefall and to all the professors, teachers, and researchers I have had the good fortune to meet and interact with at Uppsala University, Nataša Sladoje, Joakim Lindblad, Ewert Bengtsson, Ingela Nyström, Lars Oestreicher, Gunilla Borgefors, Christer Oscar Kiselman, Filip Malmberg, and Anders Hast.

A warm thank you to Juan Caicedo for welcoming me into your lab at the Broad Institute and giving me the opportunity to partake in your exciting research. Thank you, Juan and Salil Bhaté, for all our fruitful discussions and for your help during my internship. And to all of my other colleagues at the Broad Institute: I am so grateful that you made me feel right at home and helped me learn so much after a long time on parental leave. Many thanks to Rubén Fonnegra for all the fun discussions and activities in Boston. Thanks to Michael Doron, Sam Chen, Wolfgang Pernice, and Nikita Moshkov.

Thanks to Staffan Strömblad, Jianjiang Hu, Xavier Serra Picamal, and Veronica Larsson for collaborating on the Sysmic project. Jianjiang, special
thanks to you for your comments on the traction force microscopy section of this thesis.

So many thanks to my best-brainstorming partners and friends. Raphaela Heil, thank you for all the fun conversations, dinners, walks, and mushroom-picking afternoons spent in your company. Thanks to Elisabeth Wetzer for patiently explaining the mathematical theory and for all the awesome times spent in your company. You two have been the best office mates. I would also like to give a special thanks to Johan Öfverstedt for all our chats, coffee breaks, and deep dives into ways to improve the algorithms we built.

Thanks to my childhood friends, Vincent Rous, who was always interested and excited about my whereabouts, and Philipp Keil, who patiently listened to hour-long rants about the broken, obscure algorithms I was working on.

A big thanks to my parents, grandparents, and awesome brother Baptiste. Mom and dad, I cannot thank you enough for all the love and support you have given me in life.

And a special thanks to Li, my loving wife and favorite supporter who never ceased to believe in me and encouraged me to pursue a path in academia. Without you, this Ph.D. wouldn’t have been possible. And last but not least, thanks to our dearest son Émile, for giving me a break from my mathematical world, reminding me of what is most important in life, and being the best child I could have ever hoped for.
The cover of this thesis features the artwork "Have a ball!". The green background filled with creatures looking around might represent the biological world, with all its diversity and complexity. The various animals and creatures could symbolize the different personalities, attitudes, and perspectives that exist within society.

The woman in the foreground tossing out balls of energy could symbolize the sharing of ideas that we all possess, which can be harnessed to create positive change in the world. It could also represent the act of sharing research findings and contributing to the field’s collective knowledge. Just as the woman in the artwork is tossing balls of knowledge out into the world, researchers share their ideas and findings to advance our understanding of complex biological processes.

The five brains of the woman, mounted on top of one another, might represent the concept of intellectual growth and the idea that we can continually expand our knowledge and understanding of the world around us. The fact that the brains are connected to each other could symbolize the importance of collaboration and communication in achieving this growth. In the same way, deep learning methods possess stacked layers refining information at each level and helping researchers extract meaningful insights from complex biological systems.

Overall, the artwork seems to convey a message of creativity, diversity, knowledge and growth. The title of the work, "Have A Ball!", could be interpreted as an invitation to have fun and enjoy the process of learning and discovery. In the same way, despite the complexity of the subject matter, the thesis aims to make my research in bioimage processing fun and accessible to readers through clear explanations and visual aids.


— Written in collaboration with ChatGPT.
Bibliography


[23] Veritasium, “How imaginary numbers were invented.”


[29] 3Blue1Brown, “$e^{i\pi}$ in 3.14 minutes, using dynamics | de5.”


Acta Universitatis Upsaliensis

*Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 2244*

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology*. (Prior to January, 2005, the series was published under the title “Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology”.)

Distribution: publications.uu.se

urn:nbn:se:uu:diva-497143