TU WIEN Informatics

Classification of Tumor Cells in Childhood Cancer Using Automated Microscopy and Deep Learning

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Mag. Dr.rer.nat. Johannes Temme
Matrikelnummer 00306409

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Dipl.-Ing. Dr.techn. Martin Kampel
Mitwirkung: Dipl.-Ing. Dr.techn. Roxane Licandro
Mag. Dr.rer.nat. Sabine Taschner-Mandl

Wien, 8. Februar 2023

Johannes Temme | Martin Kampel

TU WIEN Informatics

Classification of Tumor Cells in Childhood Cancer Using Automated Microscopy and Deep Learning

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Mag. Dr.rer.nat. Johannes Temme
Registration Number 00306409

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Dipl.-Ing. Dr.techn. Martin Kampel
Assistance: Dipl.-Ing. Dr.techn. Roxane Licandro
Mag. Dr.rer.nat. Sabine Taschner-Mandl

Vienna, 8th February, 2023

Johannes Temme | Martin Kampel

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Mag. Dr.rer.nat. Johannes Temme

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 8. Februar 2023

Johannes Temme

Danksagung

Nach dem Beenden dieser Masterarbeit gibt es viele Menschen, denen ich zu großem Dank verpflichtet bin. Zuallererst möchte ich Sabine für ihre Betreuung dieser Arbeit danken. Sabine hatte stets Zeit für meine Frage und Anliegen und ich danke ihr für ihr Talent, komplexe Sachverhalte der Biologie und Medizin auch für Laien wie mich verständlich zu erklären. Ich danke Sabine auch für ihre wertvollen Änderungsvorschläge und Ideen zu dieser Arbeit.

Ich danke Prof. Martin Kampel und Roxane für die Betreuung auf Seiten der TU Wien. Besonders möchte ich Roxane für das sorgfältige Lesen dieser Arbeit und ihre Änderungsvorschläge danken. Ebenso danke ich ihr für ihre Tipps und Ideen, die sie in diese Arbeit einfließen hat lassen, insbesondere für ihre Anregung, pyradiomics Feature zu verwenden. Prof. Kampel möchte ich für seine stets raschen Rückmeldungen und die problemlose Bewältigung organisatorischer Herausforderungen, die beim Schreiben einer Masterarbeit entstehen, danken.

Ich danke den Kolleginnen und Kollegen in der Tumorbiologiegruppe am CCRI für den fachlichen und kollegialen Austausch. Insbesondere danke ich dabei Marie Bernkopf, Florian Kromp, Eva Bozsaky, Simon Gutwein, Daria Lazić und Fikret Rifatbegović für ihre Ideen, Anmerkungen, Hilfen und Beiträge zu dieser Arbeit. Christiane Paukner danke ich für ihre Bachelorarbeit, die das Bildmaterial für diese Arbeit geliefert hat. Lorenz Riess, Mathias Beiglböck und Julio Backhoff danke ich, dass sie mich mit der Theorie zu Wassersteindistanzen vertraut gemacht haben.

Zuletzt möchte ich mich bei meiner Frau Ada und meiner Tochter Maja für ihre Unterstützung und Geduld bedanken, dass ich ein erneutes Studium beginnen und jetzt diese Arbeit schreiben konnte. Ich bin ungemein dankbar euch zu haben, und dass wir in ein paar Monaten mit unserem kleinen Felix zu viert sein werden.

Kurzfassung

Alternative Telomerverlängerung (ALT) ist ein Mechanismus in Krebszellen, der Telomerverkürzung unterbindet, die bei der Teilung normaler Zellen stattfinden würde. Bei Neuroblastomen, einer Krebserkrankung bei Kindern und Jugendlichen, ist ein positiver ALT Status prognostisch für einen ungünstigen Verlauf. Mit Hilfe der Telomer PNA (telPNA) Fluoreszenz-in-situ-Hybridisierung werden Telomere in Mikroskopiebildern als helle punktförmige Signale dargestellt. Längere Telomere weisen dabei hellere und größere Punkte (ultrahelle Punkte) in den telPNA Bildern auf. Klinisch wird der ALT Status derzeit vor allem expertenbasiert und mit Faustregeln visuell über die ultrahellen Punkte in den Bildern bestimmt. Es existieren bisher keine bildbasierten Ansätze, die den ALT Status anhand automatisierter und eindeutiger Regeln mit hoher Konfidenz und auch für ALT-positive Zellen, die keine ultrahellen Punkte aufweisen, bestimmen.

Unter der Verwendung von Mikroskopiebildern der St. Anna Kinderkrebsforschung (Wien) werden in dieser Masterarbeit verschiedene Klassifikationsmodelle zweier Ansätze zur Vorhersage des ALT Status verglichen. Im ersten Ansatz diskutiert die Arbeit die manuelle Erzeugung von Prädiktoren, um sie in Klassifikationsmodellen zu verwenden. Die Arbeit führt dafür mit dem sogenannten Wassersteindistanzmodell auch ein neues Modell ein, das auf Wasserstein-Metriken zwischen Prädiktorverteilungen beruht. Im zweiten Ansatz werden Deep Learning Methoden verwendet, in denen die Bilder direkt genutzt werden, um Prädiktoren automatisch zu finden und die Zellen zu klassifizieren.

Die Masterarbeit behandelt zwei zentrale Fragen. Die erste Frage klärt, welches Modell der zwei Ansätze den ALT Status von Zellen am Besten voraussagt. Die zweite Frage erörtert, welche bildbasierten Prädiktoren am Ehesten geeignet sind, den ALT Status von Zellen zu bestimmen. In dieser Arbeit beantworten wir die Fragen, indem wir zeigen, dass das Wassersteindistanzmodell die mit Abstand genauesten Ergebnisse bei der Vorhersage des ALT Status liefert. Mit Ausnahme des Wassersteindistanzmodells übertreffen Deep Learning Methoden Ansätze der manuellen Prädiktorerzeugung. Zudem weist die Masterarbeit nach, dass Prädiktoren, die auf der Größe von Telomerpunkten, auf visuellen telPNA Clustern und der Schiefe und Wölbung der telPNA Intensitäten basieren, sehr gut für die Vorhersage des ALT Status von Zellen geeignet sind. Auch wenn weitere Forschung notwendig ist, um unsere Erkenntnisse für zusätzliche Zellinien und Gewebeschnitte zu bestätigen, zeigen die Resultate, dass Computer-gestützte Diagnoseverfahren zur Unterstützung von Experten bei ALT-Klassifizierungen möglich sind.

Abstract

Alternative lengthening of telomeres (ALT) is a mechanism in cancer cells that stops telomere shortening, which would accompany proliferation of normal cells. Diagnosing whether the cell's chromosomes are ALT+ or not (i.e. ALT−) is a key determinant for a poor outcome in childhood neuroblastoma tumors. Using telomere PNA (telPNA) fluorescence in-situ-hybridisation, telomeres can be visualised as spots in microscopy images. Longer telomeres exhibit brighter and more pronounced (so-called ultra-bright) spots. Currently, clinical staff determines the ALT status of cells in telPNA images based on expert judgement or rules-of-thumb on visible ultra-bright spots. To date, there are no image-derived, clear-cut objective or automated rules for determining the ALT status with high confidence, especially for ALT+ cells that do not show ultra-bright spots.

Using microscopy imaging of the Children's Cancer Research Institute (Vienna, Austria), in this master's thesis different classification models for the prediction of the cells' ALT status are designed and evaluated following two streamlines: (1) image feature-based classification approaches (2) image-based classification approaches. In the first approach, the master's thesis discusses feature generation that shall serve as explanatory variables for classification models (such as logistic regressions) to predict the ALT status. To that end, the master's thesis also first introduces a so-called Wasserstein distance model to classify the ALT status using Wasserstein distances between distributions of explanatory variables. In the second approach, the master's thesis uses deep learning to classify cells using microscopy images as direct inputs.

This master's thesis addresses two main research questions: first, we are interested which model of the two approaches predicts the ALT status of cells best. Second, we want to find image-derived features of both approaches that are best suited for determining the ALT status. We answer theses questions by showing that the Wasserstein distance model provides by far the best results when predicting the ALT status of cells. Apart from the Wasserstein distance model, image-based classification approaches outperform image feature-based approaches. Furthermore, we find that features that build on spot sizes, the presence of clusters as well as the skewness and kurtosis of telPNA intensities are best suited for predicting the ALT status. While further research is necessary to foster our findings on additional cell lines and tissues, the results show that computer-aided diagnostics of ALT is feasible and may support experts when predicting the ALT status.

Contents

CHAPTER 1

Introduction

This chapter introduces the topic of this master's thesis by providing in Section 1.1 a short overview of the medical background and discussing in Section 1.2 the aim of this thesis. In Section 1.4 we treat related work and highlight the contributions of this thesis. In Section 1.6, we summarise the results. Last, in Section 1.7, we give an overview of how this thesis is structured.

1.1 Problem Statement

A human cell consists of a cell membrane, cytoplasm and multiple separate parts (so called organelles) such as the cell nucleus. Within the nucleus, the genetic material in humans is organised in 46 chromosomes and telomere are the chromosome ends. The genetic code determines which proteins to produce that build many of the structures in a cell. The so-called alternative lengthening of telomeres (ALT) is a mechanism in cancer cells that stops telomere shortening (telomere maintenance mechanism), which would accompany proliferation of normal cells [BEDP$^+$97]. Diagnosing whether tumor cells use the ALT pathway (ALT+) or not (ALT−) is a key determinant for a poor outcome in childhood neuroblastoma tumors [PPG$^+$15, MWT$^+$21], even though ALT+ cells usually grow slowly compared to other tumor cells [HSNH$^+$21]. More specifically, prognostically favorable low-risk neuroblastoma tumors lack telomere maintenance mechanisms, whereas prognostically unfavorable intermediate-risk and high-risk tumors feature telomere maintenance mechanisms (partly in combination with specific genetic mutations) [ACH$^+$18]. Neuroblastoma is a tumor of the sympathetic nervous system, which represents the most common solid pediatric tumor outside the cranium and the most frequently diagnosed cancer in infants [MWT$^+$21, HGH$^+$96, Mar10]. Around 50% of neuroblastoma patients have a dismal outcome despite intensive treatment [ACH$^+$18, RSW$^+$19].

By using telomere peptide nucleic acid (telPNA) Fluorescence In-Situ Hybridisation (FISH), one can visualise the chromosomes' telomeres in microscopy images. Telomeres

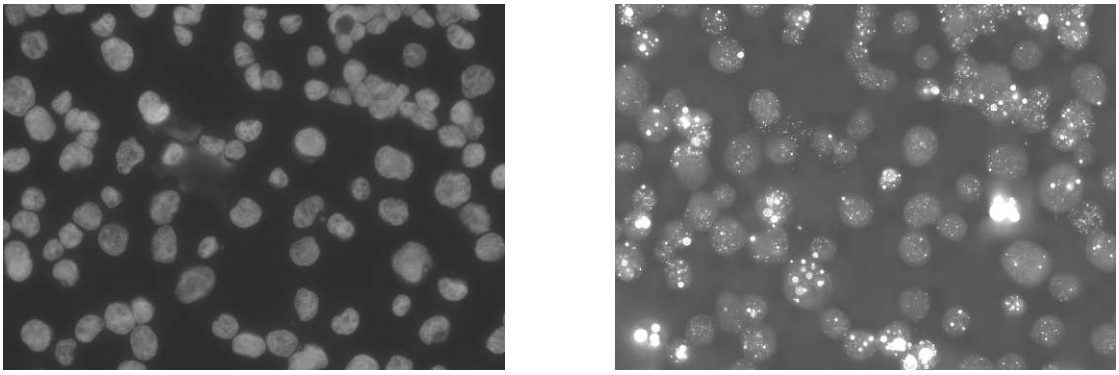


Figure 1.1: DAPI channel (left) and telPNA channel (right) of an ALT+ cell line (SK-N-MM). The microscopy images are part of the dilution series P11, see Section 2.3.1. We note pronounced ultra-bright spots in the telPNA channel but also that not all ALT+ nuclei feature these spots.

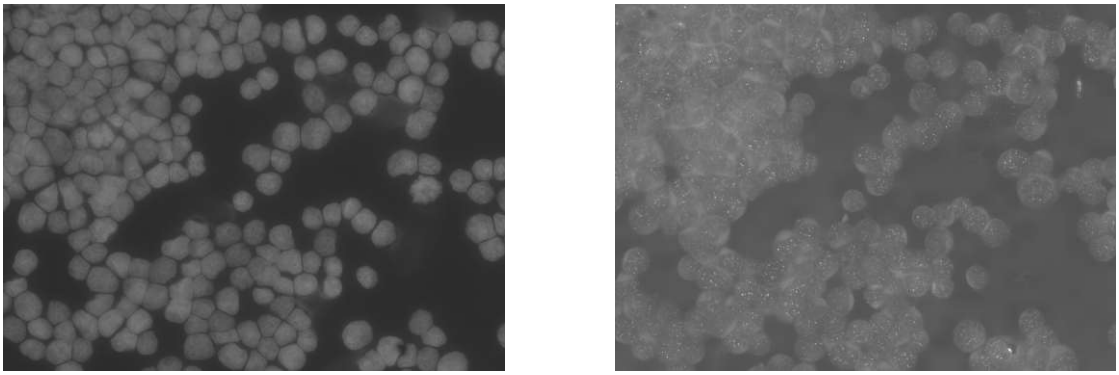


Figure 1.2: DAPI channel (left) and telPNA channel (right) of an ALT−cell line (SK-N-SH). The microscopy images are part of the dilution series P9WH, see Section 2.3.1. The telomere spots are considerably less pronounced than for the ALT+ cells of Figure 1.1.

show up as bright spots of intra-nuclear telomere foci in the red telPNA channel of the microscopy images, while nuclei are visible in the blue DAPI channel, see Figures 1.1 and 1.2. Longer telomeres will lead to brighter and more pronounced spots in the telPNA channel (so-called *ultra-bright* spots). However, not all ALT+ cells usually show such ultra-bright spots in the telPNA channel of microscopy images. Currently, clinical staff determines the ALT status of cells in microscopy images based on expert judgement. Although the literature proposes certain rules-of-thumb to determine the ALT status of cells manually in microscopy images [HSH+11], there are no clear-cut objective or automated rules yet for determining the ALT status with high confidence.

The tumor biology group at the Children's Cancer Research Institute (CCRI) in Vienna (Austria) prepared four neuroblastoma cell cultures of different cell lines, two with ALT+ and two with ALT− status [Pau21]. Here, cell lines refer to cells that originate from

one cell via cell divisions and cell cultures are cultivated cell lines for microscopy images. The group prepared several dilutions series among these cell cultures to provide samples with eight different ratios of ALT+ and ALT− cells. The ALT+ ratios cover 100%, 75%, 50%, 25%, 10%, 5%, 1% and 0%. For each of these samples, microscopy image series of the diluted ALT+ and ALT− cell cultures via FISH exist. To that end, the group cytospinned the cell cultures to microscopy carrier glasses before generating the FISH microscopy images. Furthermore, the group implemented deep neural networks (based on Cellpose [SWMP21], Mask-R-CNN [HGDG17] and U-Net ResNet [EVC$^+$19]) to identify and segment individual nuclei and their corresponding telomere spots in each pair of the DAPI and telPNA images [KFB$^+$21]. Chapter 2 provides further information on neuroblastoma, ALT and fluorescence microscopy.

1.2 Research Questions of this Master's Thesis

Using the microscopy images of the CCRI tumor biology group, a principal aim of the master's thesis is to identify suitable automatic classification approaches that predict the ALT status of neuroblastoma cells. There are two kinds of classification problems we are interested in: first, predicting the ALT status of an individual cell (ALT classification on nucleus level), and, second, predicting the above-mentioned ALT+ ratio of a given dilution series (ALT classification on series level). Models that solve the first classification problem can also be used for the second classification problem.

In this master's thesis we aim at answering two main research questions. The first question is about finding an optimal method out of a selection of models to predict the ALT status on nucleus or series level. This selection includes on the one hand models of the so-called *feature generation approach* (FGA), which requires manual feature generation to provide explanatory variables for classification models such as random forests, support vector machines, (penalised) logistic regressions and extreme gradient boosting to predict the ALT status on nucleus level. We also introduce a so-called Wasserstein distance model that predicts the ALT+ rate on series level by using Wasserstein distances between feature distributions. On the other hand, we use models of the so-called *image-based approach* (IBA), for which we use microscopy images as direct inputs to our own implementation of a convolutional neural network (CNN, called MyNet) as well as to a fine-tuned ResNet-50 to identify the cells' ALT status on nucleus level. Comparing these two approaches is popular in computer-aided diagnostics to assess whether self-learned features in image-based approaches are more predictive than hand-crafted features of feature generation approaches [XXS$^+$17].

The second research question addresses features and criteria for determining the ALT status of cells. While the rules of thumb of [HSH$^+$11] focus on whether so-called ultra-bright spots are present in the telPNA channel, or not, the FGA and IBA intend to provide more objective rules that determine the ALT status with higher confidence. In particular, the FGA and IBA shall both find image-derived criteria that describe the ALT+ status even for cells that do not show ultra-bright foci. Such criteria may also

help explaining why the models arrived at specific predictions and may also foster the users' confidence in the models' decisions.

1.3 Challenges

While the microscopy imaging protocol is standardised for all microscopy image series of the CCRI, the quality of microscopy images still vary across the series. More specifically, we can expect a higher variance in the number and position of nuclei, the appearing size of nuclei and imaging quality across the series. This implies two main technical challenges: first, nuclei might be inaccurately segmented due to overlapping cells. For example, this can happen if the cytospin preparation process led to more sticky cells. Second, image quality and fluorescence staining of the FISH might vary across the image series. The imaging quality depends on how the nuclei are attached to the carrier glass, as some images may show air bubbles or blurry content as artefacts. Furthermore, the immunofluorescence staining may have led to unwanted imaging effects such as bright stripes around the nucleus border caused by unspecific cytoplasmic staining.

Both the FGA and IBA have to address the afore-mentioned two technical challenges to correctly answer the research questions. To that end, we implement a separate post-processing model to detect inaccurately segmented nuclei based on geometric properties of the segmentation masks (e.g. size, convexity). Furthermore, we use methods that aim at generating and finding features that are *stable* in the sense that they are less affected by varying image quality. For the FGA, we pick these features via a particular variable selection algorithm that uses Wasserstein distances to identify stable features. For the IBA, we use specific data augmentation techniques to robustify the feature extraction.

1.4 Related Work

Related to our idea of predicting the ALT status of cells, the literature knows four other main approaches: First, one can use the above-mentioned rules of thumb of [HSH+11]. Second, instead of using microscopy images, one can use Whole Genome Sequencing to predict the ALT status with notable success [LTH+18]. Third, one can use the so-called C-circle assay, which is a polymerase chain reaction assay that makes use of the fact that telomer elongation is involving circular intermediates in ALT+ cells [HCH+09, HR10]. Fourth, one can refine fluorescence microscopy imaging to suppress telomere signals for ALT− cells [FRM+22]. Despite these four approaches in the literature, there are no automated rules for determining the ALT status with sufficient confidence for the fluorescence microscopy data of this master's thesis. In particular, to the best of our knowledge, there is also no IBA in the literature to predict the ALT status of cells.

To build the own network MyNet for the IBA, we therefore follow the literature of classifying so-called human elliptical 2 (HEp-2) cells, a human epidermoid carcinoma cell line, with CNNs based on indirect immunofluorescence microscopy images, which has been a very active research topic [RWSZ20]. HEp-2 cells are relevant for identifying

antibodies in blood serum via their fluorescence patterns. The patterns partly exhibit bright spots which vary in shape, position and density within the cells [oAp]. For that reason, we build our own network based on a variation of LeNet-5, that has proven to be well suitable for HEp-2 classification [GWZZ16, RNM17, RNM20].

1.5 Contributions

By leveraging insights from the literature on ALT and HEp-2 classification, I contribute the following five insights with this master's thesis: first, I introduce in this thesis Wasserstein distance models in the context of image classification and also provide algorithms to select suitable variables for the models. Second, I propose a novel CNN approach, called MyNet, to classify the ALT status on nucleus level. Third, I introduce special data augmentation techniques to mimic certain fluorescent staining patterns. Fourth, I contribute a new algorithm to select FGA features with Wasserstein distances that are stable in the sense that they behave similarly on related data sources of potentially different image quality. Fifth, I provide a list of features that predict the ALT status in the FGA best.

1.6 Results

The master's thesis shows that the Wasserstein distance model predicts the ALT+ ratio on series level by far the best. Furthermore, we find that the IBA and, most notably, ResNet-50 [HZRS16] outperforms the FGA models on nucleus level (i.e. random forests, support vector machines, penalised logistic regression, gradient boosting). The most predictive features for ALT classification refer to clusters as well as the skewness and kurtosis of the intensity distribution in the telPNA channel, or consider spot sizes. Furthermore, MyNet appears to extract image properties that are similar to these FGA features.

1.7 Thesis Outline

The thesis is structured as follows: in Chapter 2, we give further details on the medical background of neuroblastoma, ALT, fluorescence microscopy and provide information on the available deep neural networks of the CCRI to segment nuclei and spots in microscopy images. We also introduce the available data of this master's thesis and discuss statistics and potential technical challenges when working with this data. In Chapters 3 and 4, we give detailed information about state of the art approaches of FGA and IBA models for microscopy image classification, respectively. Furthermore, we provide data scientific background about how we split the available data into training and testing samples for the FGA and IBA. Chapter 5 outlines the state of the art for ALT classification and classifying fluorescence patterns. Afterwards, Chapter 6 describes our methodology that we want to apply based on the state of the art. In particular, we introduce the Wasserstein

distance model and algorithms to build it via variable selection. After having introduced the samples and FGA and IBA models that we use for this thesis, Chapter 6 also includes a dedicated section that formulates our two research questions at large. In Chapters 7 and 8 we provide thorough reasoning on how we setup further methodological details of the samples and models of the FGA and IBA, respectively. In Chapter 9, we present the results of this master's thesis and answer our research questions. Last, in Chapter 10, we summarise our findings in view of the two research questions. Furthermore, we reflect on how the models may support experts in diagnosing the ALT status for clinical reports and we discuss potential areas of research for future work.

CHAPTER 2

Medical Background and Data

This master's thesis uses microscopy images of neuroblastoma cell lines. Neuroblastoma is a common pediatric tumor and its ALT status is an important risk factor. This chapter provides background information on neuroblastoma and the telomere maintenance mechanism ALT (Section 2.1 and 2.2). Furthermore, it discusses microscopy imaging techniques and gives details about the microscopy image series that we use in this master's thesis (Section 2.3).

2.1 Neuroblastoma

Mutations are changes in deoxyribonucleic acid (DNA) sequences of a cell. They can happen due to various reasons such as copying errors during cell division or exposure to ionizing electromagnetic radiation. Usually, mutations in normal cells are restored or lead to programmed cell death (apoptosis). However, cell mutations may also result in tumors, which are abnormally and excessively growing tissues. Tumors can be benign or malignant and in the latter case they are referred to as cancer [Bun08].

This master's thesis uses microscopy images of neuroblastoma tumor cells. Neuroblastoma is a tumor of the sympathetic nervous system, which represents the most common solid pediatric tumor outside the cranium and the most frequently diagnosed cancer in infants [MWT+21, HGH+96, Mar10]. Around 50% of neuroblastoma patients have a dismal outcome despite intensive treatment [ACH+18, RSW+19]. Tumor stage, age at diagnosis, histology, and an amplified MYCN gene, which is important for cell growth, are important factors for unfavorable tumor outcome [PPG+15]. The so-called ALT pathway prevents telomeres from shortening upon cell divisions and represents another important risk factor for neuroblastoma tumors [PPG+15].

2.2 Alternative Lengthening of Telomeres - ALT

Telomeres are nucleoprotein complexes at the end of eukaryotic chromosomes, which consist of the repetitive DNA sequence TTAGGG. Telomeres usually undergo progressive shortening during cell division, see Figure 2.1. If they are falling short a critical length, the cell undergoes senescence or apoptosis, which prevents further cellular proliferation [DRCF+03]. For that reason, telomere shortening accompanies the cellular aging process and inhibits unlimited growth.

There are two known mechanisms that allow certain cancer cells to circumvent telomere shortening and to proliferate unlimitedly: first, they can activate telomerase, or, second, they employ a telomere maintenance mechanism known as ALT. This second mechanism works by replicating telomeric DNA using homologous recombination of DNA [BEG+95, BEDP+97]. More than 85% of all human tumors use the first mechanism, while only a lower number of tumors use ALT [OD12]. Most notably, 59% of neuroblastoma employ the ALT maintenance mechanism in a study on neuroblastoma of multiple stages and risk groups including both relapses and cases at diagnosis [PPG+15]. While determining the prognosis of patients with ALT positive (ALT+) tumors depends in general on the tumor entity, ALT has proven to be a key determinant for a poor outcome in neuroblastoma [PPG+15, MWT+21]. More specifically, low-risk neuroblastoma tumors lack telomere maintenance mechanisms, whereas intermediate-risk and high-risk tumors feature telomere maintenance mechanisms (partly in combination with specific genetic mutations) [ACH+18].

In routine, telPNA FISH in combination with fluorescence microscopy is used to diagnose the ALT status of tumors. telPNA FISH fluorescence microscopy allows assessing the telomere length of individual chromosomes [PBH+03]. To that end, FISH employs DNA oligonucleotide probes conjugated to a fluorescence dye which bind to specific nucleic acid sequences to assess whether these DNA sequences on chromosomes are present or not. As the fluorescence signal detected is considered proportional to the length of the telomere, bigger telPNA spots will indicate longer telomere. Here, it is worth noting that up to 92 telomeres are identifiable in a healthy human cell. This is because the number of telomeres corresponds to the number of chromosomes in a cell and would therefore equate to $92 = 46 \cdot 2$, as there are 22 chromosomes and one sex chromosome with each two ends.

In telPNA FISH microscopy images of ALT+ tumors, the distribution of telomere lengths within the nucleus and between tumor cell populations is highly heterogeneous [BEG+95]. One also assesses the ALT status by detecting cells that present ultra-bright intra-nuclear telomere foci or spots. This characteristic pattern correlates with an ALT+ status and [HSH+11] proposes a rule-of-thumb that considers tumors containing more than 1% of such cells as ALT+ (see also Section 5.2). Still, when such ultra-bright spots are absent, diagnosing the ALT status remains challenging as the proposed criteria are not applicable.

While the principles of [HSH+11] are useful indications when ultra-bright telomere spots are present, they are also not clear-cut and objective automated rules to determine

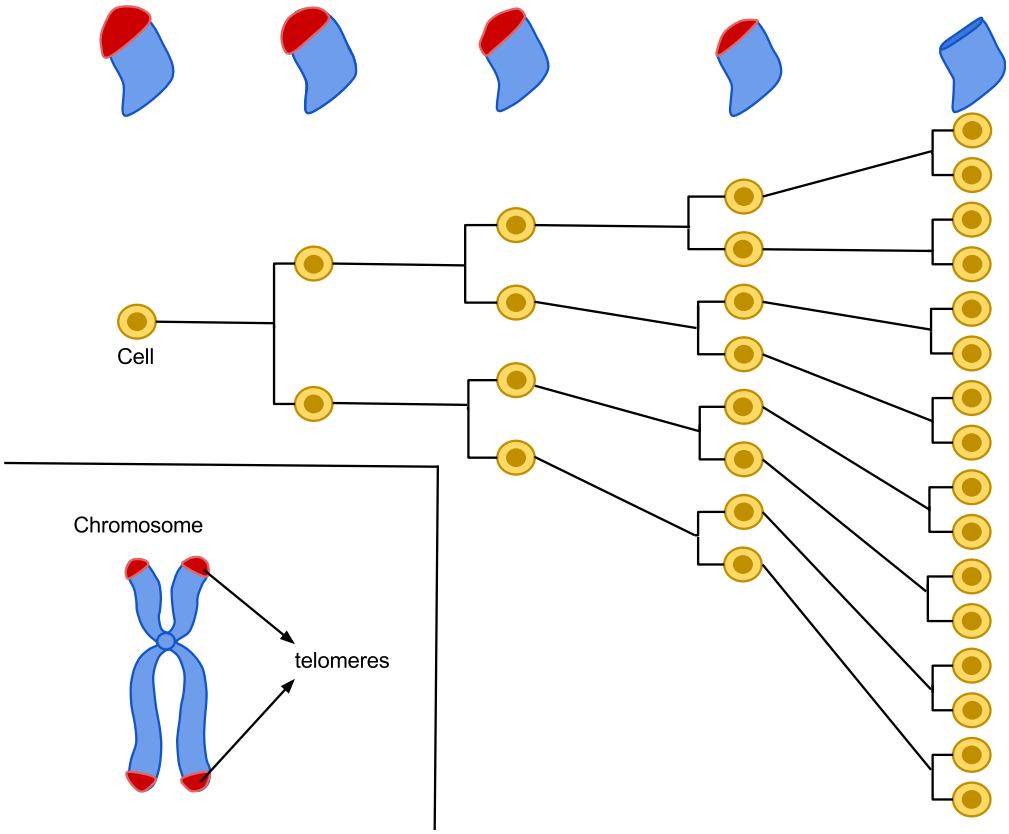


Figure 2.1: Shortening telomeres after cell division.[a]

[a]Image taken unchanged from Wikimedia Commons https://commons.wikimedia.org/wiki/File:Hayflick_Limit_%281%29.svg. Accessed: 2023-01-24.

the ALT status with high confidence. When ultra-bright telomeric spots are absent, experts still describe the telPNA FISH staining pattern of ALT+ cells to feature a more heterogeneous telomeric length distribution as compared to ALT− cells. Furthermore, ALT+ cells show more variation in the telPNA signal distance within cells.

2.3 Microscopy Images and Data

In this section and the following subsection we discuss microscopy imaging techniques and the data used in this master's thesis. For the microscopy imaging techniques, we outline fluorescence microscopy. Furthermore, we discuss the recording setup and definitions dilution series that we use in this thesis.

2.3.1 Microscopy Image Data of this Master's Thesis

Microscopy provides major insights into cells by visualising their cellular and sub-cellular structures. Automated microscopes allow for generating images and quantitative results for thousands of cells in a relatively short period of time. Such an automated analytical workflow makes it possible to detect even small cellular variations and alterations and, at the same time, to give statistical evidence by analysing a high number of cells [HM04].

The CCRI uses immunofluorescence stainings in microscopy images to visualise cellular and subcellular structures at the same time via multiplex-immunofluorescence staining techniques (see Section 2.2). Apart from tissue segments or sections or slices, the CCRI also uses cell lines grown on or cytospinned to microscopy carrier glass slides. Here, cytospins refer to specialised centrifuges that can attach cells to microscopy carrier slides.

For this master's thesis, we use microscopy images that were prepared at the CCRI in a bachelor thesis' project [Pau21]. These images are based on four neuroblastoma cell lines that were selected due to their ALT status (two ALT+, two ALT− cell lines) and specific molecular markers that help distinguishing cells in fluorescence microscopy. The two ALT+ cell lines are labeled CHLA90 and SK-N-MM, while the two ALT− cell lines are named SK-N-SH and CLB-MA. For [Pau21], the group at CCRI prepared several serial dilutions among these four cell lines to provide samples with eight different ratios of ALT+ and ALT− cells:

1. 100% to 0% (only ALT+)
2. 75% to 25% (medium-high ALT+)
3. 50% to 50% (equal ALT+ and ALT−)
4. 25% to 75% (medium-low ALT+)
5. 10% to 90% (low ALT+)
6. 5% to 95% (very low ALT+)
7. 1% to 99% (rare ALT+)
8. 0% to 100% (no ALT+, only ALT−)

For each sample, the cell lines were cultivated and afterwards cytospinned to microscopy carrier glass slides to generate immunofluorescence images. Each dilution is based on at most one ALT+ cell line and at most one ALT− cell line and received a specific coding name (such as "ALT-C.P6"), to which we will refer throughout this thesis as a *dilution series*. Table 2.1 states all dilution series that we consider in this thesis and denotes their coding name, the ALT+ and ALT− cell lines as well as the corresponding ratios of ALT+ and ALT− cells. To simplify the following discussion, we denote dilution series of 100% ALT+ cells and of 100% ALT− cells as *pure* dilution series and refer to all other

dilution series name	ALT+ cell line	ALT− cell line	%ALT+	%ALT−
ALT-C.P3∼A	SK-N-MM	-	100	0
ALT-C.P4∼A	-	SK-N-SH	0	100
ALT-C.P6	CHLA90	-	100	0
ALT-C.P7	-	CLB-MA	0	100
ALT-C.P8	SK-N-MM	-	100	0
ALT-C.P9WH	-	SK-N-SH	0	100
ALT-C.P10	CHLA90	-	100	0
ALT-C.P11	SK-N-MM	-	100	0
ALT-C.P12	-	SK-N-SH	0	100
ALT-C.P13	-	CLB-MA	0	100
ALT-C.PM1	SK-N-MM	SK-N-SH	1	99
ALT-C.PM3	SK-N-MM	SK-N-SH	10	90
ALT-C.PM4	SK-N-MM	SK-N-SH	50	50
ALT-C.PM5	SK-N-MM	SK-N-SH	75	25
ALT-C.PM9	CHLA90	CLB-MA	50	50
ALT-C.PM12	CHLA90	CLB-MA	25	75
ALT-C.PM14	CHLA90	CLB-MA	5	95
ALT-C.PM15	CHLA90	CLB-MA	1	99
ALT-C.PM22	SK-N-MM	SK-N-SH	1	99
ALT-C.PM23	SK-N-MM	SK-N-SH	5	95
ALT-C.PM24	SK-N-MM	SK-N-SH	25	75

Table 2.1: Dilution series of [Pau21] considered in this master's thesis. The horizontal line in the middle of the table separates pure dilution series (above) from actual dilution series (below).

dilution series as *actual* dilution series. We can tell pure from actual dilution series by their coding name, since the coding name of actual dilution series include the letter "M" (m for *m*ixture, such as "ALT-C.P**M**9"). In the following, we sometimes avoid the prefix "ALT-C." and only refer to the unique alphanumeric suffix of the coding name (such as "P6" or "PM9") to alleviate notation.

2.3.2 Recording Setup of the Microscopy Image Data

All samples of the dilution series were scanned with fixed exposure times and autofocus using an AxioImager-Z2 microscopy from Carl Zeiss equipped with a Plan-Apochromat lens with 63x magnification [Pau21]. For each sample, we have two channels of the immunofluorescence images available: first, a (blue) so-called DAPI channel (0.00373s exposure time) and, second, a (red) telPNA FISH staining channel (0.16s exposure time). While the blue DAPI channel allows us to identify the cell nuclei, the telomeres show up as bright spots of intra-nuclear telomere spots in the red telPNA channel of the microscopy images as described in Section 2.2. Section 6.2 gives further details on how

to automatically identify nuclei and spots in the images.

Theoretically, for each channel there is a separate digitised microscopy image available. However, due to memory reasons, these channel images are split into much smaller 100-1,500 pairs of blue and red channel images (image patches or "fields of view") which sequentially cover the original channel images. The 100-1,500 pairs of blue and red channel image patches capture each dozens of nuclei and telomere spots. Figures 1.2 and 1.1 in Section 1.1 show one field of view of the channels for two different dilution series.

An appliance of MetaSystems Hard & Software GmbH allowed for automatically scanning the microscopy images. The appliance consisted of a Metafer CoolCube4m camera and a scanning system called MetaCyte (program version 4.3.120). The corresponding software of MetaSystems consistently processed the images when digitising them [Pau21]. For that reason, we can assume that the imaging conditions are identical for all dilution series. Still, the image signals for different dilution series of the same cell lines and dilution ratios may differ because of technical reasons when preparing the carrier slides. For example, the immunofluorescence staining could attach differently across the dilution series. Section 6.3 below discusses the varying imaging quality in detail.

2.4 Summary

In this chapter, we have discussed the medical background for neuroblastoma and ALT. We provided information on risk drivers of neuroblastoma and diagnostics for ALT, based on microscopy images. We learned that information of the telPNA channel is important for classifying the ALT status. Furthermore, we introduced the microscopy image data for this master's thesis by giving details on the series names, cell lines and recording setup.

CHAPTER 3

State of the Art: Machine Learning for Microscopy Images

Classifying microscopy images such as the nuclear images of Section 2.3 is a so-called *image classification problem*. More specifically, we are in the setup of supervised learning in which we have images with labelled targets (such as the ALT status) available to train and validate potential classification models.

Broadly speaking there are two main approaches to solve image classification problems: the feature generation approach (FGA) and the image-based approach (IBA). In the FGA, one first (manually) generates features for each image that one wants to classify and then trains classification models based on these features and labelled targets of each image. We refer to this approach also as *machine learning for microscopy images*. Conversely, so-called *deep learning models* of the IBA generate features automatically on their own during training by taking the image as direct input.

In this chapter, we start off in Section 3.1 by summarising the main notation of this chapter that we will also use throughout this thesis. Afterwards, Section 3.2 provides details on the data scientific setup of training, testing and validation samples for model development. Section 3.3 discusses the state of the art for generating features of microscopy images via radiomics. Finally, in Section 3.4, we discuss the state of the art for the FGA when classifying microscopy images based on previously extracted features.

3.1 Denotation

In this thesis, we denote an microscopy image by I and a collection of images as $(I_k)_{k=1}^K$ with corresponding labels $(g_k)_{k=1}^K \subseteq \mathcal{G}$. We refer to $\mathcal{I}$ as the set of all microscopy images. We define by $\mathcal{T}, \mathcal{V}, \mathcal{T}_{test}$ training, validation and testing samples as discussed in Section 3.2. For each image I_k, we denote the corresponding feature vector by $x_k \in \mathbb{R}^p$ and $p \in \mathbb{N}$

refers to the feature dimension. In models that use feature weights, we denote these weights by $\beta = (\beta_i)_i$.

3.2 State of the Art Setup to Train, Validate and Test Models

We assume that we are given microscopy images $(I_k)_k \subseteq \mathcal{I}$ with corresponding labels $(g_k)_{k=1}^K \subseteq \mathcal{G}$. In image classification problems, we want to find a model $M : \mathcal{I} \to \mathcal{G}$ that makes a good prediction $\hat{g}_k$ of the target g_k, i.e. $M(I_k) = \hat{g}_k$. Note that this problem is a so-called supervised learning problem, where we know the labels of each image to find a model M. By contrast, in so-called unsupervised learning problems, we only know the input data $(I_k)_k \subseteq \mathcal{I}$ but no corresponding label. In unsupervised learning, we want to group observations based on the available input data to identify structures and clusters [HTF09]. For the rest of this thesis, we will focus on supervised learning methods to classify microscopy images.

Classification models depend on model parameters (e.g. intercepts/ biases, weights, hyperparameters) and we find the best parameters by *training* models. Conditional on the model setup, training models amounts to optimising a certain objective function that measures how well the model fits the available data. Still, we are usually not overly interested in a perfect fit of our model on data that we use for training. Instead, we generally aim at training models that perform well on previously unseen new data.

We find such a model by trading off low parameter biases against low parameter variances, which is also known as the bias-variance tradeoff in statistics and machine learning [HTF09]. In this context, biases refer to errors from a wrong model setup and can lead to models that are unable to catch the relations between inputs and targets. This behaviour is also known as *underfitting*. Variances refer to errors caused by small changes of the data and are related to how sensitive models react to these changes. High variances potentially indicate modelling of noise in the data, which is also known as *overfitting* [HTF09].

To ascertain underfitting or overfitting behaviour, one usually splits the available data into the following two to three disjoint samples

1. Training sample $\mathcal{T}$: one uses observations of the training sample to select inputs and train model parameters by optimising the objective functions of the models.

2. Testing sample $\mathcal{T}_{test}$: After training the models, one assesses the model performance on the testing sample to simulate how well the models perform on previously unseen data.

3. Validation sample $\mathcal{V}$: During training, one may use the validation sample to iteratively check up on any over- or underfitting behaviour.

Instead of using a specific validation sample, one can also apply so-called cross-validation (CV) on $\mathcal{T}$ to estimate out-of-sample errors and to identify over- or underfitting behaviour [Sto74]. To that end, one first splits the training data $\mathcal{T}$ into equally sized disjoint subsets (so-called *folds*). Afterwards, one uses all but one fold as a (sub-)training sample and the remaining fold as a validation sample. For each fold, one generates a different pair of (sub-)training and validation samples and averaging the behaviour (e.g. errors or performance metrics as discussed in Section 4.4.1) on all such validation samples serves as a surrogate for out-of-sample behaviour. Commonly, one uses 5 or 10 folds or so-called leave one out cross validation (LOOCV) [HTF09]. In LOOCV, each fold consists of only one observation. Therefore, LOOCCV is computationally most demanding but provides the smallest bias at the expense of greater variance than 5- or 10 fold CV [JWHT13].

In the following two sections, we provide more information on state of the art methods of the FGA in greater detail. We start off in Section 3.3 by discussing state of the art methods to generate features based on microscopy images. Section 3.4 provides an overview of state of the art models that classify images based on previously extracted features.

3.3 Radiomics Feature Generation

For FGA classification models, one first has to manually generate features for each (possibly preprocessed) image $I_k \in \mathcal{I}$. One can consider feature generation as a function $F : \mathcal{I} \to \mathbb{R}^p$ that maps each image to a corresponding p-dimensional real-valued feature vector $F(I_k) = x_k \in \mathbb{R}^p$. Each vector dimension $x_{k,l}, l = 1, \ldots, p$ refers to a separate *feature function* $F_l(I_k) = x_{k,l}$.

Radiomics is state of the art when generating features for x-ray images, magnetic resonance imaging, positron-emission tomography images and computed tomography scans [KGB+12]. Contrary to treating images as pictures for sole visual interpretation, radiomics aims at converting images into advanced quantitative imaging features of first-, second- and higher-order statistics to support decisions [GKH16, KGB+12]. As opposed to *semantic* and rather qualitative features that a radiologist may use to characterise medical imaging, radiomics extracts features computer-driven using potentially higher order image properties and data-characterisation algorithms [GKH16]. Radiomics features have proven to be successful when classifying tumors. For example, radiomics features may identify tumor phenotypes that fail to be perceived by the naked eye [YLP+17].

Radiomics features are so-called agnostic features, which are mathematically extracted quantitative descriptors of the image [GKH16]. First-order statistics describe the distribution of values of individual voxels without considering spatial relationships [GKH16]. These features are generally based on properties of histograms and reduce an image to single values for e.g. mean, median, maximum, minimum of the image's intensities, as well as the skewness and kurtosis of the intensity values [GKH16]. Second-order features are described as "texture" features, as they statistically convey interrelationships between voxels with similar or dissimilar contrast [GKH16]. Higher-order features employ filter

grids on the image to extract repetitive or non-repetitive patterns [GKH16]. They use the gray level intensities of the image to determine pairs of co-occuring gray level intensities (in gray level co-occurrence matrices GLCM), run lengths of gray level intensities (in gray level run length matrices GLCM) as well as zones of gray level intensities (in gray level size zone matrices GLSZM). Pyradiomics is a python library to generate radiomics features for medical imaging [vGFP+17]. In the following, we list exemplary higher-order pyradiomics features:

- *cluster prominence*: cluster prominence measures how asymmetric the GLCM is. Higher values indicate higher asymmetry about the mean while lower values suggest lower asymmetry [vGFP+17].
- *run length non-uniformity*: using the GLRLM, run length non-uniformity quantifies the similarity of run lengths throughout a nucleus. Lower values indicate more homogeneous run lengths in the nucleus [vGFP+17].
- *gray level non-uniformity*: based on the GLRLM, gray level non-uniformity captures the similarity of gray-level intensity values in the nucleus. Lower values indicate a greater similarity in intensity values of the nucleus.

3.4 State of the Art FGA Models

For FGA classification models, we assume that we have features $x_k \in \mathbb{R}^p$ for each microscopy image $I_k \in \mathcal{I}, k = 1, \ldots, K$ given. Hence, x_k is a vector with entries $(x_{k,1}, \ldots, x_{k,p})$. To alleviate notation, we set the first coordinate $x_{k,1} = 1$ to simplify the discussion and to implicitly reflect intercepts (also known as biases) in the state of the art machine learning models that we discuss in the following subsections separately. Note that training the FGA models involves optimising certain error or loss functions. Sections 4.4.1 and 4.4.2 provide general information on loss functions and numerical optimisation via gradient descent.

3.4.1 Logistic Regression

A logistic regression model is a so-called generalised linear model. Given feature vectors $x_k \in \mathbb{R}^p$ with corresponding *binary* labels $(g_k)_{k=1}^K \subseteq \mathcal{G}$ for each image $I_k \in \mathcal{I}$, we want to find linear weights $\beta \in \mathbb{R}^p$ such that

$$g_k \approx \frac{1}{1 + \exp\left(-\beta^T x_k\right)}.$$

$\left(1 + \exp(-\beta^T x_k)\right)^{-1}$ is the so-called *expit* link function that provides a probability estimate for binary labels of $\mathcal{G}$. We find $\beta = (\beta_1, \ldots, \beta_p)$ by maximising the (logarithmic) likelihood function

$$l(\beta) = \sum_{k=1}^{K} \left(g_k \beta^T x_k - \log\left(1 + \exp(\beta^T x_k)\right)\right), \tag{3.1}$$

see Chapter 4.4. of [HTF09] and Chapter 4.3.2. of [Bis06] for details. To avoid overfitting, we can maximise a penalised version of (3.1)

$$\max_{\beta} \left\{ \sum_{k=1}^{K} \left(g_k x_k^T \beta - \log\left(1 + \exp(x_k^T \beta)\right)\right) - \lambda\varphi(\beta) \right\}, \tag{3.2}$$

where $\varphi(\beta) = \|(\beta_2, \ldots, \beta_p)\|_1 = \sum_{i=2}^{p} |\beta_i|$ or $\varphi(\beta) = \|(\beta_2, \ldots, \beta_p)\|_2^2 = \sum_{i=2}^{p} (\beta_i)^2$ are common choices for the penalty function φ. In this case, we are talking about *lasso* and *ridge* logistic regression, respectively. Note that the intercept/ bias β_1 does not enter the penalty function. In an elastic net logistic regression, we combine the L^1 and L^2 penalties of the lasso and ridge logistic regression by maximising

$$\max_{\beta} \left\{ \sum_{k=1}^{K} \left(g_k x_k^T \beta - \log\left(1 + \exp(x_k^T \beta)\right)\right) - \lambda_1 \|(\beta_2, \ldots, \beta_p)\|_1 - \lambda_2 \|(\beta_2, \ldots, \beta_p)\|_2^2 \right\}. \tag{3.3}$$

In the penalised logistic regression of (3.2) and (3.3), $\lambda, \lambda_1, \lambda_2$ are hyperparameters that steer the impact of the penalty terms and that we find via hyperparameter tuning and cross-validation. Generally, increasing λ will decrease the variance of the parameter estimates and lead to so-called *stiffer* models. Sections 4.4.4. and 18.4. of [HTF09] provide further details on penalised logistic regressions and hyperparameter tuning. Note that it is possible to setup logistic regression models for multi-class labels of $\mathcal{G}$, too, see [HTF09] for details. While (penalised) logistic regression models are relatively simple, they are easily explainable. Furthermore, they have shown to provide decent results when classifying microscopy images of different domains [VCX18, DMC15, ZLH+19].

3.4.2 Support Vector Machines

In support vector machines, we aim at separating the groups of binary labels $g_k \in \mathcal{G} = \{-1, 1\}, k = 1, \ldots, K$ with a linear hyperplane of maximum margin [HTF09]. In $\mathbb{R}^{p-1}$, we can define a linear hyperplane by a p-dimensional vector $\beta = (\beta_1, \ldots, \beta_p) \in \mathbb{R}^p$ and the equation

$$x^T(\beta_2, \ldots, \beta_p) + \beta_1 = 0, \; x \in \mathbb{R}^{p-1}.$$

Given feature vectors $x_k \in \mathbb{R}^p$ for each image $I_k \in \mathcal{I}$ and setting $x_{k,1} = 1$ by convention, we note that hyperplanes are given by $x_k^T \beta = 0$. One can show that finding a separating linear hyperplane in our setting is tantamount to solving the following constrained convex optimisation problem:

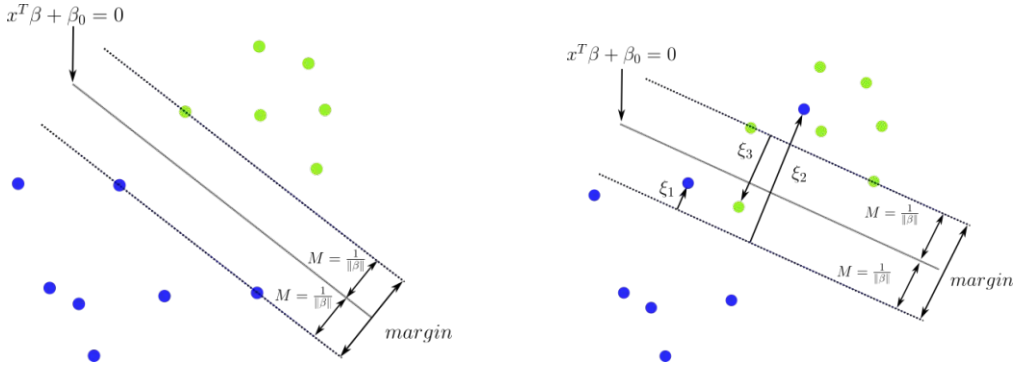


Figure 3.1: Support vector classifiers if linear separation is possible (left) or not (right). Support vectors are observations at the margin border as well as observations x_k with positive slack variable ξ_k. Image was modified from Figure 12.1 of [HTF09].

$$\min_{\beta} \left(\|\beta\|_2^2 + C\|\xi\|_1 \right) \tag{3.4}$$
$$s.t. \begin{cases} g_k x_k^T \beta \geq 1 - \xi_k & \text{for } k = 1, \ldots, K \\ \xi_k \geq 0 & \text{for } k = 1, \ldots, K \end{cases}$$

$1/\|\beta\|_2$ defines the margin width of the hyperplane and the condition $g_k x_k^T \beta \geq 1$ ensures that the groups of $\mathcal{G}$ are indeed linearly separate [HTF09]. ξ_k are so-called *slack variables* that are necessary if a linear separation is not possible. $C > 0$ is a tuning cost parameter that influences the width of the margin and the number of *support vectors*, i.e. observations x_k with $\xi_k > 0$ or x_k that are on the margin border of the separating hyperplane [HTF09, JWHT13]. Decreasing C will lead to bigger margins and usually to stiffer models with smoother decision boundaries that are less prone to overfitting [HTF09, JWHT13]. Figure 3.1 provides an overview of support vectors and hyperplanes if the groups $g_k \in \mathcal{G}$ are separable or not.

We solve (3.4) via its dual optimisation problem and the corresponding Lagrangian [HTF09]. The Lagrangian depends on inner products $x_k^T x_l$, $k, l = 1, \ldots, K$ and is simpler to optimise, see also Sections 4.4.1 and 4.4.2 for general information on loss functions and numerical optimisation via gradient descent algorithms. Via the so-called kernel-trick, support vector machines can be easily defined and solved for more complex non-linear decision boundaries [HTF09]. For that purpose, one transforms the input vectors x_k via functions $\phi : \mathbb{R}^p \to \mathbb{R}^P$, where usually $P \geq p$. Even after transformation, the structure of the convex optimisation problem is unchanged and we can again solve it by optimising the corresponding Lagrangian. In this case, the inputs x_k only enter the Lagrangian via inner products $K(x_k, x_l) = \phi(x_k)^T \phi(x_l)$, which we refer to as *kernel functions*. Popular kernel functions are for example radial basis kernel functions $K(x_k, x_l) = \exp\left(\frac{\|x_k - x_l\|_2}{2\sigma^2}\right)$ [HTF09]. Chapters 4 and 12.2 of [HTF09] as well as Chapter 6 of [Bis06] provide further details and information on support vector machines. Support vector machines have proven

to be competitive when classifying microscopy images [AKTO15, WSC+19]. While our outline of support vector machines is based on binary labels $\mathcal{G} = \{-1, 1\}$, note that it is also possible to define support vector machines for multi-class problems [AKTO15].

3.4.3 Random Forests

Random forests are ensemble models that aim at creating a large collection of lowly correlated decision trees and use the average decision of these trees as prediction [Bre01]. To ensure low correlation among the trees, random forests build the individual decision trees on different bootstrap samples and split the trees at each node by considering a random selection of the available predictors. To find the best split among this selection, random forests may use different criteria, such as minimising so-called Gini impurity [Bre01], see also Section 4.4.1 for general information on loss functions. Random Forest have proven to successfully classify microscopy images of different domains [MSS+13, KVRKB+15]. As each tree can be grown individually and independently from the others, training random forest can be easily parallelised [KVRKB+15].

Random forests depend an many hyperparameters that influence the setup and shape of the individual decision trees [Bre01, HTF09]. The most important hyperparameters of *sklearn*'s implementation of random forest are the following [PVG+11]:

- `n_estimators`: defines the number of individual decision tress in the random forest.
- `max_depth`: defines the maximum depth of each individual decision tree.
- `min_samples_split`: defines the minimum number of observations that are necessary to split a node.
- `min_samples_leaf`: defines the minimum number of observations that have to be in a leaf node (terminal node).
- `max_samples`: defines the size of the boostrap sample to build the decision tree.
- `max_features`: defines the number of randomly selected features when splitting a node.

To avoid overfitting, one can choose `n_estimators` and `min_samples_leaf` large (i.e. commonly greater than or equal to 500 and 5, respectively), as well as `max_features` and `max_samples` small (i.e. commonly equal to the square root of the number of features and less than 50% of all samples, respectively)[HTF09, PVG+11]. [Bre01] and Chapter 15 in [HTF09] provide further details and information on random forests.

3.4.4 Gradient Boosting

In gradient boosting, one finds a sequence of simple decision trees that aim at stepwise improving classification results. More specifically, at step l the previously found trees classify observations x_k via $F_l(x_k)$. By using a loss function L and considering the residual prediction errors $\sum_{k=1}^{K} L(F_l(x_k), g_k)$, we build another boosting tree T_l to correct these errors (boosting step). We then set $F_{l+1} = F_l + \eta T_l$ as classification model of step $l+1$ and repeat the procedure. Here, $0 < \eta \leq 1$ refers to the learning rate of the boosting approach.

As opposed to Random forest, gradient boosting are not easy to parallelise as each tree depends on the previously found trees. Still, they have shown to succeed in classifying microscopy images [VCX18, RSIK18].

There are various parameters to influence gradient boosting and avoid overfitting. *XGBoost* is the most prevelant framework that implements gradient boosting trees [CG16]. The most important hyperparameters of *XGBoost* are the following:

- `eta`: learning rate of the boosting procedure.
- `n_estimators`: controls the number of boosting trees.
- `gamma`: defines minimum loss reduction required to further split a node.
- `max_depth`: defines the maximum depth of each individual boosting tree.
- `min_child_weight`: defines a minimum threshold for the second derivative of the loss function L in each node. One does not split trees further if the nodes fall below this threshold. This ensures that the model stops splitting trees once a node features a certain degree of purity.
- `subsample`: defines the size of a bootstrap sample to build the boosting tree. Each boosting step considers another bootstrap sample.
- `reg_lambda`: defines an L^2 regularisation parameter similar to ridge logistic regression, see above. Each boosting tree T_l gives a prediction (score) $T_l(x_k) = w_{k,l}$. Overly large values of $w_{k,l}$ indicate a skittish behaviour of the boosting model which we can dampen by adding $\|(w_{k,l})_k\|_2$ to the loss function L. `reg_lambda` controls the impact of $\|(w_{k,l})_k\|_2$ in the loss function.
- `colsample_bytree`: defines the share of randomly selected features that are used to construct a boosting tree. In each boosting step, we newly select the features.

To avoid overfitting, we can choose `eta` small (e.g. commonly smaller than 0.3 which usually requires setting `n_estimators` large, e.g. greater than 500), `gamma` large (e.g. commonly greater than 0), `max_depth` small (e.g. commonly smaller than 6), `min_child_weight` large (e.g. commonly greater than 1), `subsample` small (e.g.

smaller than 1), `reg_lambda` large (e.g. greater than 1) and `colsample_bytree` small (e.g. smaller than 1) [CG16]. Chapter 10 of [HTF09] provides further details on gradient boosting trees.

3.5 Summary

In this chapter, we outlined state of the art methods of machine learning for microscopy images. We discussed models of the FGA to classify images, i.e. models that predict classes based on previously generated features of the images.

The state of the art data scientific setup for training models and generating reliable out-of-sample performance metrics requires splitting data into disjoint training, validation and testing samples. Cross-validation may replace dedicated validation samples if deemed computationally feasible.

Radiomics features aim at converting microscopy images into advanced quantitative imaging features of first-, second- and higher-order statistics. We have learned that radiomics features have proven successful in classifying microscopy images of tumors.

The state of the art FGA models that we discussed in this chapter comprise (penalised) logistic regression, support vector machines, random forests and gradient boosting. Logistic regressions are relatively easy to understand extensions of linear regressions for binary classification problems. Their penalised versions include penalty terms to account for overfitting. Support vector machines classify observations based on decision hyperplanes. These hyperplanes allow for more intricate than linear decisions when applying high-dimensional kernel functions (such as radial basis functions). Random forests are ensemble models for creating a large collection of lowly correlated decision trees and use the average decision of these trees as prediction. Conversely, gradient boosting trees are correlated and aim at stepwise improving classification results by building boosting trees that correct residual prediction errors. We have learned that all of the afore-mentioned four FGA models have given decent results in various domains of classifying microscopy images.

CHAPTER 4

State of the Art: Deep Learning for Microscopy Images

The image-based classification approach refers to so-called deep learning approaches or neural networks that consist of many stacked layers. Each layer is itself a comparatively simple linear or non-linear transformation that takes the previous layer as input and outputs to the next layer. We can conceive inputs and intermediate layer outputs as neurons of a directed acyclical graph. State of the art IBA classification models include so-called convolutional neural networks and residual networks [XXS+17, HZRS16].

After setting the notation in Section 4.1, the following Sections 4.2 and 4.3 discuss convolutional neural networks and residual networks for image classification. Section 4.4 gives further details on how to train such networks.

4.1 Denotation

In addition to the notation introduced in Section 3.1, we denote (high-dimensional) learnable parameters of neural networks as $\boldsymbol{\theta}$, which we split into multiplicative parameters $\boldsymbol{\theta}_w$ and additive parameters $\boldsymbol{\theta}_b$. We denote the loss function by L, its regularised version as L^{reg} and batches of the training data as $\mathcal{B}_i$ (see Sections 4.4.1 - 4.4.3). We refer to $\lambda > 0$ as hyperparameter that controls the impact of the penalty term in L^{reg}.

4.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) for image classification are deep learning networks that take images I_k as direct inputs and pass them through multiple feed forward layers. CNNs have proven to excellently perform when classifying images in various domains

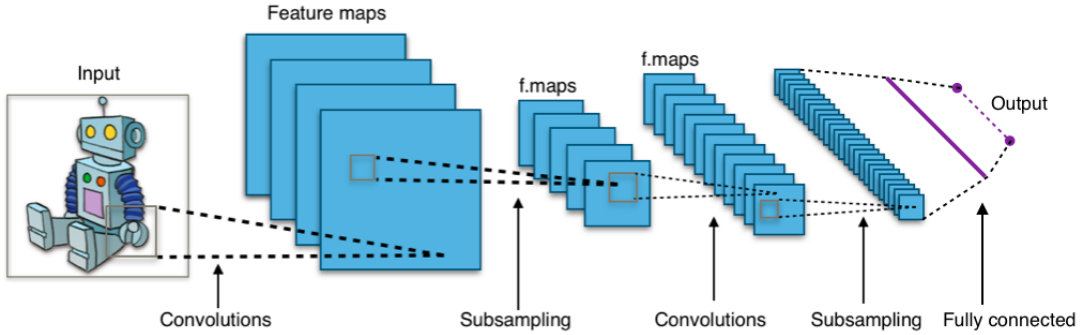


Figure 4.1: A simplified layout of a typical CNN. Activation functions are applied after each convolution.[a]

[a]Image taken from Wikimedia Commons https://commons.wikimedia.org/wiki/File:Typical_cnn.png. Accessed: 2023-01-24.

[KSH12], medical image analysis [CGGS13, LST+16] including histopathology images [SRT+16, XXS+17].

Besides special layers (such as dropout and batch-normalisation layers) that we discuss in Section 4.4.3, there are usually four different kinds of layers/ functions in a CNN:

1. convolutional layers,
2. activation functions,
3. subsampling or pooling layers,
4. linear layers.

All layers serve different purposes: broadly speaking, stacking convolutional layers, pooling layers and activation functions aims at extracting features and properties from an input image that are relevant for classification. In modern CNN architectures, linear forward connected layers are classification backends that take these features in the final layer as inputs and use them to classify the input image. Before the final layer, modern CNNs usually reiterate layer blocks where each block consists of convolutional layers, activation functions and a pooling layer, see Figure 4.1. The followings Sections 4.2.1-4.2.4 discuss each layer type in more detail.

4.2.1 Convolutions

Convolutional layers are shift invariant feature extractors that employ (possibly multiple) so-called convolutions. Convolutions take a multidimensional array I as input and use another multidimensional array of parameters K (called kernel) to extract features by generating an output array O via

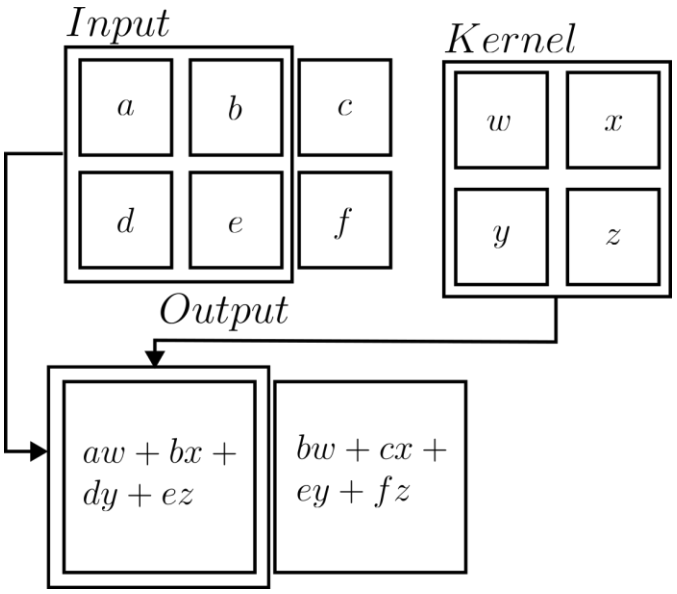
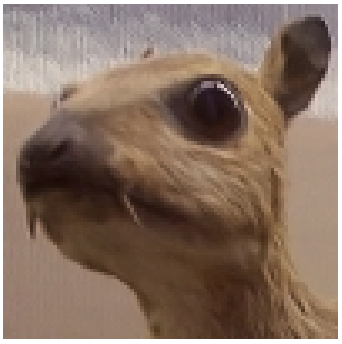


Figure 4.2: A convolutional kernel of size 2 acts on an input array of size 2×3 to create an output array of size 1×2. Image was modified from Figure 9.1 of [GBC16].

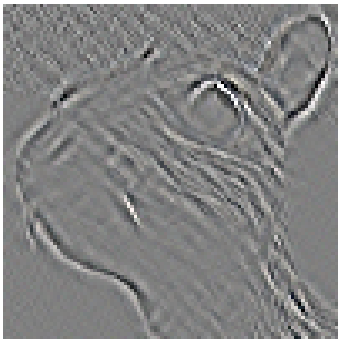
$$O_{i,j} = \sum_m \sum_n I_{m,n} K_{i-m,j-n}.$$

We set $K_{l,m} = 0$ except for a predefined number of array elements which determines the kernel size. We can omit those predefined entries $K_{l,m} = 0$ for simplicity. The remaining kernel parameters $K_{l,m}$ are learnable and modified during training [GBC16]. Figure 4.2 displays how a kernel of size 2×2 acts on an input array of size 3×4 to create another output array of size 2×3. In particular, we note that convolutional layers decrease the resolution of the output array. To influence the resolution of the output array, one can use so-called *paddings* or *strides* for the convolutional layers. Paddings increase the resolution of the input array by creating a "frame" (pad) of 0-entries around the input array. Hereby, also the resolution of the output array will increase based on the padding width. Strides define to which input entries the kernel is applied. Figure 4.2 shows the output O of a kernel with stride 1, while a kernel with, say, stride 2 will output only every second entry of O, i.e. an array of size 2×2. Chapter 9 of [GBC16] provides further details on convolutional layers.

A single kernel can extract local low-level features of an input array, such as edges or ridges, see Figure 4.3. A convolutional layer may consist of multiple kernels that extract features in separate so-called feature maps, which increase the channel dimension of the output array. Stacking multiple convolutional layers increases the so-called *receptive field* of the feature detection and allows the CNN to learn more complex high-level features [GBC16]. One can also increase the receptive field by modifying the convolutional kernels. For example, given a kernel, a dilated version of this kernel will increase the kernel size



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



(a) input image (b) applied kernel (c) output image

Figure 4.3: Example of a 3×3 kernel applied to an input image to extract edges and ridges as output.[a]

[a]Images taken unchanged from Wikimedia Commons `https://commons.wikimedia.org/wiki/File:Vd-Orig.png` and `https://commons.wikimedia.org/wiki/File:Vd-Rige2.png`. Both accessed: 2023-01-24.

by spatially uniformly distributing the original kernel values in the greater kernel and filling all unallocated kernel entries with 0. The dilated kernel will operate on a greater receptive field than the original kernel at the expense of capturing fewer details [GBC16].

4.2.2 Activation Functions

Convolutional layers linearly transform inputs into outputs. In a CNN, each output is then run through a non-linear *activation* function a to only keep stronger and therefore more relevant signals. Sigmoid/ logistic activation functions $a(x) = (1+\exp(-x))^{-1}$, area hyperbolic tangent functions $a(x) = \operatorname{arctanh}(x)$ and so-called rectified linear units (ReLU) are common activation functions [XXS+17]. Here, ReLU is defined by $a(x) = \max(x, 0)$, where we apply the maximum function element-wise. ReLU activation functions are easy to optimise with gradient-based approaches [GBC16], see Section 4.4.2.

4.2.3 Subsampling or Pooling Layers

A pooling or subsampling layer replaces the output of a layer at a given location with a summary statistic of nearby outputs [GBC16]. Most popular pooling layers are max pooling or average pooling layers which output the maximum or mean within a rectangular neighborhood [XXS+17]. Pooling layers allow condensing the signals to reduce dimensionality and only keep the most relevant information.

4.2.4 Linear Layers

As stated above, in modern CNNs the linear layer is a classification backend that uses the previously extracted features to classify the input image $I_k \in \mathcal{I}$. The output of a CNN is a vector $s_k \in \mathbb{R}^l$ where each vector entry $s_{1,k}, \ldots, s_{l,k}$ gives a class score, i.e. a

real number that expresses how likely the input image I_k is in class $i \in \mathcal{G} = \{c_1, \ldots, c_l\}$. In classification tasks, one assigns I_k to the class $\hat{g}_k := argmax_{i \in \mathcal{G}} s_{i,k}$ of the greatest class score. One can use a softmax function

$$softmax(s_{i,k}) := p_{i,k} := \frac{\exp(s_{i,k})}{\sum_{j \in \mathcal{G}} \exp(s_{j,k})}$$

to convert class scores $s_{i,k}$ into class probabilities $p_{i,k}, i \in \mathcal{G}$. The softmax function emphasises large class scores and suppresses small scores.

4.3 Residual Networks - ResNets

As CNNs have achieved remarkable results in image classification problems [XXS+17], a common idea to even further improve these results was to increase the model depth and stack even more convolutional layers. However, from a certain depth onwards, one observed performance saturation and degradation. Adding more layers increases the training and test errors instead of decreasing them [HZRS16]. This is remarkable, because in theory convolutional layers should be able to mimic the identity function. Hence, deeper CNNs should perform at least as well on the training sample $\mathcal{T}$ as more shallow CNNs. However, in practice, too deep CNNs fail to mimic identity functions in their convolutional layers.

To overcome this issue, residual networks are extensions of CNNs that include so-called skip-connections. Skip-connections represent identity functions that allow ignoring (skipping) the output of certain convolutional layers, see Figure 4.4. Residual networks facilitate much deeper networks and have shown to outperform plain CNNs in image classification problems [HZRS16]. Those networks are named ResNet-N, where N refers to the number of layers with learnable parameters (e.g. ResNet-34). Due to the high number of parameters and deep layers, it is common to use pre-trained off-the-shelf ResNet models and only fine tune the final classification layer depending on the specific task that one wants to solve. In that manner, one can re-use the feature extractor of the ResNet model and employ the features for other classification task. This method provided competitive results in various domains of microscopy image classification [FMS+18, MSR+20].

4.4 Training CNNs and ResNets

As discussed in the previous sections, CNNs and ResNets usually consist of thousands or even millions of learnable parameters in addition to hyperparameters such as kernel sizes, padding and stride settings. We can conceive a CNN or ResNet as a function $M(\cdot, \boldsymbol{\theta}) : \mathcal{I} \to \mathcal{G}$, where $\boldsymbol{\theta}$ denotes the (high-dimensional) vector of learnable parameters. We can separate theses parameters into weights $\boldsymbol{\theta_w}$ (multiplicative parameters) and biases $\boldsymbol{\theta_b}$ (additive parameters). To train, validate and test models, we assume throughout this

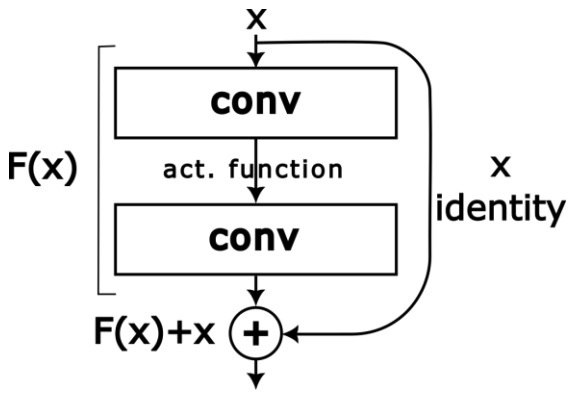


Figure 4.4: Example of a skip connection. The output $F(x)$ of two stacked convolutional layers is added to the input x via a skip connection. Image was modified from [HZRS16].

section that a training sample $\mathcal{T}$, a validation sample $\mathcal{V}$ and a testing sample $\mathcal{T}_{test}$ are given as introduced in Section 3.2.

To train $\boldsymbol{\theta}$ and to find the optimal parameter settings of $\boldsymbol{\theta}$, we first have to define a so-called loss function L and metrics that measure how well our model fits the data. We then iteratively update the parameters in a gradient descent algorithm by determining the gradient $\nabla_{\boldsymbol{\theta}} L(M(\cdot, \boldsymbol{\theta}))$ of the loss function with respect to the learnable parameters. Deep learning networks are prone to overfitting due to the high number of parameters compared to the number of training samples (curse of dimensionality) and regularisation methods are necessary to combat overfitting [GBC16].

Below, Section 4.4.1 introduces loss functions and performance metrics for binary image classification problems. Afterwards, Section 4.4.2 discusses gradient descent algorithms and related improved methods. Finally, Section 4.4.3 treats methods to spot and combat overfitting when training networks.

4.4.1 Loss Functions and Metrics

Loss functions aim at measuring the performance of a network $M(\cdot, \boldsymbol{\theta})$ on the training sample $\mathcal{T}$ with respect to parameters $\boldsymbol{\theta}$. In general, choosing the loss functions depends on the task that the network tries to solve. For classification tasks, the cross-entropy loss function is most common [GBC16]. Other loss functions include the mean absolute error or mean squared error function [GBC16, Bis06]. In case of unbalanced samples in which one class exhibits more observations than others, one can also employ a weighted loss function. In weighted loss functions, one assigns weights to each observation to influence how much they contribute to the total loss. This way, one can weigh observations of majority classes down to favour models that perform well also on minority classes.

To define the cross-entropy loss function for binary classifications, we assume that our model $M(\cdot, \boldsymbol{\theta})$ outputs for each input image $I_k \in \mathcal{T}$ with label $g_k \in \{0, 1\}$ a vector $p_k = (p_{k,0}, p_{k,1})$ of class probabilities, see Section 4.2.4. We encode the label g_k also into

a class probability vector $u_k = (u_{k,0}, u_{k,1})$, where $u_{k,i} = 1$ if $i = g_k$ and $u_{k,i} = 0$ otherwise. Note that $u_{k,0} = 1 - u_{k,1}, p_{k,0} = 1 - p_{k,1}$. We then define the cross-entropy loss of our model $M(\cdot, \boldsymbol{\theta})$ on $\mathcal{T}$ as

$$\begin{aligned} L(M(\mathcal{T}, \boldsymbol{\theta})) &= -\sum_{I_k \in \mathcal{T}} \left(u_{k,0} \ln(p_{k,0}) + u_{k,1} \ln(p_{k,1})\right), \\ &= -\sum_{I_k \in \mathcal{T}} \left(u_{k,0} \ln(p_{k,0}) + (1 - u_{k,0}) \ln(1 - p_{k,0})\right). \end{aligned}$$

The cross-entropy loss measures how similar or dissimilar the true probabilities u_k and the predicted probabilities p_k are and how well the model $M(\cdot, \boldsymbol{\theta})$ performs for each observation $I_k \in \mathcal{T}$. By definition of u_k and p_k, we can rewrite the cross-entropy loss as

$$L(M(\mathcal{T}, \boldsymbol{\theta})) = -\sum_{I_k \in \mathcal{T}} u_{k,g_k} \ln\left(softmax\left([M(I_k, \boldsymbol{\theta})]_{g_k}\right)\right),$$

where $[M(I_k, \boldsymbol{\theta})]_j$ denotes the j-th component of the 2-dimensional output vector of $M(I_k, \boldsymbol{\theta})$. Chapter 6 of [GBC16] provides further details on loss functions (objective functions).

While lower values of loss functions relate to better model performances, they are still hard to interpret in practice. In addition, regularisation methods may amend the loss function and make it even harder to construe losses (see Section 4.4.3). For that reason, we usually do not assess losses on the validation sample $\mathcal{V}$, but use metrics that are easier to understand than cross-entropy losses. The most relevant metrics for a binary classification problem are the following:

- *accuracy*: For each observation $I_k \in \mathcal{V}$ we can compare the true label g_k with the predicted label $\hat{g}_k$ and infer true and false positives, as well as true and false negatives. By denoting the sum of all true and false negatives as well as true and false negatives over $\mathcal{V}$ as TP, FP, TN, FN, respectively, we define accuracy as

  $$acc = \frac{TP + TN}{K},$$

  where $K = TP + FP + TN + FN$ denotes the number of observations in $\mathcal{V}$. The metric *acc* measures how often $M(\cdot, \theta)$ outputs correct results.

- *recall*: recall measures how often the model $M(\cdot, \theta)$ correctly classifies (by convention) the class $1 \in \mathcal{G}$ and is defined as

  $$rec = \frac{TP}{TP + FN}.$$

- *precision*: precision measures how often the model's classification as $1 \in \mathcal{G}$ is correct and is defined as
$$prec = \frac{TP}{TP + FP}.$$

- *F1-score*: Both recall and precision are important metrics to measure different aspects of the model performance. The F1-score combines them into one score by determining their harmonic mean
$$f1 = 2\frac{rec \cdot prec}{rec + prec}.$$

When training a network, we compare training losses on $\mathcal{T}$ as well as a suitable performance metric on the validation sample $\mathcal{V}$. After training, we can assess the model via the relevant performance metric on the testing sample $\mathcal{T}_{test}$. In Section 4.4.3, we discuss in more detail how we can spot overfitting during training.

4.4.2 Gradient Descent

Given current parameters $\boldsymbol{\theta}_i$, we generate losses $L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ on the training sample $\mathcal{T}$ for a network M. We can determine the gradient $\nabla_{\boldsymbol{\theta}} L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ of $L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ with respect to $\boldsymbol{\theta}$ to find the directions of the greatest increase and decrease of losses $L(M(\mathcal{T}, \boldsymbol{\theta}_i))$. Choosing a small *learning rate* $\alpha > 0$ as hyperparameter (e.g. $\alpha = 0.01$), we can then update θ iteratively via

$$\theta_{i+1} = \theta_i - \alpha \nabla_{\boldsymbol{\theta}} L(M(\mathcal{T}, \boldsymbol{\theta}_i)). \tag{4.1}$$

It is important to note that $\nabla_{\boldsymbol{\theta}} L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ is indeed the true analytical gradient and not a numerical (approximated) derivative. We are able to calculate the exact gradient because the simple component functions (convolutions, pooling layers, activation functions) of our network feature (relatively) simple gradients. Using the chain rule, the gradient $\nabla_{\boldsymbol{\theta}} L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ is a sum of products of gradients of the simple component functions of our network. As we can conceive a deep forward neural network as a directed acyclical graph, the so-called backpropagation algorithm allows calculating the gradient $\nabla_{\boldsymbol{\theta}} L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ efficiently by storing and reusing intermediate gradients in each node of the graph [GBC16].

Calculating $L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ and its gradient $\nabla_{\boldsymbol{\theta}} L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ over the whole training sample $\mathcal{T}$ can be computationally demanding. As one needs many iterations to update $\boldsymbol{\theta}$ via (4.1), determining $\nabla_{\boldsymbol{\theta}} L(M(\mathcal{T}, \boldsymbol{\theta}_i))$ is even more serious. To alleviate the computational complexity, we partition $\mathcal{T}$ randomly into disjoint batches $\mathcal{B}_i$ of a given batch size S, i.e. $|\mathcal{B}_i| \leq S$ and $\bigcup_i \mathcal{B}_i = \mathcal{T}$. We then update $\boldsymbol{\theta}$ for each batch $\mathcal{B}_i$ separately by calculating the loss

$$L(M(\mathcal{B}_i, \boldsymbol{\theta})) = -\sum_{I_k \in \mathcal{B}_i} \sum_{j=0}^{1} u_{k,j} \ln(p_{k,j}), \tag{4.2}$$

and its corresponding mini-batch gradient $\nabla_{\boldsymbol{\theta}} L(M(\mathcal{B}_i, \boldsymbol{\theta}))$ for each batch $\mathcal{B}_i$. Decreasing the batch size S also decreases the computational effort to update $\boldsymbol{\theta}$, the required GPU memory but also the accuracy of gradient estimates. Hence, finding an adequate batch size S is crucial. One swipe through the whole training sample $\mathcal{T}$ via batches is called *epoch*. We shuffle training observations and newly setup batches after each epoch.

In optimisation problems, there are global and local optima, as well as so-called critical points that feature very small gradients that are (in terms of length) close to machine precision. Not all critical points are optima, but they let plain gradient descent algorithms stop because of their vanishing gradient. However, mini-batch gradients are known as noisy estimates and are able to let the gradient descent algorithm escape critical points that are not optima [Bis06].

Plain gradient descent as implemented in (4.1) or (4.2) ignores any information from previous iterations. This can cause oscillations and slow down the speed of convergence. For that reason, there are two main ideas among many others to improve plain gradient descent: first, one adapts gradient descent by so-called *momentum*, which incorporates an exponential moving average of previous gradients in gradient descent. The influence of previous gradients decays exponentially with the number of iterations and improves the speed of convergence by dampening oscillations. An additional hyperparameter $\beta > 0$ controls the impact of previous gradients. Second, in so-called *RMSprop*, one adapts parameters with higher variance less than parameters with lower variances. This also helps dampening oscillations and improves the speed of convergence. One of the currently most popular algorithms to improve plain gradient descent is *Adam*, which incorporates both momentum as well as RMSprop. It includes hyperparameters $\alpha > 0$ (learning rate) and $\beta_1, \beta_2 > 0$, where the latter impact momentum and RMSprop, respectively. Chapter 8 in [GBC16] provides further details. While we discussed gradient descent and loss functions in this chapter in the setup of deep learning, note that these concepts also apply beyond. Finding optimal parameters and training e.g. machine learning models (such as random forests, support vector machines, gradient boosting and logistic regression as discussed in Chapter 3) may also involve optimising loss functions via numerical methods of gradient descent.

4.4.3 Combatting Overfitting

When training a network, we record training losses on $\mathcal{T}$ as well as a suitable performance metric on the validation sample $\mathcal{V}$ for each epoch. Due to the high number of parameters, deep learning networks are usually prone to overfitting [GBC16]. To spot overfitting during training, we compare the losses on $\mathcal{T}$ with the performance metric on $\mathcal{V}$. If losses on $\mathcal{T}$ keep on decreasing for each epoch or are already close to 0 while the performance

metric on $\mathcal{V}$ remains high or even gets worse, this is usually a sign of overfitting because the model is not capable of generalising well to unseen data of $\mathcal{V}$.

To avoid overfitting and to train a model that copes with new data, it is best to increase the training sample $\mathcal{T}$. However, in practice, data are limited and increasing $\mathcal{T}$ is not feasible. Instead, we can apply certain strategies to combat overfitting, among which the most popular are the following five approaches:

- *data augmentation*: one can use images $I_k \in \mathcal{T}$ and transform them without affecting the class label g_k by using a transformation $\phi : \mathcal{I} \to \mathcal{I}$ and employing $\phi(I_k)$ as new training observations (fake data) [Bis06]. This method is known as data augmentation and common transformations ϕ include random image cropping, contrast, sharpness an intensity changes, as well as random rotations, scaling and adding noise [Bis06]. Modern hardware allows applying these transformation online during training such that storing transformed images is not necessary.

- *regularisation*: similar to ridge logistic regression, see Section 3.4, we can amend the loss function L by introducing a term that penalises large network weights $\boldsymbol{\theta_w}$:

$$L^{reg}(M(\mathcal{T}, \boldsymbol{\theta})) = L(M(\mathcal{T}, \boldsymbol{\theta})) + \frac{\lambda}{2}\|\boldsymbol{\theta_w}\|_2^2,$$

  where $\lambda > 0$ is a hyperparameter that controls the impact of the penalty term [Bis06]. Similarly to ridge regression, the regularised loss function prevents certain inputs from dominating the model prediction and usually leads to a model that resides to all inputs. Normally, we choose $\lambda \in [0.0001, 0.01]$.

- *batch normalisation*: deep networks involve chained multiplications that can lead to vanishing or exploding networks signals. For that reason, one can normalise input images either by ensuring that the channel intensities fully cover a pre-defined interval (say $[0, 255]$; known as *min-max normalisation*) or by subtracting from each channel the corresponding mean and dividing by the standard deviation of the channel on $\mathcal{T}$ [GBC16]. Batch normalisation layers includes the latter approach also in the hidden layers of a network by determining the channel means and standard deviation of the currently processed batch $\mathcal{B}_i$ [GBC16]. During training, batch normalisation layers aggregate statistics on means and standard deviations of the processed inputs that one uses after training to normalise the input neurons [GBC16]. Batch normalisation ensures consistent inputs signals and better behaved loss functions. As mini-batch means and standard deviations are usually more noisy, batch normalisation layers also have a regularising effect on the network similar to intensity changes in data augmentation.

- *dropouts*: dropouts are specific network layers that only amend inputs while training the network. After training, dropout layers pass inputs unchanged as outputs. During training, dropout layers output for each input neuron 0 with a predefined probability p and otherwise output the unchanged input neuron. One usually

places dropout layers before the last linear layer to "drop" certain feature neurons. Typically, this turns models more robust and less dependent on specific features to classify an input image [HSK+12].

- *early stopping*: As discussed above, we usually identify overfitting if the validation performance metric does not improve anymore during training. A simple but yet effective method to prevent overfitting is to stop training whenever validation performance has not improved anymore after a predefined number of epochs [Bis06]. One then uses the model of the last improvement as final model.

4.5 Summary

In this chapter, we outlined CNNs and ResNets as state of the art methods to classify microscopy images via deep learning. CNNs consist of different kinds of layers, such as convolution layers, activation functions, subsampling layers and linear layers which are usually placed at the end of the networks to classify images. ResNets amend these layers via skip-connections to allow for deeper and more predictive networks.

We have also learned the state of the art for training CNNs and ResNets. The cross-entropy loss function is one of the most common loss functions for binary classification problems and there are different kinds of metrics (such as accuracy) to monitor model performance during training. The plain gradient descent algorithm for batches of images allows for easily training neural networks. Momentum and RMSprop enhance this method by increasing the speed of convergence and avoiding oscillations. There are several methods to prevent and combat overfitting of networks, such as batch normalisation, dropouts, early stopping, data augmentation and regularisation of the loss function.

CHAPTER 5

State of the Art: Wasserstein Distances and Classification of ALT Status and Fluorescence Patterns

This chapter discusses state of the art methods to classify the ALT Status and fluorescence patterns of so-called HEp-2 cells. To the best of our knowledge, there are no FGA or IBA models yet in the literature to predict the ALT status of tumor cells. Classifying HEp-2 cells based on immunofluorescence microscopy images represents a very active research topic in the last ten years. HEp-2 cells and their corresponding fluorescence patterns are relevant for identifying antibodies in blood serum e.g. to diagnose autoimmune diseases. The patterns partly exhibit bright spots which vary in shape, position and density within the cells [oAp]. For that reason, HEp-2 classification models are also promising candidates to identify the ALT status of nuclei.

Furthermore, this chapter outlines basic concepts of the theory of Wasserstein distances, the optimal transport problem and Wasserstein barycenters. Sections 6.6.3 and 7.2.2 below use Wasserstein distances to define a so-called Wasserstein distance model and to select features. Wasserstein distances have become increasingly popular in machine learning, in particular when training generative adversarial networks [ACB17].

In this chapter, we start off in Section 5.1 by summarising the main notation of this chapter. The following Section 5.2 treats state of the art methods to classify the ALT status of cells. Section 5.3 discusses the state of the art for classifying HEp-2 cells with CNNs in indirect immunofluorescence microscopy images. Last, Section 5.4 discusses the theory of Wasserstein distances.

5.1 Denotation

In this thesis, we denote measures on $\mathbf{R}^p$ by μ and refer to the L^2 norm on $\mathbf{R}^p$ as $\|\cdot\|_2$. We consider $x, x_k \in \mathbf{R}^p$ as vectors in $\mathbf{R}^p$.

5.2 State of the Art for ALT Classification

[MWT+21] provides a concise overview of the state of the art methods for predicting the ALT status. There are four main approaches: first, there exist rule-of-thumb criteria to categorise the ALT status based on microscopy images. Second, Whole Genome Sequencing (WGS) is used to predict the ALT status of cells. Third, one uses the so-called C-circle assay to determine the ALT status. Fourth, one refines the fluorescence microscopy imaging approach to suppress telomeric signals for ALT− cells. The following four paragraphs discuss these approaches separately.

To diagnose the ALT status using telPNA FISH microscopy images, [HSH+11] proposed certain rule-of-thumb criteria. More specifically, to categorise ALT positivity, it is necessary to identify ultra-bright telomere spots in the microscopy images and to quantify these cells. If more than 1% of all cells exhibit these ultra-bright spots, [HSH+11] proposed to categorise them as ALT+. While these principles are useful indications when ultra-bright telomere spots are present, they are not clear-cut and objective automated rules to determine the ALT status with high confidence, in particular if no ultra-bright telomere spots are visible.

Instead of using microscopy images, [LTH+18] employ WGS to identify the ALT status of cells. More specifically, the authors train a random forest model on WGS tumors to classify the ALT status with high in-sample and out-of-sample accuracy. In particular, despite being developed on a sample of very specific classes of tumors, the random forest classifier manages to satisfactorily predict the ALT status out-of-sample for various kinds of tumors. Hence, different kinds of ALT+ cancer are likely to share a common genome sequencing profile.

Another possibility for identifying the ALT status is the so-called C-circle assay, which is a polymerase chain reaction assay that makes use of the fact that telomer elongation is involving circular intermediates in ALT+ cells [HCH+09, HR10]. While the C-cirlce assay is comparably easy to perform, telPNA FISH is more reliable and diagnosing faster [MWT+21]. Another disadvantage of the C-circle assay and also WGS is that both methods analyse cells in bulk and not on single-cell level. Hence, one cannot determine the exact ratio of ALT+ cells and low ratios might not be detected at all.

[FRM+22] introduce a special one-step FISH microscopy imaging approach (ALT-FISH) that aims at suppressing the telomeric signal for ALT− cells. They show that single-stranded DNA and RNAs containing telomere sequences that are rich in the nucleotide bases cytosine and guanine are reliable markers for the ALT status [FRM+22], which can be quantified visually by ALT-FISH. More specifically, the ALT-FISH approach shows

considerably more telomeric spots in ALT+ cells than in ALT− cells. Note that this method is not applicable in our setting as we use telPNA and not ALT-FISH images.

5.3 State of the Art of HEp-2 Classification Via Fluorescence Patterns

[RWSZ20] provides a recent and comprehensive review on deep learning approaches to predict the HEp-2 category on cell-level as well as specimen-level, where the latter refers to an image that captures multiple cells similar to field of view in dilution series, see Section 2.3.2. While also deep autoencoding-classification networks have been used recently to classify HEp-2 cells, most approaches use custom CNNs or build on employing existing CNN architectures (such as GoogLeNet or AlexNet). When using CNNs to classify HEp-2 cells, there are two main paradigms: on the one hand, one uses CNNs to generate self-trained features which then serve as predictors in another classification model (such as a SVM). On the other hand, one uses CNNs to directly classify the input by using the self-learned features. There are four publicly available standardised data sets of HEp-2 cells that allow comparing results across publications. The following three paragraphs record the state of the art approaches for CNNs that are based on the relatively simple LeNet-5 network [LBBH98].

[GWZZ16] propose a custom CNN to directly predict the HEp-2 category on cell-level. The custom CNN shares the basic architecture of LeNet-5 but uses different filter sizes, max-pooling instead of average-pooling layers and additional subsampling layers. The authors investigate the impact of rotation-based data augmentation and image foreground masks on the classification performance. They have found that augmenting data via rotations significantly improves the prediction accuracy of the final CNN. Furthermore, the authors state that applying image foreground masks decreases the classification performance compared to a CNN that is trained on cell images that also includes the background. Hence, they have found in their experiment that the background of HEp-2 cells includes information to predict HEp-2 categories with their custom CNN.

The authors of [RNM17] use the three well established CNN architectures LetNet-5, AlexNet and GoogleNet to classify HEp-2 cells. They assess how various preprocessing approaches (contrast stretching, histogram equalisation, pixel substraction methods) and data augmentation (image rotation based on predefined angle steps) affect the performance. They have found that each CNN behaves differently for the proposed preprocessing strategy. While GoogleNet provides the highest accuracy without any preprocessing or data augmentation steps, the other CNNs benefit from certain preprocessing steps. For example, LeNet-5 improved most when training on augmented data with contrast stretching and pixel subtraction.

In the follow-up paper [RNM20] of [RNM17], the authors further compare the CNN architectures LeNet-5, AlexNet, Inception-V3, VGG-16 and ResNet-50 with similar preprocessing and data augmentation steps. Furthermore, they assess whether training

Inception-V3, VGG-13 and ResNet-50 from scratch or fine-tuning them gives better results. While their previous work [RNM17] determined the test accuracy on a hold out sample, the authors use 5-fold CV as a more robust experimental setup to estimate the test accuracy in [RNM20]. The authors find in their experiments that training Inception-V3, VGG-13 and ResNet-50 from scratch gave slightly better results than training via fine-tuning. Furthermore, LeNet-5 gave decent accuracy results and the other much deeper CNNs outperfomed LeNet-5 by at most a few accuracy percentage points. The authors also find that LeNet-5 performs best with data augmentation and no pre-processing steps, while the preferred approach according to [RNM17] (data augmentation with contrast stretching and pixel subtraction) gives the second best result.

5.4 Wasserstein Distances

This section introduces Wasserstein distances and Wasserstein barycenters. We employ the theory of Wasserstein distances when defining so-called Wasserstein distance models in Section 6.6.3 and when selecting features in Section 7.2.2. To simplify the following discussion, we only consider Wasserstein distances of arbitrary probability measures μ on $(\mathbb{R}^p, \|\cdot\|_2)$, i.e. in particular with finite $\|\cdot\|_2$ norm $\int_{\mathbb{R}^p} \|z\|_2^2 d\mu(z) < \infty$. The Wasserstein distance between such probability measures μ_1, μ_2 is defined as

$$W_2(\mu_1, \mu_2) := \left(\inf_{\gamma \in \Gamma(\mu_1, \mu_2)} \int_{\mathbb{R}^{2p}} \|x_1 - x_2\|_2^2 \, d\gamma(x_1, x_2) \right)^{1/2},$$

where $\Gamma(\mu_1, \mu_2)$ denotes the set of all measures γ on $\mathbb{R}^{2p}$ with marginals μ_1 and μ_2 on the first and second component, i.e. $\gamma(A \times \mathbb{R}^p) = \mu_1(A)$ and $\gamma(\mathbb{R}^p \times A) = \mu_2(A)$ for all measurable sets $A \subseteq \mathbb{R}^p$ [Kle13]. It can be shown that the Wasserstein distance W_2 gives the minimum cost of transporting masses between measures μ_1 and μ_2 and solves the so-called *optimal transport problem*, where costs c are given in our setup by $c(x_1, x_2) = \|x_1 - x_2\|_2^2$ [Vil09]. Hence, intuitively, W_2 gives the costs of moving a pile of sand given by the density of a probability measure μ_1 to holes in the ground given by the density of a probability measure μ_2 (and vice versa) [Mon81, Kan42]. W_2 therefore measures how far the two measures μ_1 and μ_2 are away from each other. Chapter 6 of [Vil09] provides more details and background information on Wasserstein distances and the connection to optimal transport.

Other popular methods to measure differences between two probability measures include the Kullback-Leibler divergence and the Jensen-Shannon distance. Contrary to the Kullback-Leibler divergence, the Wasserstein distance is a metric and in particular symmetric. It has shown to provide more intuitive results than the Kullback-Leibler divergence, see also Figure 5.1 for illustration. Compared to the Jensen-Shannon distance, the Wasserstein distance has proven to compare favorably especially if the support sets of μ_1, μ_2 are disjoint [FCG+21, KPMR18]. Due to numerical advances in calculating the Wasserstein distance [Cut13], Wasserstein distances have become a very popular

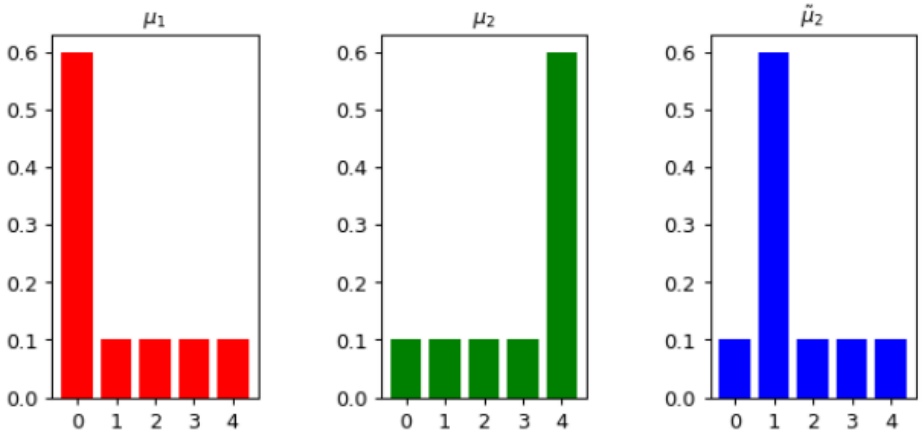


Figure 5.1: This figure illustrates that the Kullback-Leibler divergence does not always provide reasonable results compared to the Wasserstein distance. It makes sense to argue that μ_1 (left, red) is farther away from μ_2 (center, green) then from $\tilde{\mu}_2$ (right, blue) because the modes of μ_1 and $\tilde{\mu}_2$ are closer than the modes of μ_1 and μ_2. Indeed, the Wasserstein distance W_2 between μ_1 and μ_2 is greater than between μ_1 and $\tilde{\mu}_2$. However, the Kullback-Leibler divergence between μ_1 and μ_2 is the same as between μ_1 and $\tilde{\mu}_2$.

method in machine learning, most notably when training generative adversarial networks [ACB17].

We used the L^2 norm to define Wasserstein distances. Note that Wasserstein distances are also known as *earth mover distances* when using the L^1 norm.

5.4.1 Wasserstein Barycenters

Another favourable property of Wasserstein distances is that they allow for determining so-called barycenters for a set of measures $\{\mu_i\}$. A barycenter is itself a probability measure μ_{bary} that minimises the sum of its Wasserstein distances to each element in $\{\mu_i\}$ [CD14]. One can therefore think of μ_{bary} as an average mixture of all measures in $\{\mu_i\}$. Hence, μ_{bary} will also capture the analytical properties of the measures in $\{\mu_i\}$, see Figure 5.2 for illustration.

5.5 Summary

In this chapter, we discussed state of the art methods to classify the ALT status and fluorescence patterns of HEp-2 cells. We have learned that there are no FGA and IBA models in the literature yet to classify the ALT status of cells based on telPNA channel images. Instead, one currently classifies the ALT status based on rules-of-thumb, whole genome sequencing, C-circle assay or refined fluorescence microscopy imaging (ALT-FISH). For HEp-2 cell classification, there are various state of the art IBA models in the

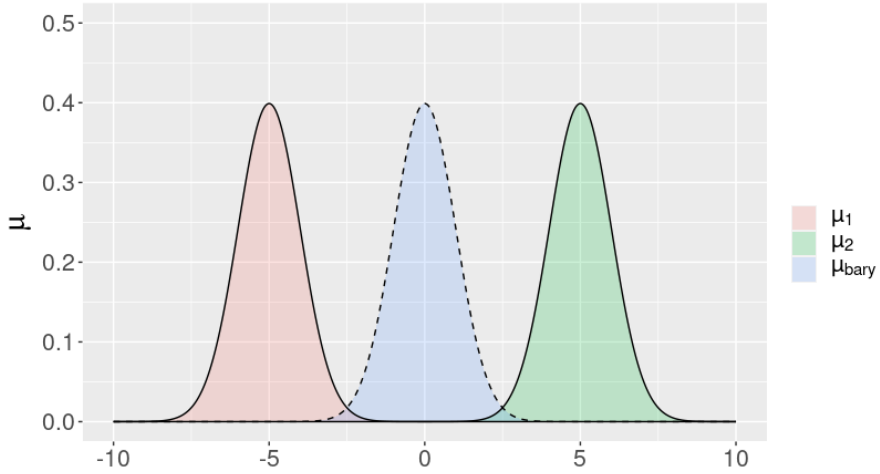


Figure 5.2: Two standardised normal distributions $\mu_1 = N(-5, 1), \mu_2 = N(5, 1)$ with mean -5 (left, red) and 5 (right, green) and their barycenter μ_{bary} (blue, dashed line). The barycenter itself is again normally distributed with center 0 and standard deviation of 1. Hence, μ_{bary} captures the analytic properties of μ_1, μ_2 and is an average mixture of them.

literature, including LeNet-5, Alexnet, ResNet-50 and GoogleNet. While LeNet-5 did not excel, we have learned that it still provided decent results when predicting the HEp-2 cell patterns. We also introduced Wasserstein distances and Wasserstein barycenters. We will use this theory when selecting variables in Section 7.2.2 and defining Wasserstein distance models in Section 6.6.3.

CHAPTER 6

Methodology

Estimating the ALT status based on two channel nuclear images introduced in Section 2.3 is a supervised classification problem with labelled targets (such as the ALT status). Chapters 3 and 4 delineate several state of the art methods to solve microscopy image classification using models of the FGA and IBA, respectively.

In this chapter, we outline which methodology we apply for the FGA and IBA of this thesis. Setting the methodology requires a precise definition of the data sets that we use to train, validate and test our models. Furthermore, we specify which methods we apply to extract features for the FGA and to classify images based on these features. Similarly, our methodology stipulates which IBA models we use and which methods we consider for training the neural networks and making these more robust.

We start off in Section 6.2 by discussing how we identify nuclei and spots in the DAPI and telPNA channels that we introduced in Section 2.3. Determining the nuclei and spots in the images is crucial for the FGA, but also the IBA depends on input images that are correctly cropped to identified nuclei. Afterwards, we highlight summary statistics and potential technical challenges of our data in Section 6.3. This information shall underscore how and why we decided on the methodology to address the technical challenges. Section 6.4 sets the general notation to define the two classification problems that we are interested in: ALT classification on nucleus level as well as on series level.

After the necessary introductory information of Sections 6.2-6.4, we are in a position to start outlining our methodology: Section 6.5 starts by providing details on our methodology to train, test and validate samples for the IBA and FGA based on the state of the art methods of Section 3.2. In Section 6.6, we discuss the methodology for the FGA and the chosen approaches for this thesis in detail. Likewise, in Section 6.7, we provide further details on how we apply the state of the art approaches of Section 4 in this master's thesis. In the last section, we formulate the two main research questions that we want to answer in this master's thesis.

Note that this chapter describes the general outline of our methodology. The preliminary experiments summarised in Chapters 7 and 8 give information on how and why we setup further methodological details of our samples (e.g. based on inclusion and exclusion criteria) and FGA and IBA models (e.g. tuning hyperparameters).

6.1 Denotation

In addition to the notation introduced in the previous sections, we denote the ALT+ share of a dilution series by $\rho \in \mathcal{AS}$ and denote by $\mathcal{AS} = 0., 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1$ the set of ALT+ shares.

6.2 Nucleus and Spot Segmentation

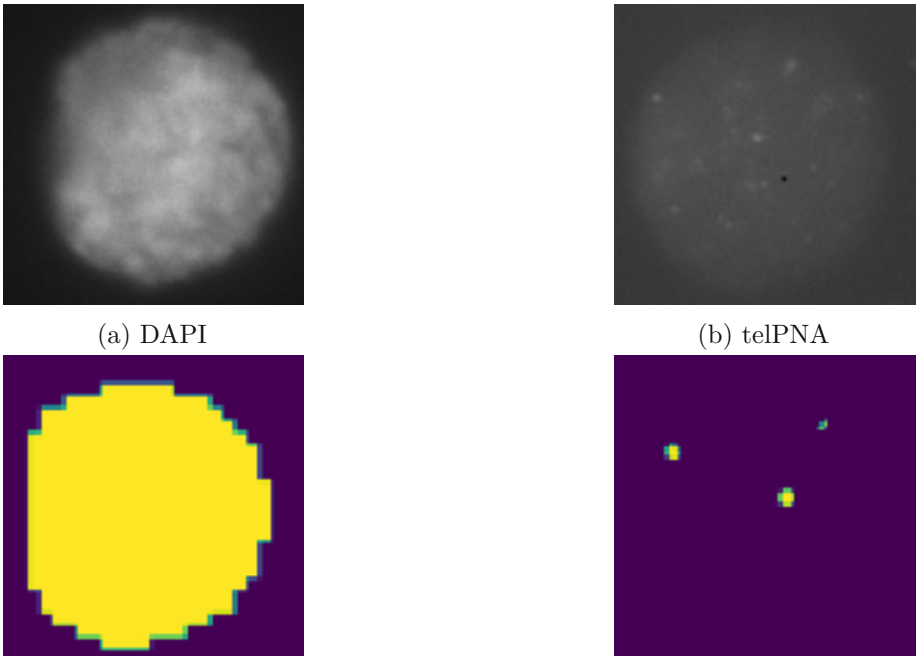
For both the FGA as well as IBA we have to identify nuclei in the fields of view of the dilution series. We need this information to correctly determine manual features per nucleus for the FGA and to properly crop the microscopy image as an input for the IBA, see Section 6.4. For the FGA, we also have to detect the telomere spots in the telPNA channel to generate additional manual features for each corresponding nucleus (e.g. standard deviation of spot distances within a nucleus, see Section 6.6.1).

The tumor biology group at the CCRI implemented deep neural networks (based on Cellpose [SWMP21], Mask-R-CNN [HGDG17] and U-Net [EVC+19]) to segment individual nuclei and their corresponding telomere spots in each pair of the DAPI and telPNA channel images [KFB+21]. The authors of [KFB+21] systematically compared their performance on complex fluorescence nuclear images of various types and of different imaging conditions (signal-to-noise ratio, sharpness, presence of damaged nuclei). Furthermore, they analysed how data augmentation and artificial data improves the accuracy in terms of specific performance measures (e.g. F1 score, precision, recall, see Section 4.4.1). Amongst others, [KFB+21] found that the neural network of Cellpose copes best with identifying nuclei in previously unseen imaging conditions. Furthermore, the authors found that U-Net, ResNet and Mask-R-CNN benefit most from artificial images.

We take these already implemented networks for granted and apply them for the purposes of this master's thesis to segment nuclei and spots. More specifically, we use the Cellpose-based network described in [KFB+21] to segment nuclei and a U-Net to identify telomere spots. Figure 6.1 shows the nucleus and spot segmentation masks for one nucleus of the series P12.

6.3 Motivation: Statistics on the Microscopy Image Data and Technical Challenges

In Table 6.1, basic statistics of the dilution series of Section 2.3.1 are summarised. We note that pure dilution series exhibit on average a smaller number of cells in each field



(a) DAPI (b) telPNA

(c) nucleus mask (d) spot mask

Figure 6.1: DAPI and telPNA channels as well as nucleus and spot masks of one nucleus of the ALT− series P12.

of view than actual dilution series. In the microscopy images, the nuclei in the pure dilution series also appear bigger than in the actual dilution series. To understand the latter observation, we note that we determine the nucleus sizes in Table 6.1 based on the nucleus segmentation masks of the DAPI channel via the neural networks of [KFB+21] (see Section 6.2). As cells in the actual dilution series are more numerous, they tend to overlap and occlude one another more easily. As a result, the masks segment occluded cells partially, which in turn underestimates the corresponding nucleus sizes.

While the microscopy imaging protocol is standardised for all microscopy image series (see Section 2.3.2), the statistics in Table 6.1 indicate that the microscopy images still vary across the series. More specifically, we can still expect a higher variance in the number and position of nuclei, the appearing size of nuclei and imaging quality across the series. This implies the following two technical challenges:

1. *Inaccurate nucleus segmentation.* The number of nuclei and their position varies, as they depend on how the prepared cell lines are applied on the carrier slides of the microscope. Nuclei appear smaller if several nuclei overlap, which is more likely in series with a high number of nuclei or in series where the cytospin preparation process led to more sticky cells. Similarly, if several nuclei are cramped together, it is hard to differentiate them for nucleus segmentation with the neural networks of [KFB+21].

dilution series name	mean cell count	mean nucleus size
ALT-C.P3∼A	34,1	10.138,3
ALT-C.P4∼A	28,9	10.650,3
ALT-C.P6	54,7	8.622,1
ALT-C.P7	35,5	7.549,6
ALT-C.P8	73,0	8.273,7
ALT-C.P9WH	87,9	6.584,7
ALT-C.P10	92,9	7.945,4
ALT-C.P11	88,1	7.815,3
ALT-C.P12	87,0	6.336,2
ALT-C.P13	93,2	6.526,2
ALT-C.PM1	182,0	4.277,4
ALT-C.PM3	202,5	5.886,7
ALT-C.PM4	212,2	5.541,3
ALT-C.PM5	199,7	5.950,6
ALT-C.PM9	44,0	6.085,8
ALT-C.PM12	134,9	5.622,8
ALT-C.PM14	109,1	5.757,5
ALT-C.PM15	120,6	6.161,1
ALT-C.PM16	133,3	6.022,2
ALT-C.PM17	36,6	5.852,0
ALT-C.PM22	190,5	5.375,4
ALT-C.PM23	196,4	5.247,9
ALT-C.PM24	182,9	5.780,8

Table 6.1: Sample statistics for dilution series considered in this master's thesis. The horizontal line in the middle of the table separates pure dilution series (above) from actual dilution series (below). Mean cell count refers to the average number of cells in a field of view of the dilution series. The colour gradient varies between green (small mean cell count) and red (greater mean cell count). Mean nucleus size gives the average size of a cell's nucleus in a field of view via a mesh surface measure [vGFP+17]. The colour gradient varies from green (greater mean nucleus size) over yellow (average mean nucleus size) to red (smaller mean nucleus size).

Hence, occluded or cramped nuclei lead to inaccurate nucleus segmentation. Based on Table 6.1, we see that this problem is specific for actual dilution series and not that relevant for pure dilution series. Furthermore, one has to note that generating fields of view of the DAPI and telPNA channel images leads to cropped nuclei at the border of the field of view. Inevitably, the segmentation masks of these nuclei will also be cropped and therefore be inaccurate.

2. *Varying image quality and fluorescence staining.* The imaging quality depends on how the nuclei are attached to the carrier glass, as some images show air bubbles or blurry content. Similarly, images at the edge of the carrier glass can

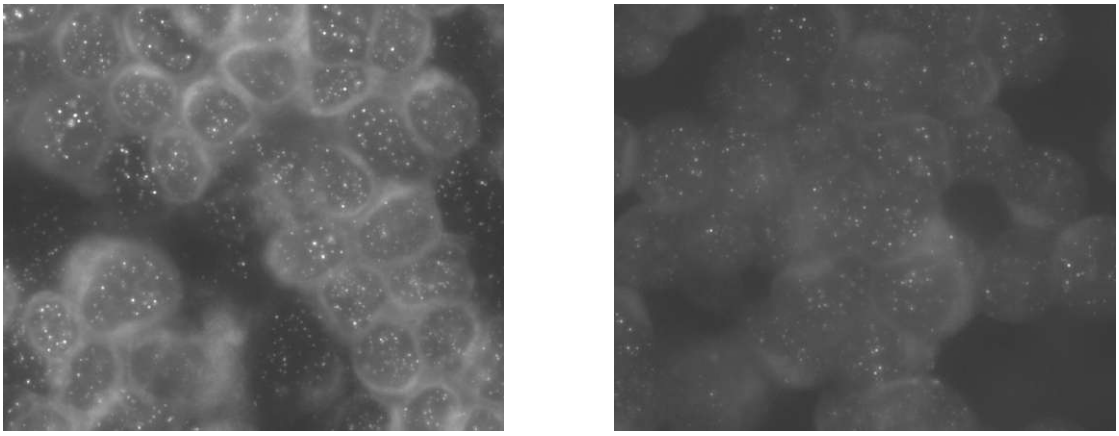


Figure 6.2: Cropped view of telPNA channels for dilution series PM22 (left) and P9WH (right). The nuclei in PM22 show bright stripes around the nucleus border due to cytoplasmic staining described in Challenge 2. The nuclei in P9WH do not feature these stripes.

be overexposed. Furthermore, the immunofluorescence staining may have led to unwanted imaging effects such as bright stripes around the nucleus border caused by unspecific cytoplasmic staining, see Figure 6.2. Experts of the CCRI assessed that the dilution series P12, PM1, PM14, PM15, PM17, PM22, PM24 particularly show staining issues in the nuclear background.

Our methodology for both the FGA and IBA that we introduce in Sections 6.6, 6.7 have to address the afore-mentioned two technical challenges. Challenge 1. is a specific problem for the actual dilution series that we address in Section 7.3 via a post-processing model of the nucleus segmentation. This model aims at detecting inaccurately segmented nuclei based on geometric properties of the segmentation masks (e.g. size, convexity). As discussed in Section 7.3, we will exclude these nuclei from our analyses.

Challenge 2. impacts both the FGA and IBA in two ways: first, Challenge 2. may itself impede nucleus segmentation in the DAPI channel as well as spot segmentation in the telPNA channel due to blurry or overexposed content. We address inaccurate nucleus and spot segmentation via the post-processing model of the nucleus segmentation mentioned above as well as by imposing a minimum number of visible spots per nucleus to consider nuclei in our analyses (see Section 7.2.1). Second, Challenge 2. may unduly influence the extracted features of the FGA and IBA. To that end, we aim at generating features that are *stable* in the sense that they are less affected by varying image quality. Sections 6.6, 6.7, 7.2.2 and 8.3 will provide further details.

6.4 Methodology for Image Classification of ALT Microscopy Images

After applying the nucleus segmentation algorithm of Section 6.2, we can conceive our collection of image data in Section 2.3.1 as images $(I_k)_k \subseteq \mathcal{I}$, where $\mathcal{I}$ includes the multi-channel fields of view cropped to each individual nucleus. Therefore, every image I_k refers to one nucleus. Each channel image is a two-dimensional array of pixel values between 0 (black) and 255 (white) and I_k can therefore be seen as a three-dimensional array. There are four potential channels to consider: the telPNA channel, the DAPI channel, the nucleus segmentation masks and the spot segementation masks, see Figure 6.1. In this master's thesis, we want to solve two kinds of ALT classification problems: first, ALT classification on nucleus level and, second, ALT classification on series level. The following two sections provide further details.

6.4.1 ALT Classification on Nucleus Level

In ALT classification on nucleus level, we assume that all images $(I_k)_{k=1}^K$ have corresponding targets $(g_k)_{k=1}^K \subseteq \mathcal{G}$ available and we want to find a model $M : \mathcal{I} \to \mathcal{G}$ that makes a good prediction $\hat{g}_k$ of the target g_k, i.e. $M(I_k) = \hat{g}_k$. This is a binary classification problem with $\mathcal{G} = \{\text{ALT}+, \text{ALT}-\}$. Obviously, knowing the ALT status g_k for an image I_k of a nucleus requires I_k to originate from a pure dilution series (e.g. P10), see Table 2.1. For an actual dilution series (e.g. PM12), we don't know the ALT status for each individual nucleus, but we only know the overall share of ALT+ nuclei in the dilution series (25%, see Table 2.1). Hence, we can only use images $(I_k)_k$ of pure dilution series to train models that classify the ALT status of a nucleus, see Section 3.2. Still, we can employ already trained ALT classification models on nucleus level to predict the share of ALT+ nuclei in an actual dilution series. The following section provides further details.

6.4.2 ALT Classification on Series Level

For ALT classification on series level, we have a sequence $(I_k)_{k=1}^K$ of images and we want to find a model $M : \Pi_{k=1}^{\infty}\mathcal{I} \to \mathcal{G}$ that predicts the *share* $\pi \in \mathcal{G}$ of ALT+ cells among the sequence $(I_k)_{k=1}^K$. Thus, if we assume that we have labels g_k of all images I_k available and set $g_k = 0$ for ALT− cells and $g_k = 1$ for ALT+ cells, we want to predict the ALT+ share $\pi = \frac{1}{K}\sum_{k=1}^K g_k$. In practice, the sequence $(I_k)_{k=1}^K$ originates from the same dilution series (e.g. PM.15), for which we do *not necessarily* have nucleus labels $(g_k)_{k=1}^K$ available but we know the actual ALT+ share (e.g. $\pi = 1\%$, see Table 2.1). This is a multi-class classification problem with $\pi \in \mathcal{G} = \{0, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1\}$, i.e. π refers to the corresponding *ALT+ class* of the dilution series.

To find the ALT+ share of a sequence $(I_k)_{k=1}^K$, we can use two approaches: first, we can re-use already-built ALT classification models on nucleus level or, second, we build new models that incorporate the feature information of all nuclei in $(I_k)_{k=1}^K$ as direct inputs. For the first approach, we simply classify nuclei of $(I_k)_{k=1}^K$ separately to predict ALT

status $(\hat{g}_k)_{k=1}^K$ and then use $\hat{\pi} = \frac{1}{K}\sum_{k=1}^K \hat{g}_k$ to predict the ALT+ share. For the second approach, we will use only FGA models in this thesis, but not IBA models. Section 6.6.3 discusses the second approach in more detail.

6.5 Methodology to Train, Validate and Test Models

Following the state of the art setup to generate samples for training, validating and testing models in Section 3.2, we split the data $\mathcal{I}$ of the dilution series in this master's thesis into the following two to three disjoint samples

1. Training sample $\mathcal{T}$: we use observations of the training sample to select inputs and train model parameters by optimising the objective functions of the models.
2. Testing sample $\mathcal{T}_{test}$: After training the models, we can assess the model performance on the testing sample to simulate how well the models perform on previously unseen data.
3. Validation sample $\mathcal{V}$: During training, we may use the validation sample to iteratively check up on any over- or underfitting behaviour.

In the IBA, we use all three samples $\mathcal{T}, \mathcal{T}_{test}, \mathcal{V}$ to develop our models. In the FGA, we mainly use $\mathcal{T}$ and $\mathcal{T}_{test}$. However, in some cases, we also employ cross-validation and split $\mathcal{T}$ into various training and validation folds.

As discussed in Section 6.4.1, the samples $\mathcal{T}, \mathcal{T}_{test}, \mathcal{V}$ for ALT classification on nucleus level have to consist of pure dilution series, as we have to know the ALT status $g_k \in \mathcal{G} = \{\text{ALT+}, \text{ALT}-\}$ of each nuclear image I_k. For ALT classification on series level, we are in principle free to use both pure as well as actual dilution series for training and testing. However, to compare all models of ALT classification on nucleus and series level in the same data scientific setup, we choose models for ALT classification on series level that allow for training on pure dilution series only. Hence, in any case, all nuclear images I_k in the training, testing and validation samples $\mathcal{T}, \mathcal{T}_{test}, \mathcal{V}$ have labels $g_k \in \mathcal{G} = \{\text{ALT+}, \text{ALT}-\}$. For these samples, we use nuclei of the pure dilution series P3∼A, P6, P8, P9WH, P10, P11, P12, P13. As discussed below in more detail, we will not consider the ALT− nuclei of the pure dilution series P4∼A and P7 in $\mathcal{T}, \mathcal{T}_{test}, \mathcal{V}$.

We also have to ensure that the samples $\mathcal{T}, \mathcal{T}_{test}, \mathcal{V}$ are consistent for all models, irrespective of whether they belong to the FGA, IBA or aim at solving ALT classification on nucleus or series level. As we only use $\mathcal{V}$ in the IBA and employ CV on $\mathcal{T}$ for the FGA, we implement the following approach:

- We split $\mathcal{I}$ into disjoint sets $\mathcal{T}$ for training and $\mathcal{T}_{test}$ for testing, i.e. $\mathcal{I} = \mathcal{T} \cup \mathcal{T}_{test}$, $\mathcal{T} \cap \mathcal{T}_{test} = \emptyset$.
- We use the whole sample $\mathcal{T}$ for training models of the FGA.

- We split $\mathcal{T}$ according to an 80%:20% split into disjoint samples $\mathcal{T}_{IBA}$ and $\mathcal{V}$ that we use for training and validating models of the IBA. Hence, $\mathcal{I} = \mathcal{T} \cup \mathcal{T}_{test} = \mathcal{T}_{IBA} \cup \mathcal{V} \cup \mathcal{T}_{test}$.

We use the same testing sample $\mathcal{T}_{test}$ for all approaches. Furthermore, we ensure that the cross-validation split of $\mathcal{T}$ in the FGA (5-fold CV) corresponds to the split of $\mathcal{T}$ into $\mathcal{T}_{IBA}$ and $\mathcal{V}$ in the IBA (80%:20% split). Note that the FGA uses less channels than the IBA, see Sections 6.6.1 and 6.7. Hence, $\mathcal{I}$ for FGA includes lower-dimensional arrays than $\mathcal{I}$ for IBA. Still, to simplify matters, we do not reflect these differences in our notation.

As stated above, the training, testing and validation samples of the FGA and IBA may only contain nuclei of pure dilution series, for which we have nucleus labels $g_k \in \mathcal{G} = \{\text{ALT+}, \text{ALT}-\}$ available. To assess how well the already trained models predict the ALT+ ratio on actual dilution series, we denote by $\widetilde{\mathcal{D}} = \{\mathcal{D}_{PM1}, \mathcal{D}_{PM3}, \ldots, \mathcal{D}_{PM24}\}$ the family of sets $\mathcal{D}_{PM\cdot}$, which contain nuclei of actual dilution series, see Table 2.1 for the PM encoding. For each $\mathcal{D}_{PM\cdot}$, we know the actual ALT+ ratio $\rho_{PM\cdot} \in \{0.01, 0.05, \ldots, 0.75\}$. Furthermore, we also consider in $\mathcal{D} = \{\mathcal{D}_{P4\sim A}, \mathcal{D}_{P7}\} \cup \widetilde{\mathcal{D}}$ the family of actual dilution series as well as the pure dilution series P4∼A and P7, see Table 2.1. This is because we want to use the ALT− nuclei of the pure dilution series P4∼A and P7 in $\mathcal{D}$ to test the model performance on ALT− cells by determining false positive rates. We exclude P4∼A and P7 from $\mathcal{T}, \mathcal{T}_{IBA}, \mathcal{T}_{test}, \mathcal{V}$, as these dilution series feature poorer image qualities according to experts of the CCRI.

Note that we will detail our setup of $\mathcal{T}, \mathcal{T}_{IBA}, \mathcal{T}_{test}, \mathcal{V}$ in Chapter 7 based on our findings of preliminary experiments. There, we will set more specific inclusion and exclusion criteria for the samples.

6.6 Methodology for FGA

In this section, we will treat the FGA in greater detail. We start off in Section 6.6.1 by discussing the features that we generate for the FGA to classify the ALT status as well as for the post-segmentation model to correct for wrongly segmented nuclei in actual dilution series, see Section 6.3. Section 6.6.2 provides an overview of relevant ALT classification models on nucleus level, while Section 6.6.3 presents the proposed so-called Wasserstein distance models that we will use for ALT classification on series level.

6.6.1 Feature Generation for ALT Classification and Postsegmentation Processing

As discussed in Section 3.3, the FGA requires to manually generate features $x_k \in \mathbb{R}^p$ for each image I_k as inputs of the FGA models. If necessary, we can normalise or standardise the brightness intensity curves of each image I_k before we extract features $x_k \in \mathbb{R}^p$. Section 7.2.3 discusses this topic in more detail for our setting of ALT classification.

It is important to note that in the FGA we only consider the telPNA channel as well as the nucleus and spot segmentation masks of I_k to generate features. This is because there is no biological reasoning why the DAPI channel explains the ALT status. Hence, for the FGA, we only use the DAPI channel to generate the nucleus masks and to identify nuclei.

To determine features for each image I_k, we use two approaches: first, we define features based on biological reasoning. Second, we use features of the pyradiomics library [vGFP+17] as discussed in Section 3.3. In total, this gives us 87 features that we can use for FGA modelling. The following paragraphs discuss each approach separately.

Manually Defined Features

Based on biological reasoning, we identified 17 feature functions to potentially explain the ALT status for a nucleus, see Section 3.3 for the definition of a feature function. The following three classes group these features:

- *telPNA signal intensity of spots*: by restricting the telPNA channel to spots of a nucleus identified by the spot and nucleus segmentation masks, we can determine a distribution of signal intensities. For this distribution, we calculate separate features based on summary statistics such as the mean or maximum spot intensity as well as quantiles of the spot intensities. Furthermore, we determine the intensity of the largest spot of a nucleus. This group comprises 5 features.
- *telPNA signal intensity of the whole nucleus*: by restricting the telPNA channel to the relevant nucleus, we can determine summary statistics of the signal intensities, such as the mean, maximum, standard deviation or certain quantiles. This group comprises 7 features.
- *analytical and geometric properties of spots*: by considering the spot segmentation mask, we can determine analytical and geometric properties of the identified spots in each nucleus, such as the average spot size, the standard deviation of their size, number of spots per nucleus, or the size of the brightest spot. This group comprises 5 features.

As spot sizes and the signal intensity of the telPNA channel are important determinants for the ALT status, see Chapter 2, the above-mentioned features are able to biologically explain the ALT status.

Pyradiomics Features

As discussed in Section 3.3, pyradiomics is a python library to generate radiomics features for medical imaging [vGFP+17]. In pyradiomics, we consider first-order features as well as higher-order features based on gray level matrices. In total, we extract 70 pyradiomics features from the telPNA channel, see Table 6.2. First-order features consider properties

of the telPNA signal intensity distribution such as kurtosis or skewness. Higher-order features use the gray level intensities of the telPNA channel to determine pairs of co-occuring gray level intensities (in gray level co-occurrence matrices GLCM), run lengths of gray level intensities (in gray level run length matrices GLCM) as well as zones of gray level intensities (in gray level size zone matrices GLSZM).

Geometric Features for Post-Segmentation Processing

In addition to the afore-mentioned features for ALT classification, we also generate features that we use in a post-segmentation model to correct for wrongly segmented nuclei in actual dilution series, see Section 6.3. These 16 features capture shape properties of the nuclei (e.g. size, convexity, elongation, roundness) that we generate using the pyradiomics package. Furthermore, leveraging on nucleus segmentation masks, we construct features that provide for each nucleus the average and smallest distance to other nuclei and a binary indication whether the nucleus is at the border of a field of view or touches other nuclei. More specifically, we slightly enlarge each segmentation mask by binarily dilating it 6 times and then intersect the dilated mask with the original segmentation masks of all other nuclei to identify touching nuclei. To that end, we use the `binary_dilation` method of `scipy` [VGO+20]. Slightly enlarging segmentation masks is necessary, as all segmentation masks are disjoint.

Methodology to Select Robust Features

For the FGA, we extract 87 manually defined and pyradiomics features from the telPNA channel as discussed above. To address Challenge 2. of Section 6.3 (varying imaging quality), we first have to assess whether the extracted features of nuclei are valid and reliable. In particular, we have to ensure that they are not based on image artifacts. For that reason, we will assess whether the extracted features are robust and stable across pure and actual dilution series. We will use Wasserstein barycenters as discussed in Section 6.6.3 to determine how robust and stable features are. Section 7.2.2 provides the full algorithm that we have developed to select robust variables. We outline our method in Section 7.2.2 as it depends on and is better motivated by previous findings of our preliminary experiments in Chapter 7.

6.6.2 FGA Classification Models on Nucleus Level

For FGA classification models on nucleus level, we assume that we have features $x_k \in \mathbb{R}^p$ for each image $I_k \in \mathcal{I}, k = 1, \ldots, K$ of a nucleus according to Section 6.6.1 given. We will also refer to FGA classification models on nucleus level as *nuclear FGA models*.

For ALT classification on nucleus level, we want to predict the ALT status for the nucleus that is depicted in the image I_k. The corresponding targets are binary labels $g_k \in \mathcal{G} = \{\text{ALT}+, \text{ALT}-\}$. Thus, as discussed in Section 6.4.1, the training sample $\mathcal{T}$ has to consist of pure dilution series.

A corresponding nuclear FGA model $M : \mathbb{R}^p \to \mathcal{G}$ operates on the feature space $x_k \in \mathbb{R}^p$ of images I_k to classify the ALT status. In this thesis, we want to use logistic regressions, support vector machines, random forests and gradient boosting, which we discussed in Chapter 3. We train these classification models on previously selected robust features as outlined in Sections 3.4, 6.6.1 as well as Section 7.2.2 below and tune the relevant hyperparameters (see Section 7.2.4). We use accuracy as the main performance measure that we want to maximise with our models, see Section 4.4.1 for details.

6.6.3 FGA Classification Models on Series Level

For FGA classification models on series level, we again assume that we have features $x_k \in \mathbb{R}^p$ for each image $I_k \in \mathcal{I}$ of a nucleus. Given a sequence $(I_k)_{k=1}^K$ of nuclear images, we want to predict the *share* of ALT+ cells among the sequence $(I_k)_{k=1}^K$, i.e. $\frac{1}{K}\sum_{k=1}^K g_k$. As discussed in Section 3.2, we assume that all images $I_k \in \mathcal{T}$ originate from pure dilution series with available targets $g_k \in \{\text{ALT+}, \text{ALT}-\}$. Still, after training the classification model on series level, the sequence $(I_k)_{k=1}^K$ originates from the same actual dilution series (e.g. PM.15), for which we have the actual ALT+ share available (e.g. 1%, see Table 2.1).

As discussed in Section 6.4.1, there are two approaches to find the ALT+ share of a given sequence $(I_k)_{k=1}^K$: first, we can re-use nuclear FGA models or, second, we build new models that incorporate the feature information of all nuclei as direct inputs. In this section, we want to focus on the second approach by using a so-called Wasserstein distance model that employs Wasserstein distances as discussed in Section 5.4. We will also refer to this model as *serial FGA model*.

Wasserstein Distance Models

To use Wasserstein distances for ALT classification on series level, we have developed a tailor-made approach as part of this master's thesis. The following paragraphs discuss this new method thoroughly and the next subsection provides further details on the variable selection.

For Wasserstein distance models, we assume that we have q-many feature functions $F_1, \dots, F_q$ given and generate for each image I_k of a nucleus corresponding features $x_1, \dots, x_q$. Based on this q-dimensional feature vector $(x_1, \dots, x_q)$ of each image I_k, we can therefore determine a q-dimensional joint distribution μ over all images $(I_k)_k$. Furthermore, as $\mathcal{T}$ consists of only pure dilution series, see Section 3.2, we can split the images $(I_k)_k$ of our training data $\mathcal{T}$ into ALT+ and ALT− nuclei. Hence, we can generate separate q-dimensional probability measures μ_+ and μ_- of the features for ALT+ and ALT− nuclei, respectively.

Assuming that our training data $\mathcal{T}$ is representative of the whole population $\mathcal{I}$, we can use μ_+ and μ_- to construct the q-dimensional feature distribution of all dilution series via convex combinations of μ_+ and μ_-. More specifically, under the aforementioned assumptions, an actual dilution series with a given ALT+ share of $\pi \in \mathcal{AS} =$

$\{0, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1\}$ has to exhibit a q-dimensional feature distribution of

$$\mu_\pi = \pi\mu_+ + (1-\pi)\mu_-.$$

If we are confronted with a dilution series of nucleus images $(\tilde{I}_k)_k$ and unknown ALT+ share ρ, we can generate the q-dimensional feature distribution μ_ρ of $(\tilde{I}_k)_k$ and determine the minimal Wasserstein distance between μ_ρ and μ_π for $\pi \in \mathcal{AS}$

$$\hat{\rho} = arg\min_{\pi\in\mathcal{AS}} W_2(\mu_\rho, \mu_\pi). \tag{6.1}$$

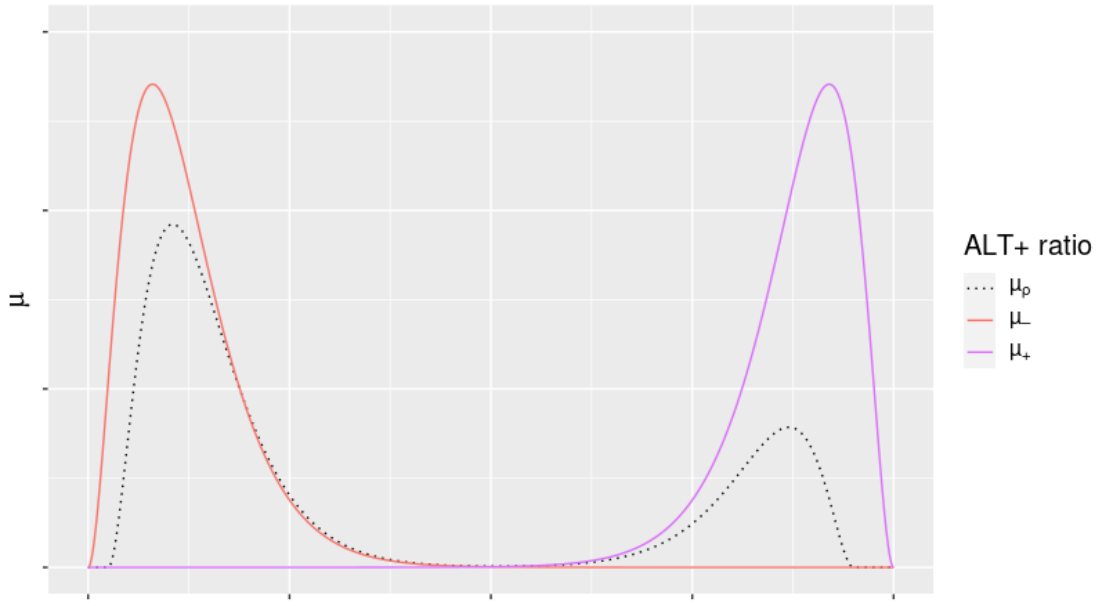
Hence, ratio $\hat{\rho}$ of minimal Wasserstein distance is our Wasserstein distance model estimate for the ALT+ share in the dilution series $(\tilde{I}_k)_k$. Figure 6.3 illustrates the Wasserstein distance model for hypothetical 1-dimenstional feature distributions.

To summarise, a Wasserstein distance model is based on q-many feature functions $F_1, \ldots, F_q$ and corresponding q-dimensional joint distributions μ_+ and μ_- of these features for ALT+ and ALT− nuclei in the training sample $\mathcal{T}$. During "training", we generate convex combinations μ_π of μ_+ and μ_- with $\pi \in \mathcal{AS}$. Wasserstein distance models predict via (6.1) by minimising Wasserstein distances to μ_π. They are therefore so-called "lazy learners" similar to K-nearest neighbour models [JWHT13]. Indeed, conceptually, one can think of Wasserstein distance models as 1-nearest neighbour models when considering μ_π with $\pi \in \mathcal{AS}$ as neighbours and measuring distances with W_2.

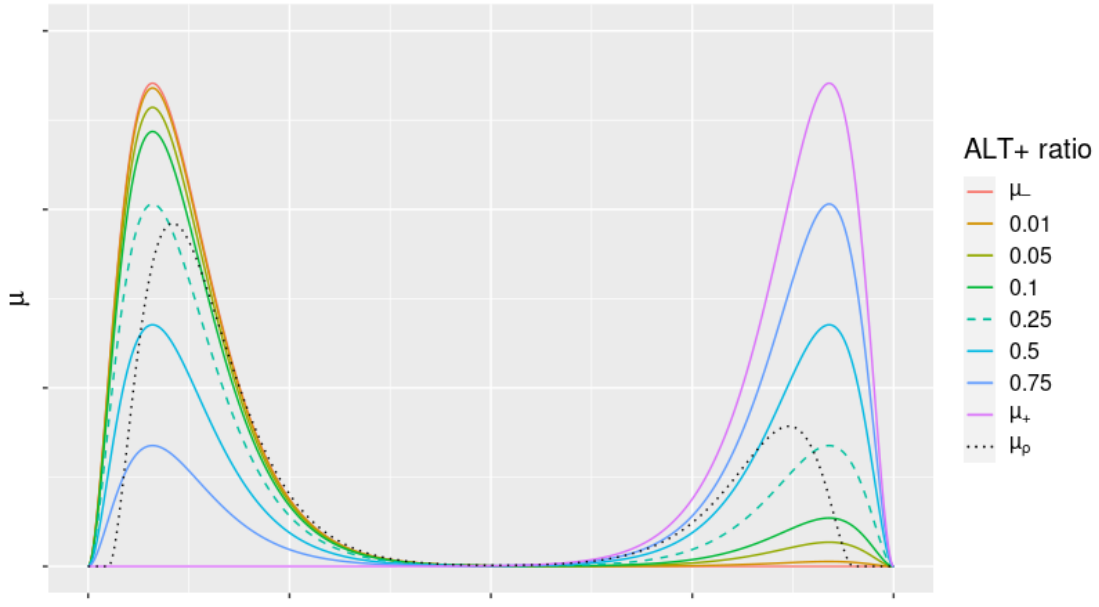
Variable Selection for Wasserstein Distance Models

It still remains finding the right feature functions $F_1, \ldots, F_q$ for our Wasserstein distance model. As the data necessary to sample the distributions μ_+, μ_- grows exponentially with q, we will select only two feature functions F_1, F_2. Higher-dimensional distributions may become intractable based on the available amount of data (curse of dimensionality). To find the best feature functions F_1, F_2, we use 5-fold CV in the following way:

1. Choose two feature functions F_1, F_2.
2. On $\mathcal{T}$, select four training folds and one test fold based on 5-fold CV.
   a) On the training folds, determine the 2-dimensional feature distributions $\mu_+^{train}, \mu_-^{train}$ of ALT+ and ALT− nuclei, respectively. Setting a finer ALT+ share grid $\mathcal{AS}_{fine} = \{0.01, 0.02, \ldots, 0.99, 1\}$, define
   $$M_{train} = \left\{\mu_\pi^{train} = \pi\mu_+^{train} + (1-\pi)\mu_-^{train} | \pi \in \mathcal{AS}_{fine}\right\}.$$
   b) On the test fold, determine the 2-dimensional feature distributions $\mu_+^{test}, \mu_-^{test}$ of ALT+ and ALT− nuclei, respectively. We then set
   $$M_{test} = \left\{\mu_\pi^{test} = \pi\mu_+^{test} + (1-\pi)\mu_-^{test} | \pi \in \mathcal{AS}\right\}.$$



(a) μ_+ and μ_- of ALT+ and ALT−nuclei and μ_ρ of unknown ALT+ share ρ.



(b) Convex combinations $\mu_\pi = \pi\mu_+ + (1-\pi)\mu_-$ of μ_+, μ_- and μ_ρ.

Figure 6.3: Subfigure 6.3a shows hypothetical 1-dimensional feature distributions μ_+ (purple, right) and μ_- (red, left) of ALT+ and ALT−nuclei as well as the corresponding 1-dimensional feature distribution μ_ρ (black, dotted) of unknown ALT+ share ρ. In Subfigure 6.3b, we see all convex combinations μ_π of μ_+ and μ_- with $\pi \in \mathcal{AS}$. The cyan dashed line of ALT+ ratio 0.25 denotes the convex combination $\mu_{0.25}$ with smallest W_2 distance to μ_ρ. Hence, $\hat{\rho} = 0.25$ is the Wasserstein distance model estimate for the ALT+ share ρ of μ_ρ.

c) Determine for each $\mu_\pi^{test} \in M_{test},\ \pi \in \mathcal{AS}$, the distribution $\mu_{\hat{\pi}}^{train} \in M_{train},\ \hat{\pi} \in \mathcal{AS}_{fine}$ with smallest W_2 distance to μ_π^{test} and calculate the accuracy $a = \frac{1}{\|\mathcal{AS}\|} \sum_{\pi \in \mathcal{AS}} \mathbb{1}_{\pi=\hat{\pi}}$. $\hat{\pi}$ is therefore the grid ratio of $\mathcal{AS}_{fine}$ for which $\mu_{\hat{\pi}}^{train}$ provides the smallest distance to μ_π^{test}.

3. Repeat Step 1. for the other four possible permutations of training and test folds on $\mathcal{T}$. We therefore find accuracy estimates $a_1, \ldots, a_5$ for each of the five test folds. Determine the average accuracy $\bar{a} = \frac{1}{5} \sum_i^5 a_i$ and its standard deviation s_a.

4. If we repeat Steps 1.-3. for all possible choices of feature functions F_1, F_2, we find corresponding mean accuracy values and standard deviations $\bar{a}_{F_1,F_2}, s_{F_1,F_2}$. We can determine the pair of feature functions F_1^{max}, F_2^{max} with maximum average accuracy $\bar{a}_{max} = \bar{a}_{F_1^{max},F_2^{max}}$ and corresponding standard deviation $s_{max} = s_{F_1^{max},F_2^{max}}$. We note that all other pairs F_1, F_2 with

$$\bar{a}_{max} - s_{max} \leq \bar{a}_{F_1,F_2} \leq \bar{a}_{max} \tag{6.2}$$

show very similar accuracy values, as their mean accuracy values lie within one standard deviation of the maximum average accuracy. To find a pair of features F_1, F_2 that provide high accuracy values on all folds, we therefore select the pair of features F_1, F_2 which satisfies (6.2) and exhibits the smallest standard deviation s_{F_1,F_2} among all pairs for which (6.2) holds.

In this thesis, we apply above's algorithm to select features for the Wasserstein distance model. We select these features among the previously identified robust features according to Section 6.6.1.

6.7 Methodology for IBA

In the IBA, we train CNNs and residual networks on samples $\mathcal{T}_{IBA}, \mathcal{V}, \mathcal{T}_{test}$ for binary ALT classification on nucleus level, see Section 6.5 for the sample definitions. We then re-use the trained IBA models for ALT classification on series level following the idea of Section 6.4.2.

In the following, we discuss how we use and amend state of the art methods introduced in Chapter 4 to classify the ALT status in the IBA as part of this master's thesis. While to the best of our knowledge there is no active research on classifying the ALT status via deep learning, choosing an appropriate CNN architecture for the IBA is crucial. In this master's thesis, we will focus on CNN architectures that have proven to be successful in classifying HEp-2 cell patterns.

Due to the following two reasons we consider that CNN architectures are promising candidates for classifying the ALT status if they also manage to predict the HEp-2 cell patterns well: first, certain HEp-2 cell patterns exhibit bright spots of varying sizes comparable to ultra-bright spots in the telPNA channel images of ALT+ nuclei.

To distinguish between the corresponding HEp-2 classes, the CNN architecture has to take into account the density, size, brightness and position of spots, which is similar to distinguishing between ALT+ and ALT− nuclei. Second, immunofluorescence microscopy images of HEp-2 cells also require attaching cells to a carrier glass similar to the images of this master's thesis described in Section 2.3. In particular, the position of a cell on the carrier glass to other cells is irrelevant (as opposed to, say, tissue sections).

For the IBA, we therefore use LeNet-5 [LBBH98] as a starting point for developing a custom CNN, because LeNet-5 has proven to give satisfactory results when predicting HEp-2 cell patterns [GWZZ16, RNM17, RNM20]. The following five amendments of LeNet-5 are necessary to find a custom CNN for the IBA: first, the target of the CNN has to be binary to match ALT classification on nucleus level, see Section 6.4.1. Second, as the input images are larger than in LeNet-5, which aimed at recognising handwritten letters, we have to amend the kernel sizes in the convolutional layers by following the approach of [GWZZ16]. Similarly, in line with the input image sizes, we also consider increasing the sequence of convolutional and pooling layers based on the approach of [GWZZ16]. Third, we exchange average-pooling layers in LeNet-5 by max-pooling layers, which have typically become more popular in microscopy image analysis [XXS+17]. Fourth, modern CNNs and ResNets usually stack two convolutional layers before applying pooling layers [HZRS16]. For that reason, we also implement double convolutional layers in our custom CNN. Fifth, we will use ReLU activation functions instead of arctanh that is used in LeNet-5.

Note that for our custom CNN we use single-channel telPNA images of nuclei. This is because the telPNA channel is known to biologically explain the ALT status, see Section 6.6.1. We scale all nuclear images to ensure that $I_k \in \mathcal{I}$ are of pixel size 224×224. Figure 6.4 provides an overview of our custom CNN. For easier reference, we denote in the following discussion our own custom CNN by *MyNet*.

To address the challenge of varying imaging quality, we consider preprocessing nuclear images (min-max normalisation) and apply image augmentation techniques. More specifically, we consider the following data augmentation techniques:

- *Random blurring*: blurs a nuclear image with a probability of 20% or otherwise leaves it unchanged. Implemented using `RandomAdjustSharpness` in `torch.transforms`.
- *Random sharpness*: sharpens a nuclear image with a probability of 20% or otherwise leaves it unchanged. Implemented using `RandomAdjustSharpness` in `torch.transforms`.
- *Random halo effect*: lightens up borders of a nucleus image with probability of 20% or otherwise leaves it unchanged. This data augmentation technique shall account for varying fluorescence staining as described in Challenge 2. of Section 6.3 and Figure 6.2. We implemented this method manually using nuclear segmentation masks.

first order	GLRLM
Energy	GrayLevelNonUniformity
TotalEnergy	GrayLevelNonUniformityNormalized
10Percentile	GrayLevelVariance
90Percentile	HighGrayLevelRunEmphasis
Entropy	LongRunEmphasis
InterquartileRange	LongRunHighGrayLevelEmphasis
Kurtosis	LongRunLowGrayLevelEmphasis
MeanAbsoluteDeviation	LowGrayLevelRunEmphasis
Median	RunLengthNonUniformity
Minimum	RunLengthNonUniformityNormalized
Range	ShortRunEmphasis
RobustMeanAbsoluteDeviation	ShortRunHighGrayLevelEmphasis
RootMeanSquared	ShortRunLowGrayLevelEmphasis
Skewness	RunEntropy
Uniformity	RunPercentage
	RunVariance
GLCM	**GLSZM**
Autocorrelation	GrayLevelNonUniformity
ClusterProminence	GrayLevelNonUniformityNormalized
ClusterShade	GrayLevelVariance
ClusterTendency	HighGrayLevelZoneEmphasis
Contrast	LargeAreaEmphasis
Correlation	LargeAreaHighGrayLevelEmphasis
DifferenceAverage	LargeAreaLowGrayLevelEmphasis
DifferenceEntropy	LowGrayLevelZoneEmphasis
DifferenceVariance	SizeZoneNonUniformity
Id	SizeZoneNonUniformityNormalized
Idm	SmallAreaEmphasis
Idmn	SmallAreaHighGrayLevelEmphasis
Idn	SmallAreaLowGrayLevelEmphasis
Imc1	ZoneEntropy
Imc2	ZonePercentage
InverseVariance	ZoneVariance
JointAverage	
JointEnergy	
JointEntropy	
MCC	
MaximumProbability	
SumEntropy	
SumSquares	

Table 6.2: List of 70 pyradiomics features extracted from the telPNA channel and grouped according to their first-order or higher-order feature type. See Section 3.3 and [vGFP+17] for a definition of each feature.

Figure 6.4: Custom CNN that we use in this thesis for ALT classification on nucleus level. Each box corresponds to one layer and denotes the number of input and output channels, as well as kernel, padding and stride sizes (if applicable). The networks starts by aggressively reducing the array dimension by a first convolutional kernel of size 7, stride 2 and a max pooling layer of size 2, stride 2. In return, the first convolutional layer extracts 32 feature maps. Afterwards, the CNN iterates blocks of double convolutional layers and max pooling layers that each decrease the array dimension by two and increase the number of feature maps by 2. Before applying the last average pooling layer, the array dimension decreases to 7×7. In the final layer, each feature map provides one of the 512 features. Note that the network does not show dropout or batch normalisation layers.

We refrain from using other classical image augmentation techniques such as random cropping and rotations, as the nuclear images are already the result of aleatoric cropping and rotations. This is because nuclei are randomly attached to the carrier glass of the microscope and are cropped according to field of views, see Section 2.3.2.

We use batch normalisation, dropouts and the regularised version AdamW of the Adam optimiser to make our custom CNN more robust. We use accuracy as the main performance measure that we want to maximise with MyNet, see Section 4.4.1 for details.

In addition to MyNet, we use a pretrained ResNet-50 for which we will finetune the classification backend on $\mathcal{T}_{IBA}$. Due to the computational complexity of training ResNets, we have decided to use a pretrained ResNet-50 instead of training it from scratch. We use the same preprocessing steps and data augmentation techniques as discussed above for the custom CNN. Since ResNets are trained on three-channel RGB images, we have to choose the input channels of I_k appropriately to apply our pretrained ResNet. In addition to the "red" telPNA channel, we will therefore also include the nuclear segmentation mask and DAPI as "green" and "blue" channels, respectively. Of course, increasing the number of input channels comes at the cost of limited biological explainability. We hazard these consequences to compare our custom CNN with another much deeper network. For easier reference, we will drop the suffix of ResNet-50 and only refer to ResNet in the following discussion of this master's thesis. Again, we use accuracy as the main performance measure that we want to maximise with ResNet, see Section 4.4.1 for details.

6.8 Master's Thesis Research Question

After having initially set up the involved samples in Section 3.2 and the selected methods in Sections 6.6 and 6.7, we are in a position to formulate the two main research questions that we want to answer in this master's thesis.

RQ1 Which model of the FGA and IBA introduced in Sections 6.6 and 6.7 predicts the ALT status of cells best?

RQ2 What are objective image-derived criteria to describe the ALT status of cells?

For the first research question RQ1, we determine good prediction quality on samples that we have not used when training the models. Therefore, we focus on how well the FGA and IBA models solve the problem of predicting the ALT+ status and ALT+ ratio on the training sample $\mathcal{T}_{test}$ and on the dilution series of $\mathcal{D}$. This problem includes both ALT classification on nucleus and series level, respectively. On the one hand, we use the accuracy metric introduced in Section 4.4.1 to determine the prediction quality on $\mathcal{T}_{test}$. While we can readily determine accuracy scores for nuclear FGA and IBA models based on their individual predictions of ALT classification on nucleus level, Section 9.2 gives further details on how we calculate a surrogate accuracy score for the Wasserstein distance model. On the other hand, we determine absolute differences between actual and

predicted ALT+ ratios and ALT+ classes to assess the prediction quality on the dilution series of $\mathcal{D}$. For example, if a model predicts an ALT+ rate of 25% for an actual ratio of 75%, the absolute difference in terms of ALT+ ratios is 50% and in terms of ALT+ classes it is 2 (i.e. two ALT+ classes off), see also Section 6.4.2. We note that our approach of answering research question RQ1 equally involves models for ALT classification on nucleus and series level. Related to question RQ1, we also want to find measures of determining how confident the models are when predicting ALT status and ALT+ rates, see the next section 6.8.1.

The second research question RQ2 is particularly important since the currently available rules of thumb of [HSH+11] for predicting the ALT+ status focus on whether so-called ultra-bright spots are available in the telPNA channel, or not, see Section 5.2. The FGA and IBA models aim at providing more objective rules that determine the ALT status with higher confidence using image-derived criteria that describe the ALT+ status even for cells that do not show ultra-bright foci. This will also help explaining why the models arrived at specific predictions and foster the users' confidence in the models' decisions. To answer RQ2, we focus on feature importance scores of nuclear FGA models, selected variables for the Wasserstein distance model and look at examples of feature maps for our own CNN MyNet. Sections 9.3 and 9.4 provide further details.

To avoid confusions, note that the following Chapters 7 and 8 include additional, more technical research questions that corroborate how we setup the FGA and IBA models of Sections 6.6 and 6.7. These questions are not to be mixed with RQ1 and RQ2 that are the main research questions which we answer in Chapter 9 of this master's thesis.

6.8.1 Prediction Confidence

When considering prediction results, we are also interested in the models' confidence when predicting the ALT status on nucleus or series level. Determining the prediction confidence for all FGA and IBA models relates to research question RQ1, as it allows us to judge the prediction quality. In this section, we want to discuss how to determine the prediction confidence for all FGA and IBA models.

We start with nuclear FGA models and IBA models, which all provide probability estimates for assigning the ALT status of a given nucleus. More specifically, logistic regressions and IBA models directly provide these estimates via expit link function and softmax function, see Sections 3.4 and 4.2.4. For support vector machines, one can estimate class probabilities in `sklearn` by using so-called Platt scaling [Pla99], which is partly based itself on logistic regressions [PVG+11]. For random forests, we can use the class assignments of each individual decision tree to estimate class probabilities [PVG+11]. Last, in gradient boosting, we can use the weighted sum of predictions in the individual boosting to find a probability estimate similar to the expit link function of logistic regressions, see Chapter 10 of [HTF09] for details.

Using the probability estimates for each of the afore-mentioned models, we can assess how confident the models are when assigning the ALT status of a nucleus. To that end,

we use the following bands for a probability estimate p of a given nucleus:

1. $p \in [0\%, 20\%]$: sure ALT− assignment.
2. $p \in (20\%, 80\%)$: unsure assignment.
3. $p \in [80\%, 100\%]$: sure ALT+ assignment.

For the Wasserstein distance model, we cannot use probabilities to assess the model's confidence when predicting the ALT+ class of a given dilution series. Instead, we consider the feature distribution μ_ρ of the dilution series and the Wasserstein distance $W_2(\mu_\rho, \mu_{\pi_1})$ to the closest class π_1 as well as the Wasserstein distance $W_2(\mu_\rho, \mu_{\pi_2})$ to the second closest class π_2. We can then determine the ratio

$$0 \leq \gamma = 1 - \frac{W_2(\mu_\rho, \mu_{\pi_1})}{W_2(\mu_\rho, \mu_{\pi_2})} \leq 1,$$

to which we refer as *confidence ratio*. If $W_2(\mu_\rho, \mu_{\pi_1})$ and $W_2(\mu_\rho, \mu_{\pi_2})$ are very close, we are less confident that π_1 is the correct class. In this case, γ is close to 0. Hence, we can use the following heuristic to assess how confident the Wasserstein distance model is when predicting the ALT+ class:

1. $\gamma \leq 0.2$: unsure assignment.
2. $\gamma > 0.2$: sure assignment.

6.9 Summary

In this chapter, we outline our chosen methodology for ALT classification. We discussed statistics on the microscopy data of this master's thesis to motivate certain methodological decision to address two associated technical challenges. The first technical challenge is about inaccurate nucleus segmentation in actual dilution series due to cramped nuclei in the images. The second technical challenge concerns varying image quality and fluorescence staining, which requires stable feature extraction both in FGA as well as IBA models.

We specified that ALT classification involves two kinds of classification: first, on nucleus level by predicting the ALT status for an individual cell, and, second, on series level by determining the share of ALT+ cells among an image of multiple nuclei.

We defined the training, testing and validation samples for our FGA and IBA models, as well as a sample of (mostly) actual dilution series $\mathcal{D}$ that we use to assess the model performances on previously unseen image series. To that end, we specified which dilution series of Section 2.3 we will use for the individual samples $\mathcal{T}, \mathcal{T}_{test}, \mathcal{T}_{IBA}, \mathcal{V}, \mathcal{D}$.

For the FGA, we provided details about the features that we extract for each nucleus using own definitions and `pyradiomics`. We stipulated to use penalised logistic regressions,

support vector machines, gradient boosting and random forests as FGA classification models on nucleus level. Afterwards, we define the Wasserstein distance model, which we use as FGA classification model on series level.

For the IBA, we use two main models: first, our own implementation of a CNN based on LeNet-5 (called MyNet), and, second, a pretrained ResNet-50 for which we finetune the last linear layer to serve our purposes. We based MyNet on LeNet-5 as it has proven to give satisfactory results when predicting HEp-2 cell patterns, which are similar to ALT staining patterns of the telPNA channel. We also specified data augmentation techniques and methods to combat overfitting that we want to consider when training the models in Section 8.

There are two main research questions that we intend to answer in this thesis for the FGA and IBA models: first, which of the FGA and IBA models predicts the ALT status best, and, second, which image-derived criteria describe the ALT status of cells. The first research question also involves estimates on how confident the models are when predicting the ALT status. Depending on the FGA and IBA model, we use the models' probabilities or second closest predictions to assess this confidence.

CHAPTER 7

Preliminary Experiments for the FGA and Segmentation Post-Processing Model

In this chapter, we address the technical research questions that corroborate how we set up the final FGA models based on the methodology introduced in Section 6.5 to discuss our answers to RQ1, RQ2 (see Section 6.8) presented in Chapter 9. On the one hand, these technical questions address criteria to include or exclude observations from the training, testing and validation samples $\mathcal{T}, \mathcal{T}_{IBA}, \mathcal{T}_{test}, \mathcal{V}$ as well as $\mathcal{D}$. On the other hand, we analyse how we shall optimally setup the FGA models with respect to selected features or chosen hyperparameters.

In Section 7.1, we start off by preliminarily defining the training, testing and validation samples for this chapter based on our setting of Section 6.5. Note that any insights from our posed technical research questions may immediately influence and change the setup of these samples. We will indicate these changes whenever necessary. In Section 7.2, we pose research questions to define FGA models. In Section 7.3, we define a post-processing model for the nucleus segmentation of Section 6.2 in line with our observations in Section 6.3.

7.1 Preliminary Sample Setup

To build our training, testing and validation samples $\mathcal{T}, \mathcal{T}_{IBA}, \mathcal{T}_{test}, \mathcal{V}$ based on our setting of Section 6.5, we use nuclei of the pure dilution series P3∼A, P6, P8, P9WH, P10, P11, P12, P13 series, for which we have nucleus labels $g_k \in \mathcal{G} = \{\text{ALT+}, \text{ALT−}\}$ available. We use the ALT− nuclei of the pure dilution series P4∼A and P7 in the

family of out-of-sample dilution series $\mathcal{D}$ to test the model performance on ALT− cells by determining false positive rates.

To setup $\mathcal{T}, \mathcal{T}_{IBA}, \mathcal{T}_{test}, \mathcal{V}$, we use the following rules:

1. *Balanced targets.* While the pure dilution series P3∼A, P6, P8, P9WH, P10, P11, P12, P13 include different amounts of ALT+ and ALT− cells originating from different cell lines SK-N-MM, CHLA90, SK-N-SH, CLB-MA, we ensure that $\mathcal{I} = \mathcal{T} \cup \mathcal{T}_{test}$ represents nuclei of all cell lines equally. In particular, $\mathcal{I}$ consists of 50% of ALT+ and ALT− cells.

2. *Minimum number of spots per nucleus.* Using the nucleus and spot segmentation masks of Section 6.2, we can count the number of telomere spots per nucleus. As discussed in Section 7.2.1, certain FGA features require at least two telomere spots per nucleus. For that reason, we include only nuclei that feature this minimum number of spots. Section 7.2.1 discusses in more detail, which minimum number of telomere spots is optimal.

3. *Reduced sample size.* The pure dilution series P3∼A, P6, P8, P9WH, P10, P11, P12, P13 contain more than 100,000 nuclei. To limit computational efforts, we reduce the training sample size to 20,000 nuclei. The following Sections 7.2.1 -7.2.6 indicate reduced training sample sizes if necessary.

4. *Split on field of view level.* We split $\mathcal{I}$ via an 80% and 20% ratio into $\mathcal{T}$ and $\mathcal{T}_{test}$. More specifically, we split the observations by ensuring that all nuclei of a field of view either pertain to $\mathcal{T}$ or $\mathcal{T}_{test}$, see Section 2.3.2 for details on fields of view. Again, we ensure that $\mathcal{T}, \mathcal{T}_{test}$ consist each of 50% ALT+ cells. For the IBA, we further split $\mathcal{T}$ into $\mathcal{T}_{IBA}$ and $\mathcal{V}$ according to an 80% and 20% ratio. Again, we split nuclei on field of view level.

5. *Scaling FGA features*: For FGA models of ALT classification on nucleus and series level, we scale the features on $\mathcal{T}$ by centering the mean and scaling the standard deviation to approximately 1. We use robust statistical estimates of the mean and standard deviation according to `sklearn`'s `RobustScaler` class [PVG+11]. We use the scaler that we estimated on $\mathcal{T}$ to also scale the features on $\mathcal{T}_{test}$.

To assess how well the already trained models predict the ALT+ ratio on actual dilution series, we also consider in $\mathcal{D} = \{\mathcal{D}_{P4\sim A}, \mathcal{D}_{P7}, \mathcal{D}_{PM1}, \mathcal{D}_{PM3}, \ldots, \mathcal{D}_{PM24}\}$ the family of actual dilution series as well as the pure dilution series P4∼A and P7, see Table 2.1. For each dilution series in $\mathcal{D}$ we know the ALT+ ratio $\rho \in \{0, 0.01, 0.05, \ldots, 0.75\}$. For the FGA models, we use the scaler that we estimated on $\mathcal{T}$ to also scale the features of all dilution series in $\mathcal{D}$. Furthermore, similarly to $\mathcal{T}$, we only include nuclei with a specific minimum number of telomere spots and reduce the number of nuclei of each dilution series in $\mathcal{D}$ to 4,000.

7.2 Feature Generation Approach

The methodology of Section 6.5 outlines our setup for FGA models and the relevant samples. In this section, we discuss all research questions that we pose to specify the setup the FGA models as well as the samples in detail.

7.2.1 Number of Telomere Spots and its Influence on the FGA Model Performance

Every human cell contains 46 chromosomes and therefore 92 telomeres. Due to mutations, tumor cells may contain less or more than 92 telomeres. In the telPNA channel, we should therefore identify up to around 92 telomere spots for both ALT+ and ALT− cells. In practice, we will usually see less spots due to occlusions or noise. One can argue that the number of identifiable spots per nucleus corresponds to the quality of the microscopy image. Lower number of spots might indicate more noisy images of poorer quality.

We want to analyse whether an imposed minimum number of telomere spots per nucleus affects the prediction quality of the nuclear FGA models. If so, we can impose this minimum number of telomere spots in the definition of the training, testing and validation samples $\mathcal{T}, \mathcal{T}_{IBA}, \mathcal{T}_{test}, \mathcal{V}$. Hence, we pose the following research question:

Does a minimum number of telomere spots per nucleus influence the performance of nuclear FGA models?

To answer the above research question, we fix a set of required minimum spot numbers $\underline{n}_{spots} \in \{2, \ldots, 15\}$ and check how the FGA model types of logistic regression, random forest, support vector machine, gradient boosting predict the ALT status for cells that have at least $\underline{n}_{spots}$ spots. We choose $\underline{n}_{spots} > 1$ as some FGA features require at least two spots per nucleus (e.g. distances between spots in a nucleus, see Section 6.6.1). More specifically, for each $\underline{n}_{spots}$, we will use the following experimental setup:

1. We generate $\mathcal{T}$ as described in Section 7.1 but we only include nuclei that feature at least $\underline{n}_{spots}$-many spots. For that purpose, we do not additionally reduce the sample size of $\mathcal{T}$. We then split $\mathcal{T}$ into four training folds and one validation fold according to 5-fold CV. There are five possibilities to combine the folds into four training folds and one validation fold.

2. We train the FGA models with `sklearn`'s standard parameters (without hyperparameter tuning) on the training folds of Step 1. For each FGA model type of Section 3.4, there are five differently trained models because of the five different training and validation fold splits.

3. We evaluate the trained FGA models of Step 2 on the corresponding validation folds to determine the out-of-sample accuracy, see Section 4.4.1. For each FGA

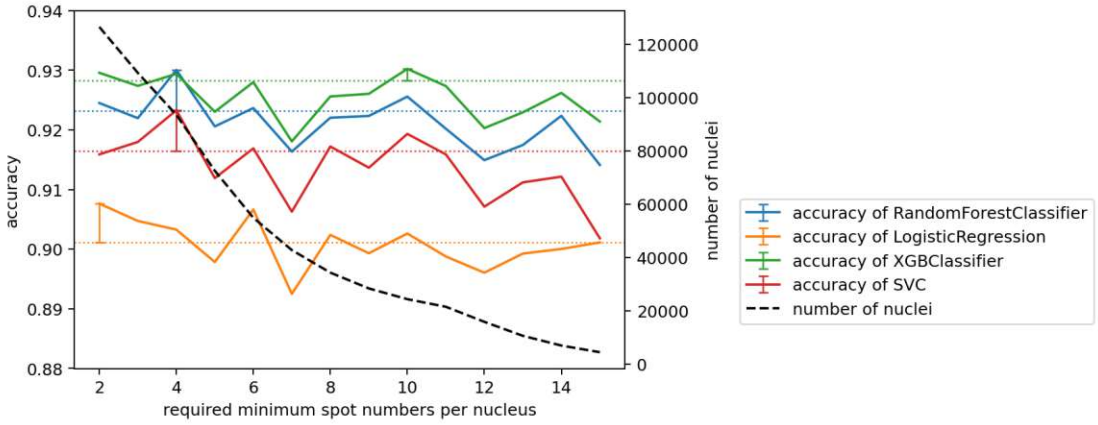


Figure 7.1: Plot of mean accuracy scores and number of observations in $\mathcal{T}$ with respect to the required minimum number of telomere spots $\underline{n}_{spots} \in \{2, \ldots, 15\}$ per nucleus. The left axis shows the mean accuracy scores for random forest (RandomForestClassifier, blue), logistic regression (LogisticRegression, yellow), gradient boosting (XGBClassifier, green) and support vector machine (SVC, SVM, red, which refers to the `sklearn.svm.SVC` classifier of [PVG+11]). The right axis depicts the number of nuclei in $\mathcal{T}$ that satisfy the required minimum number of telomere spots. For each model type, the plot also shows an error bar attached to the maximum mean accuracy score. The width of the error bar is given by the standard deviation of the corresponding mean accuracy score. The error bars set the score levels of one standard deviation *below* the maximum scores. Coloured horizontal dashed lines display this level for easier reference.

model type, we therefore get five accuracy scores. We determine the mean accuracy and its standard deviation for each FGA model type.

Following the above setup, we get mean accuracy values and corresponding standard deviations for each $\underline{n}_{spots}$ and each model type of Section 3.4. Figure 7.1 shows the results. We see that accuracy values do not differ much for all models and $\underline{n}_{spots} \in \{2, \ldots, 15\}$, while increasing $\underline{n}_{spots}$ of course considerably decreases the sample size. More specifically, the training sample size decreases from around 126.000 to 4.550 nuclei when increasing $\underline{n}_{spots}$ from 2 to 15. We also note that the accuracy values for $\underline{n}_{spots} = 2$ are always within one standard deviation of the highest accuracy values for each model setup.

Hence, we conclude that a minimum number of telomere spots per nucleus does not considerably influence the model performance of ALT classification models on nucleus level. In the remaining part of our analyses of this master's thesis, we will therefore use the smallest possible $\underline{n}_{spots} = 2$ as minimum number of spots that each nucleus has to feature to be included in $\mathcal{T}, \mathcal{T}_{test}$ as well as $\mathcal{T}_{IBA}, \mathcal{V}$ for consistency's sake.

7.2.2 Robust Feature Selection

If we use the classifiers from the previous Section 7.2.1, Figure 7.1 states that we can expect an out-of-sample accuracy score of 90%-93% on $\mathcal{T}_{test}$. It is important to note that this accuracy score is already quite high although we have not yet tuned hyperparameters. However, if we apply the same models for ALT classification on series level and aim at predicting the ALT+ ratio ρ in the dilution series of $\mathcal{D}$ as discussed at the beginning of Section 6.6.3, we notice a precipitous decline of the model performances. For all dilution series of $\mathcal{D}$, the models' predictions of ALT+ rates $\hat{\rho}$ are far off from the actual ratios ρ, see Figure 7.2.

We observe this behaviour for the classifiers of Section 7.2.1, as some features behave quite differently on the actual dilution series of $\mathcal{D}$. Given a certain feature function F_l, we would expect that an actual dilution series with very low ALT+ share (say 1% or 5%) should show a feature distribution that is quite similar to ALT− nuclei in $\mathcal{T}$. However, for some features, we observe quite the opposite: actual dilution series in $\mathcal{D}$ with very low ALT+ share have distributions that are completely off compared to ALT− nuclei in $\mathcal{T}$, see Figure 7.3. We call these features *unstable* and we are aiming at including only features in our models that are stable or robust in the sense that their distribution on actual dilution series with low ALT+ share is very similar to the distribution of ALT− nuclei. We may observe this unstable behaviour due to different image quality and fluorescence staining/ cytoplasmic background as discussed in Section 6.3. We therefore pose the following research question:

Which features are stable/robust enough to include them in the FGA model development?

To answer this question in accordance with Section 6.6.1, we first fix a sub-family $\mathcal{D}_{sub} = \{\mathcal{D}_{PM14}, \mathcal{D}_{PM15}, \mathcal{D}_{PM22}, \mathcal{D}_{PM23}\}$ of actual dilution series with very low ALT+ share of 1% or 5%, see also Table 2.1. Note that this sub-family includes actual dilution series that partly feature stronger cytoplasmic background that might lead to unstable features, see Section 6.3. Furthermore, we define by $\mathcal{T}_+ = \{\mathcal{D}_{P3\sim A}, \mathcal{D}_{P6}, \mathcal{D}_{P8}, \mathcal{D}_{P10}, \mathcal{D}_{P11}\}$ and by $\mathcal{T}_- = \{\mathcal{D}_{P9WH}, \mathcal{D}_{P12}, \mathcal{D}_{P13}\}$ the dilution series of $\mathcal{T}$ of ALT+ and ALT− nuclei, respectively. We setup the training sample $\mathcal{T}$ and the actual dilution series as discussed in Section 7.1 using a required minimum number of two spots per nucleus and 20,000 training observations. Given a feature function F_l, we can then use the following algorithm to find robust features

1. Determine the distributions $\mu^{F_l}_{P3\sim A}, \dots \mu^{F_l}_{P11}$ of F_l for all dilution series in $\mathcal{T}_+$ separately. Similarly, find the distributions $\mu^{F_l}_{P9WH}, \dots \mu^{F_l}_{P13}, \mu^{F_l}_{PM14}, \dots \mu^{F_l}_{PM23}$ of F_l for all dilution series in $\mathcal{T}_-$ and in $\mathcal{D}_{sub}$ separately.
2. Find the common mean m^+ and common standard deviation $\sigma^+$of $\mu^{F_l}_{P3\sim A}, \dots \mu^{F_l}_{P11}$. Center the distributions according to m^+ and scale them by $1/\sigma^+$ to have all

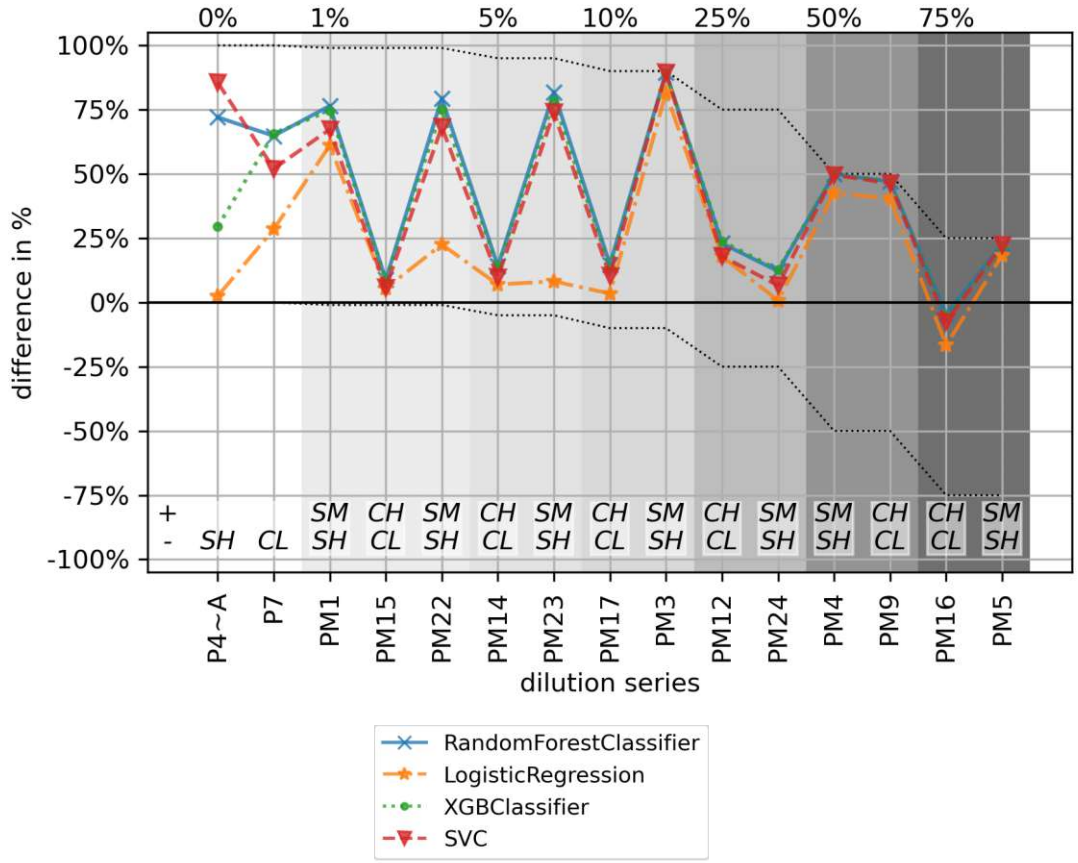
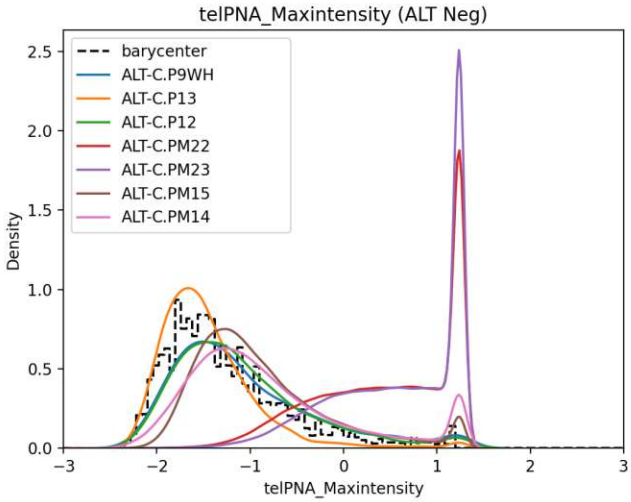
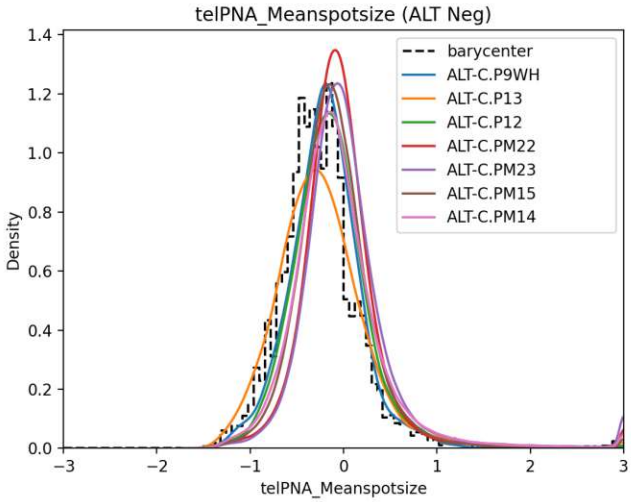


Figure 7.2: Differences of predicted ALT+ rate and true ALT+ ratio of dilution series in $\mathcal{D}$. The figure shows the differences for all classifiers from Section 7.2.1 with different colours, line styles and markers. The dotted lines on top and at the bottom show the worst possible differences for each dilution series. The shaded background corresponds to the true ALT+ ratio, which is also indicated on top of the figure. At the bottom of the figure, we find the cell lines of each dilution series in two separate rows with the following encoding: the top row (+) records the ALT+ cell lines SK-N-MM (SM) or CHLA90 (CH). In the bottom row (-), we find the ALT− cell lines SK-N-SH (SH) or CLB-MA (CL).



(a) unstable feature



(b) stable feature

Figure 7.3: The above figures show the (scaled and centered) distributions of two features that measure the maximum intensity of a nucleus in the telPNA channel (above) and the average spot size of a nucleus in the telPNA channel (below), respectively. The figures illustrate the distributions for all ALT− dilution series in $\mathcal{T}_-$ as well as in dilution series of very low ALT+ share in $\mathcal{D}_{sub}$. The dashed black line shows the Wasserstein barycenter μ^-_{bary} of the dilution series in $\mathcal{T}_-$. In the above figure, we see that the distributions on the actual dilution series PM22 and PM23 are far from the barycenter and therefore lead to greater Wasserstein distances d_{PM22}, d_{PM23}. In the figure below, we note only very small differences among all distributions.

distributions on the same scale. Determine the Wasserstein barycenter μ^+_{bary} of the centered and scaled distributions, see Section 6.6.3. Find the Wasserstein distances W_2 between μ^+_{bary} and each centered and scaled distribution. Denote these distances by $d^{F_l}_{P3\sim A}, \dots, d^{F_l}_{P11}$. To calculate Wasserstein distances and barycenters, we use the python optimal transport (POT) package [FCG+21].

3. Find the common mean m^- and common standard deviation σ^- of $\mu^{F_l}_{P9WH}, \dots \mu^{F_l}_{P13}$, $\mu^{F_l}_{PM14}, \dots \mu^{F_l}_{PM23}$. Center the distributions according to m^- and scale them by $1/\sigma^-$ to have all distributions on the same scale. Determine the Wasserstein barycenter μ^-_{bary} of the centered and scaled distributions of the *pure* dilution series $\mu^{F_l}_{P9WH}, \mu^{F_l}_{P12}, \mu^{F_l}_{P13}$. Find the Wasserstein distances W_2 between μ^-_{bary} and each centered and scaled distribution of $\mu^{F_l}_{P9WH}, \dots \mu^{F_l}_{P13}$, $\mu^{F_l}_{PM14}, \dots \mu^{F_l}_{PM23}$. Denote these distances by $d^{F_l}_{P9WH}, \dots, d^{F_l}_{PM23}$.

For stable features of corresponding feature functions F_l, we would expect small distances $d^{F_l}_{P9WH}, \dots, d^{F_l}_{PM23}$ for dilution series in $\mathcal{T}_-, \mathcal{D}_{sub}$, as well as small distances $d^{F_l}_{P3\sim A}, \dots, d^{F_l}_{P11}$ for dilution series in $\mathcal{T}_+$. To the best of our knowledge, there is no statistical reasoning for determining a threshold below which we consider Wasserstein distances small enough. Instead, by visually inspecting the centered and scaled distributions of above's algorithm, we have decided to use 0.0185 as threshold and consider all Wasserstein distances $d \leq 0.0185$ as small, and all other Wasserstein distances $d > 0.018$ as too big. We therefore consider a feature robust, if both of the following two conditions hold:

1. The feature exhibits Wasserstein distances $d^{F_l} \leq 0.0185$ for all dilution series in $\mathcal{T}_-, \mathcal{T}_+$ and $\mathcal{D}_{sub}$ and
2. $M^- + M^+ \leq 0.03$, where $M^- = \max(d^{F_l}_{P9WH}, \dots, d^{F_l}_{PM23})$ and $M^+ = \max(d^{F_l}_{P3\sim A}, \dots, d^{F_l}_{P11})$. This condition ensures that a feature does not exhibit Wasserstein distances that are below but close to the threshold 0.0185 both on $\mathcal{T}_-, \mathcal{D}$ as well as $\mathcal{T}_+$.

Based on these two conditions, we find the following 14 stable features of the telPNA channel, see also Section 6.6.1 for details on the definition:

- *mean spot size.*
- *cluster prominence.*
- *cluster shade*: measures skewness and uniformity of the GLCM. Higher cluster shade values imply greater asymmetry about the mean [vGFP+17].
- *gray level non-uniformity.*

- *run length non-uniformity.*
- *large area high gray level emphasis*: based on the GLSZM, this feature measures the proportion in the image of the joint distribution of larger size zones with higher gray-level values [vGFP+17].
- *size zone non uniformity*: based on the GLSZM, this features measures the variability of size zones in the image, with a lower value indicating more homogeneity in size zones [vGFP+17].
- *kurtosis.*
- *skewness.*
- *standard deviation of spot size.*
- *size brightest spot.*
- *size largest spot.*
- *number of spots.*
- *standard deviation of spot distances*: for each nucleus, we can determine the telomere spots and find the distances between all spots. We then use the standard deviation of these differences.

Note that these 14 features are stable but are not necessarily important for determining the ALT status. In Section 9.3, we will discuss the corresponding feature importances and their biological interpretation for each FGA model.

7.2.3 Image Normalisation for Feature Selection

As discussed in Section 6.6.1, image preprocessing is a common option before extracting features of a microscopy image [RNM17] . For example, one often normalises the image intensity levels to ensure that all images are on the same common scale. To that end, one can min-max normalise a field of view F by linearly scaling the pixel intensities to a common minimum, say 0, and a common maximum, say 255 via

$$p_{norm} = \frac{p - \min(F)}{\max(F) - \min(F)} \cdot 255,$$

where p denotes the unnormalised intensity of a given pixel and $\min(F), \max(F)$ refer to the minimum and maximum intensity levels of nuclei in F before normalisation. One can argue that properly preprocessed images are better suited for feature extraction and therefore provide more accurate classification models. Based on the 14 robust features selected in Section 7.2.2, we therefore pose the following research question:

I_k normalised	Rand. Forest	Log. Regression	XGB	SVC
no	0.880 ± 0.009	0.856 ± 0.007	0.877 ± 0.006	0.864 ± 0.008
yes	0.880 ± 0.008	0.856 ± 0.006	0.876 ± 0.009	0.865 ± 0.007

Table 7.1: Mean accuracy scores and corresponding standard deviations for nuclear FGA models using normalised or unnormalised robust features.

Does image normalisation improve the performance of nuclear FGA models?

To answer this question, we setup the training and testing samples $\mathcal{T}, \mathcal{T}_{test}$ as discussed in Section 7.1 using a required minimum number of two spots per nucleus and 20,000 training observations. We then extract the 14 features of Section 7.2.2 once based on normalised images and once using unnormalised (original) images. Separately for features based on normalised and unnormalised images, we use 5-fold cross validation on $\mathcal{T}$ to train 5 different models for each model type of random forest, logistic regression, support vector machine and gradient boosting, similarly to the setup of Section 7.2.1. We then evaluate the models on the corresponding validation fold and determine mean accuracy scores and corresponding standard deviations as discussed in Section 7.2.1. Table 7.1 summarises the results for each model type using features of normalised and unnormalised images. For each model type, we find that accuracy scores are almost identical for features based on normalised or unnormalised images. Hence, we conclude that image normalisation does not improve the performance of nuclear FGA models. We will therefore not normalise images when extracting robust features for FGA models.

7.2.4 Hyperparameter Tuning

As discussed in Sections 3.4 and 6.6, nuclear FGA models depend on several hyperparameters. Choosing hyperparameters is crucial for optimal model performance on $\mathcal{T}_{test}$ as well as on $\mathcal{D}$ and for finding models that generalise well on previously unseen data. We therefore pose the following research question:

Which hyperparameters are necessary for nuclear FGA models to perform well and to be capable of generalising on previously unseen data?

To answer above's question, we again use 5-fold CV on $\mathcal{T}$ as defined in Section 7.1 using a required minimum number of two spots per nucleus, 20,000 training observations and robust features based on unnormalised images. To illustrate our approach, we fix a certain FGA model with tunable hyperparmeters $(h_1, \ldots, h_l)$, e.g. $h_1 = \lambda, h_2 =$ "l1","l2" for penalised logistic regression indicating whether we use an L^1 or L^2 penalty, see Section 3.4. We then define a finite grid $\mathcal{H}$ of possible values for $(h_1, \ldots, h_l)$ (e.g. $\mathcal{H} = \{(h_1, h_2)\} = \{0.01, 0.1, 1\} \times \{$"l1","l2"$\}$). For a given hyperparameter setup $H = (h_1, \ldots, h_l) \in \mathcal{H}$, we then conduct the following steps[1]:

[1]Note that we use the same 5-fold CV splits for each hyperparmeter setup $H \in \mathcal{H}$ and FGA model.

1. We split the training sample $\mathcal{T}$ into training and validation folds as described in Section 7.2.1 and train the FGA model on the training folds using hyperparmeters H.

2. We then determine the accuracy of the FGA model on the validation fold.

3. We repeat the two steps above for each possible permutation of training and validation folds. We therefore find 5 accuracy scores for each setup H of hyperparameters of $\mathcal{H}$ for which we can determine the mean accuracy a_H and the corresponding standard deviation s_H.

4. For each FGA model, we can therefore determine the hyperparameter setup H^{max} with maximum average accuracy $\bar{a}_{max} = \bar{a}_{H^{max}}$ and corresponding standard deviation $s_{max} = s_{H^{max}}$. Similarly to Section 6.6.3, we note that all other hyperparameter setups H with
$$\bar{a}_{max} - s_{max} \leq \bar{a}_H \leq \bar{a}_{max} \tag{7.1}$$
show very similar accuracy values, as their mean accuracy values lie within one standard deviation of the maximum average accuracy. To find hyperparameters that give models that generalise well on previously unseen data, we choose a hyperparameter setup H which satisfies (7.1) and gives the stiffest model setup among all H for which (7.1) holds. Section 3.4 discusses for each model type separately how hyperparameters may provide stiff models. If there are multiple hyperparameters that provide stiff models, we optimise them consecutively in the order of the description of Section 3.4. If there are still multiple feasible stiff model setups, we choose the one with the highest mean accuracy.

The last step of above's algorithm is important to ensure that the models generalise well on previously unseen data. Stiff models usually provide smoother decision boundaries with lower variance at the expense of higher biases. Sections 3.2 and 3.4 provide further details. Using the above algorithm, Table 7.2 denotes the chosen grid values of $\mathcal{H}$ and the optimal hyperparameters for all nuclear FGA models. In the following analyses of this master's thesis, we will use these optimal hyperparameters.

7.2.5 Sound Coefficients of Logistic Regression Model

The penalised logistic regression model as introduced in Section 3.4 is a comparably simple and yet easy to understand model. As we standardised the input features, the coefficients of the logistic regression model directly indicate how important the corresponding features are for the model. Contrary to support vector machines, gradient boosting and random forests, correlated features may unduly influence the coefficients of the logistic regression [HTF09]. For that reason, we are using penalised logistic regression with L^1 penalty to reduce the influence of highly correlated features, see Section 7.2.4. Lasso logistic regression usually dampens the impact of highly correlated inputs to avoid exploding coefficients, which are a common symptom of high correlations among the covariates. In

FGA Model Type	Parameter	Grid Values	Optimal Values
Logistic Regression	C	$2^{[-3,-2,\ldots,3]}$	0.125
	penalty	l1, l2	l1
Random Forest	max_depth	2, 5, 8	8
	max_features	0.5, sqrt	sqrt
	max_samples	0.1, 0.5	0.1
	min_samples_leaf	2, 5, 10	10
	min_samples_split	5, 10	5
	n_estimators	100, 500, 1,000	1,000
SVC	C	$2^{[-4,-3.5,\ldots,0]}$	0.0625
	kernel	rbf, poly	rbf
	gamma	scale, auto	auto
XGB	colsample_bytree	0.5, 0.8	0.8
	eta	0.05, 0.2, 0.3	0.05
	gamma	0, 1	1
	max_depth	1, 3, 5	1
	min_child_weight	1, 3, 5	3
	n_estimators	100, 500, 2,000	2,000
	subsample	0.5, 0.8	0.5
	reg_lambda	1, 1.5, 2	1

Table 7.2: Optimal hyperparameters and chosen grid values of $\mathcal{H}$ of nuclear FGA models. See Section 3.4 and [PVG+11] for details on the grid values. Note that the hyperparameter $C > 0$ of the logistic regression corresponds to the inverse weight $1/\lambda > 0$ of the penalty term λ in Section 3.4. Furthermore, note that the hyperparameter gamma of SVC corresponds to a scaling parameter of radial basis functions.

this section, we discuss whether the regression coefficients of lasso logistic regression are indeed reasonable by answering the following question:

Are the signs of the coefficients of the logistic regression model in line with biological reasoning?

We consider the logistic regression model based on the hyperparameters and training/testing samples $\mathcal{T}, \mathcal{T}_{test}$ as described in Section 7.2.4 and the robust features of Section 7.2.2. Table 7.3 gives the coefficients of the resulting lasso logistic regression model and also provides the sign that we expect based on biological reasoning.

While we see that the L^1 successfully prevents coefficients from exploding, we find that the coefficients of the six features *gray level non-uniformity*, *standard deviation of spot size*, *cluster prominence*, *average spot size*, *size zone non-uniformity* and *kurtosis* show the wrong sign. This can be due to correlated features in the same group (e.g. cluster prominence and cluster shade of the feature group GLCM), where one feature (e.g. cluster

Feature Name	**Feature Group**	**Exp. Sign**	β	$\beta_{\textbf{re-est}}$
skewness	pyrad. 1st order	+	2.40	2.40
run length non-uniformity	pyrad. GLRLM	+	1.51	−0.01
gray level non-uniformity	pyrad. GLRLM	+	−1.32	excluded
size largest spot	man. defined	+	0.74	0.21
spotcount	man. defined		0.51	0.55
cluster shade	pyrad. GLCM	+	0.35	0.33
σ of spot size	own definition	+	−0.34	excluded
cluster prominence	pyrad. GLCM	+	−0.16	excluded
average spotsize	man. defined	+	−0.12	excluded
large area high gray lev. emph.	pyrad. GLSZM	+	0.12	0.10
σ of spot distances	man. defined		0.07	excluded
size zone non-uniformity	pyrad. GLSZM	+	−0.04	excluded
kurtosis	pyrad. 1st order	+	−0.03	excluded
size brightest spot	man. defined	+	0.02	excluded

Table 7.3: β coefficients of the lasso logistic regression model according to the hyperparameters of Section 7.2.4. The table shows the feature group according to Section 6.6.1, the expected sign according to biological reasoning, the β coefficients of the lasso logistic regression model using all 14 features and the $\beta_{\text{re-est}}$ coefficients of the re-estimated lasso logistic regression model using the six features with correct sign and sufficiently large coefficient. The following arguments underpin the expected coefficient signs: for ALT+ nuclei, we expect greater 1st order features due to heavy tailed brightness curves, bigger cluster and non-uniformity features (GLCM, GLRLM, GLSZM), and greater values for spot size and spot brightness, since ALT+ cells usually exhibit bigger and brighter spots as well as more heterogeneous telPNA images, see Section 2.2. For *spotcount* and *standard deviation of spot distances*, we cannot infer a biologically sound coefficient sign.

shade) shows the correct sign but the other feature (e.g. cluster prominence) not. Figure 7.4 shows the pairwise Pearson correlations among all 14 features.

Moreover, we find that the coefficients of the two features *standard deviation of spot distances*, *size brightest spot* are already negligibly small. As a result, we note that the coefficients of the afore-mentioned variables are not reasonable or indicate that the features are irrelevant. To find a logistic regression model with biologically plausible coefficient signs, we re-estimate the model with the same hyperparameters and excluding all afore-mentioned eight variables. By excluding these variables, we still keep at least one feature per feature group according to Table 7.3 in the model. Table 7.3 shows the new $\beta_{\text{re-est}}$ coefficients in the outmost right column. We note that most of the remaining six features have kept a similar coefficient estimate β. Only the coefficients of *run length non-uniformity* and *size largest spot* have changed considerably. *run length non-uniformity* has become unimportant in the re-estimated model by showing a very small coefficient (of wrong sign), while the weight of *size largest spot* is much lower than before. We can attribute these changes again to correlations among the 14 features, see

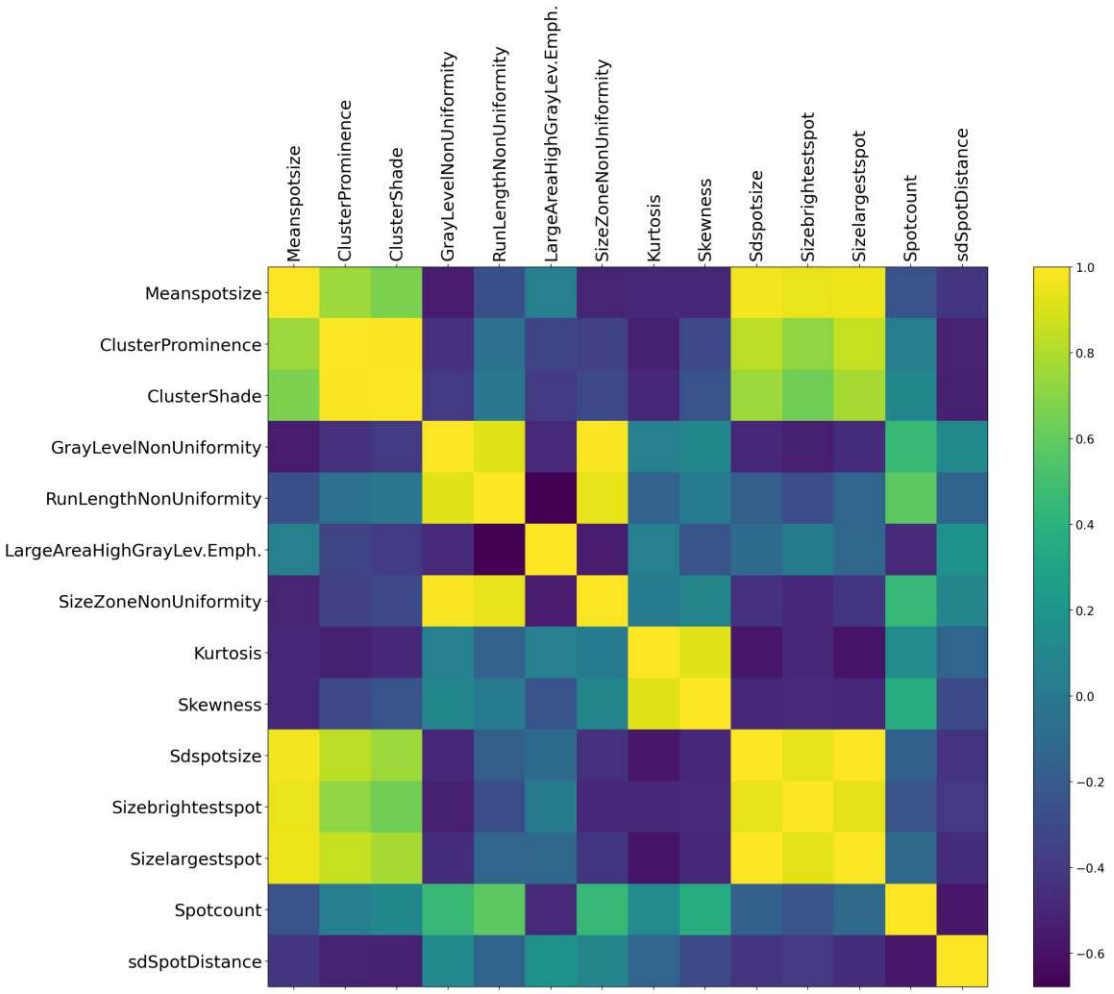


Figure 7.4: Pairwise Pearson correlation coefficients of the 14 robust features on $\mathcal{T}$. We note high correlations among features that are related to size (*mean spot size*, *standard deviation of spot size*, *size brightest spot*, *size largest spot*), clusters (*cluster prominence*, *cluster shade*), uniformity (*gray level non-uniformity*, *run length non-uniformity*, *size zone non-uniformity*) and distribution shape (*kurtosis*, *skewness*).

Figure 7.4. For example, β of *size largest spot* is offset by the coefficients of the highly correlated features *size brightest spot*, *average spot size*, *standard deviation spot size*. $\beta_{\text{re-est}}$ of *size largest spot* and the re-estimated model reverts this offset. In summary, we conclude that the reduced set of six features provides sound coefficients in the logistic regression model.

7.2.6 Variable Selection for Wasserstein Distance Models

As discussed in Section 6.6.3, we have to select a pair of features or corresponding feature functions F_1, F_2 to define a Wasserstein distance model of ALT classification on series level. We want to find a pair of feature functions that gives the best results by providing both high and reliable accuracy scores on $\mathcal{T}_{test}$ and previously unseen data. The variable selection algorithm proposed in Section 6.6.3 aims at finding such a pair of variables. In this section, we want to apply the algorithm and thereby answer the following research question:

Which pair of features provides the best results of the Wasserstein distance models?

To answer above's question, we again use $\mathcal{T}, \mathcal{T}_{test}$ as defined in Section 7.1 using a required minimum number of two spots per nucleus, 20,000 training observations and robust features based on unnormalised images. To calculate Wasserstein distances, we use the python optimal transport `POT` package [FCG+21]. According to the algorithm of Section 6.6.3, the pair of the features *cluster prominence* and *standard deviation of spot size* gives the best results by providing both high and reliable accuracy scores on the validation folds of $\mathcal{T}$, see Section 6.6.1 for a definition of the features. We will therefore use this pair of features for the Wasserstein distance model of ALT classification on series level.

7.3 Segmentation Post-Processing Model

As discussed in Section 6.3, occluded or cramped nuclei in the microscopy images may lead to inaccurate nucleus segmentation. Furthermore, generating fields of view of the telPNA channel images leads to cropped nuclei at the border. Inevitably, the segmentation masks of these nuclei will also be cropped and therefore be inaccurate. Inaccurate segmentation masks lead to wrongly extracted nuclei and therefore adversely affect classification of FGA and IBA models.

We want to address inaccurate (i.e. too big or too small) nucleus segmentation via a post-processing model of the segmentation. As mentioned in Section 6.3, we aim at applying this model to all images of dilution series in $\mathcal{D}$. The model shall incorporate geometric properties of nuclei (such as elongation, convexity, roundness) and information on bordering nuclei to predict whether the segmentation mask of a nucleus is wrong. If so, we will exclude the nucleus from images of $\mathcal{D}$. Finding such a model is tantamount to answering the following research question:

Dilution Series	Field of View ID	Number of Nuclei
ALT-C.PM3	Img-000126	250
ALT-C.PM4	Img-000353	265
ALT-C.PM22	Img-000849	216
ALT-C.PM23	Img-001155	346
ALT-C.PM12	Img-001037	168
ALT-C.PM14	Img-000969	101
ALT-C.P10	Img-000985	146
ALT-C.P11	Img-000875	87
ALT-C.P9WH	Img-000216	113
ALT-C.P12	Img-001116	129

Table 7.4: Data used to train and test the post-segmentation model. The data consists of nuclei in the quoted fields of view.

How can we exclude wrongly segmented nuclei from $\mathcal{D}$ with high precision?

As mentioned in Section 4.4.1, we aim at high precision instead of high accuracy because we do not want to unduly influence the ALT+ ratio by excluding nuclei that are actually correctly segmented. To simplify matters, we will build a logistic regression model to answer above's research question. To that end, we need geometric features of nuclei that we have introduced in Section 6.6.1. Furthermore, we need labelled data of nuclei that are correctly or wrongly segmented. We obtain this data from experts of the CCRI that manually labelled fields of view of pure and actual dilution series. We have to include actual dilution series in the data, as they are specifically affected by occluded and cramped nuclei according to Section 6.3. Table 7.4 gives an overview of the considered data.

We split the data into a trainings sample $\mathcal{T}^{post}$ and a testing sample $\mathcal{T}^{post}_{test}$ according to an 80%-20% split. We train the logistic regression model by first tuning its hyperparameters $C > 0$ and the penalty function via cross validation on $\mathcal{T}^{post}$ similarly to Section 7.2.4. However, contrary to Section 7.2.4, we do not aim at maximising the mean accuracy score across the validation folds but we use the precision score as our target function to optimise the hyperparameters.

As a result, we optimally build the logistic regression model with an L^2 penalty function and $\lambda = 8$ (i.e. $C = 0.125$). After training on $\mathcal{T}^{post}$ with these hyperparameters, the post-processing model provides out-of-sample accuracy and precision scores on $\mathcal{T}^{post}_{test}$ of around 90% and 71%, respectively. From now onwards, we will apply it to every dilution series of $\mathcal{D}$ and only include nuclei if the post-processing model predicts that they are correctly segmented. Depending on the dilution series of $\mathcal{D}$, this will exclude 2-30% of all nuclei.

7.4 Summary

In this chapter, we discussed six research questions to setup further methodological details of the FGA models as well as $\mathcal{T}, \mathcal{T}_{test}, \mathcal{T}_{IBA}, \mathcal{V}$: we have learned that a minimum number of two spots for each nucleus is best to setup $\mathcal{T}, \mathcal{T}_{test}, \mathcal{T}_{IBA}, \mathcal{V}$. Moreover, we identified 14 stable features for the FGA models using Wasserstein distances. Furthermore, we established that normalising features does not improve the performance of nuclear FGA models and found optimal hyperparameters for the nuclear FGA models. For the logistic regression model, we further reduced the number of predictors to ensure sound coefficients. Last, we selected an optimal pair of variables for the Wasserstein Distance Model.

We have also set up a segmentation post-processing model to exclude wrongly segmented nuclei with high precision. This model is necessary to address the technical challenges that we discussed in Section 6.3.

CHAPTER 8

Preliminary Experiments for the IBA

By analogy with the previous chapter, this chapter discusses specific, more technical research questions that substantiate how we set up the IBA models based on the methodology of Section 6.5. These preliminary experiments allow us to specify how we train the IBA models with respect to image normalisation, hyperparameter tuning and robustification techniques to answer our main research questions RQ1 and RQ2 in Chapter 9. Using `pytorch` [PGM+19], we implement the MyNet and ResNet models of Section 6.7 and start by summarising the sample setup in Section 8.1. Afterwards, we discuss in Section 8.2 how we find optimal hyperparameters for the IBA models. Finally, in Section 8.3, we outline how we can robustify the IBA models to gain better out-of-sample performance on $\mathcal{D}$.

8.1 Sample Setup

For all of our experiments of this chapter, we will use $\mathcal{T}_{IBA}, \mathcal{V}, \mathcal{T}_{test}$ based on our methodology of Section 6.5 and insights from the experiments of Chapter 7. More specifically, we take $\mathcal{T}, \mathcal{T}_{test}$ of the experiment in Section 7.2.4 as a basis. In particular, $\mathcal{T}, \mathcal{T}_{test}$ only include nuclei of pure dilution series given in Section 7.1 if they feature at least two telomere spots. Furthermore, $\mathcal{T}$ includes 20,000 observations. As stated in Sections 6.5 and 7.1, we partition $\mathcal{T}$ into $\mathcal{T} = \mathcal{T}_{IBA} \cup \mathcal{V}$ according to an 80%-20% split on field-of-view level. We ensure that $\mathcal{T}_{IBA}$ and $\mathcal{V}$ feature the same ALT+ ratio (i.e. 50%). Whenever we use dilution series of $\mathcal{D}$, we include nuclei based on the criteria defined above and, in addition, whether the corresponding nucleus segmentation was correct according to the the post-segmentation processing model of Section 7.3. Finally, as discussed in Sections 6.5 and 6.7, note that we will use only the telPNA channel to train and apply MyNet,

while we use all three channels (telPNA, nuclear segmentation mask, DAPI channel) for ResNet.

8.2 Image Normalisation, Optimal Parameters of regularised Adam

As discussed in Section 4.4, training CNNs and ResNets depends on multiple parameters, such as learning rate α, regularisation parameter λ of the regularised version `torch.optim.AdamW` of Adam, and batch size S. Furthermore, we have the possibility to min-max normalised images before inputting them into the networks. Finding optimal hyperparameters is crucial for ensuring good model performance. Therefore, we want to answer the following two research questions in this section:

Does image normalisation improve the performance of IBA models?
What are optimal hyperparameters α, λ, S of IBA models?

To answer above's questions, we follow a similar approach as discussed in Section 7.2.4. We fix a grid of hyperparameters, train the IBA models on $\mathcal{T}_{IBA}$ for all hyperparameter combinations of the grid and evaluate the models on $\mathcal{T}_{test}$. We do not use cross validation due to the computational complexity of training deep learning networks. For the same reason, the grid is much smaller than in Section 7.2.4. More specifically, we choose α either 0.001 or 0.0001, λ either 0.01 or 0.005, S either 32 or 64 and use either normalised or not normalised nuclear images as inputs. This grid varies among the default parameters of `torch.optim.AdamW` and we use default values for all other parameter settings of `torch.optim.AdamW` (such as momentum parameters). The following two paragraphs discuss the results for both models MyNet and ResNet separately.

Table 8.1 shows the accuracy scores on $\mathcal{T}_{test}$ of the corresponding MyNet models. We see that the best result for normalised images is given by $(\alpha, \lambda, S) = (0.0001, 0.005, 64)$, while for not normalised images $(\alpha, \lambda, S) = (0.0001, 0.01, 64)$ gives a slightly greater test accuracy score. As discussed in Section 7.2.2, we note that high accuracy scores on $\mathcal{T}_{test}$ do not necessarily imply good results on $\mathcal{D}$. For that reason, we also apply the afore-mentioned two models on[1] $\mathcal{D}_{sub}$ to see which of them gives more stable predictions, similarly to what we have done in Section 7.2.2 to select robust features. Based on the results depicted in Figure 8.1, we find that the unnormalised model gives better predictions on PM14 and PM15, but shows much worse results on PM22 and PM23 than the normalised model. Overall, the results of the normalised model are more stable. For that reason, we will use normalisation and the hyperparameters $(\alpha, \lambda, S) = (0.0001, 0.005, 64)$ for MyNet.

Table 8.2 shows the accuracy scores on $\mathcal{T}_{test}$ of the ResNet models based on varying hyperparameters. We see that the best result for normalised images is given by $(\alpha, \lambda, S) = (0.001, 0.005, 32)$, while for not normalised images $(\alpha, \lambda, S) = (0.001, 0.01, 32)$ gives an

[1] see Section 7.2.2 for a definition of $\mathcal{D}_{sub}$.

Normalisation	α	λ	S	**Epochs Until Early Stopping**	**Accuracy on $\mathcal{T}_{test}$**
yes	0.0001	0.01	32	42	0.923
yes	0.0001	0.01	64	66	0.930
yes	0.0001	0.005	32	62	0.929
yes	0.0001	0.005	64	72	**0.932**
yes	0.001	0.01	32	34	0.882
yes	0.001	0.01	64	37	0.899
yes	0.001	0.005	32	39	0.908
yes	0.001	0.005	64	34	0.884
no	0.0001	0.01	32	46	0.921
no	0.0001	0.01	64	72	**0.935**
no	0.0001	0.005	32	45	0.931
no	0.0001	0.005	64	66	0.927
no	0.001	0.01	32	34	0.912
no	0.001	0.01	64	42	0.906
no	0.001	0.005	32	34	0.873
no	0.001	0.005	64	41	0.904

Table 8.1: Accuracy on $\mathcal{T}_{test}$ for MyNet with different learning rate α, weight decay λ, and batch size S, as well as based on normalised and not normalised nuclear images. For better reference, the highest accuracy score for normalised as well as not-normalised nuclear images is indicated in bold font. Accuracy scores are given in decimal numbers between 0 (0%) and 1 (100%).

even greater test accuracy score. Overall, the test accuracy scores do not vary much for the different hyperparmeter setups. We again also apply these two models on $\mathcal{D}_{sub}$ to see which of them gives more stable predictions. Based on the results depicted in Figure 8.1, we find that normalisation considerably improves the prediction results on $\mathcal{D}_{sub}$. For that reason, we will use normalisation and the hyperparameters $(\alpha, \lambda, S) = (0.001, 0.005, 32)$ for our ResNet model.

In summary, we note that normalising nuclear images considerably improves the model predictions of MyNet and ResNet on $\mathcal{D}_{sub}$. As discussed in Section 7.2.3, this result is quite contrary to our findings for FGA models, for which normalisation did not change the prediction results based on robust features of Section 7.2.2.

8.3 Data Augmentation Techniques, Dropouts, Batch Normalisation

In Section 4.4.3, we discuss various methods to combat overfitting of deep learning networks. Most notably, we can apply various data augmentation techniques, apply batch normalisation or dropouts. In this section, we assess whether and how these techniques

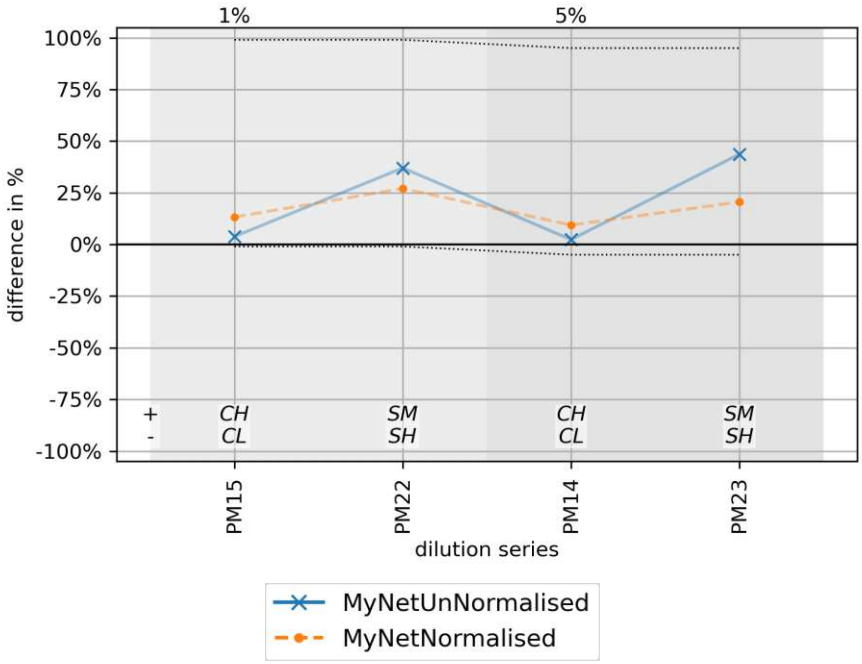


Figure 8.1: Differences of predicted ALT+ rate and true ALT+ ratio of dilution series in $\mathcal{D}_{sub}$. The Figure shows the differences for MyNet using image normalisation and the hyperparameters $(\alpha, \lambda, S) = (0.0001, 0.005, 64)$, as well as MyNet without normalisation and $(\alpha, \lambda, S) = (0.0001, 0.01, 64)$. The shaded background corresponds to the true ALT+ ratio, which is also indicated on top of the figure. At the bottom of the figure, we find the cell lines of each dilution series in two separate rows with the following encoding: the top row (+) records the ALT+ cell lines SK-N-MM (SM) or CHLA90 (CH). In the bottom row (-), we find the ALT− cell lines SK-N-SH (SH) or CLB-MA (CL).

improve the predictions of MyNet and ResNet based on the optimal hyperparameters of Section 8.2. More specifically, we want to answer the following research question:

Do data augmentation techniques, batch normalisation and dropouts improve the performance of the IBA models?

To answer this question, we again train the IBA models using the optimal hyperparameters of Section 8.2 and different robustification setups to assess their performance on $\mathcal{T}_{test}$ and $\mathcal{D}_{sub}$, similarly to our approach of Section 8.2. We will use the data augmentation techniques *random blurring*, *random sharpness* and *random halo effect* as discussed in Section 6.7. To further reduce high computational efforts, we will assess the effect of batch normalisation, dropouts and data augmentation via the following five setups to combat overfitting:

1. use only batch normalisation and early stopping.

Normalisation	α	λ	S	**Epochs Until Early Stopping**	**Accuracy on $\mathcal{T}_{test}$**
yes	0.0001	0.01	32	139	0.939
yes	0.0001	0.01	64	111	0.939
yes	0.0001	0.005	32	69	0.935
yes	0.0001	0.005	64	109	0.937
yes	0.001	0.01	32	63	0.934
yes	0.001	0.01	64	61	0.939
yes	0.001	0.005	32	51	**0.940**
yes	0.001	0.005	64	51	0.935
no	0.0001	0.01	32	89	0.937
no	0.0001	0.01	64	70	0.934
no	0.0001	0.005	32	87	0.937
no	0.0001	0.005	64	113	0.939
no	0.001	0.01	32	92	**0.944**
no	0.001	0.01	64	101	0.931
no	0.001	0.005	32	79	0.941
no	0.001	0.005	64	118	0.943

Table 8.2: Accuracy on $\mathcal{T}_{test}$ for ResNet with different learning rate α, weight decay λ, and batch size S, as well as based on normalised and not normalised nuclear images. For better reference, the highest accuracy score for normalised as well as not-normalised nuclear images is indicated in bold font. Accuracy scores are given in decimal numbers between 0 (0%) and 1 (100%).

2. like 1., but also using random halo effect.
3. like 2., but also using random blurring and random sharpness.
4. like 1, but also using dropouts.
5. use all techniques combined, i.e. 2., 3. and 4. In this case, we augment data on average for 51.2% of all nuclear images.

Note that we apply the five setups above for MyNet, but we can only apply setups 2 and 3 for ResNet. This is because we only fine-tune the last layer of ResNet and therefore we cannot use batch normalisation and additional dropout layers. The following two paragraphs discuss the results for MyNet and ResNet separately.

Table 8.3a shows the accuracy scores on $\mathcal{T}_{test}$ as well as the absolute difference between predicted ALT+ rate and actual ALT+ rate on $\mathcal{D}_{sub}$ for MyNet. Interestingly, we again note that highest accuracy scores on $\mathcal{T}_{test}$ do not necessarily come together with good predictions on $\mathcal{D}_{sub}$. Setups 3., 4. and 5. give good results on $\mathcal{T}_{test}$ as well as for dilution series PM14 and PM15. However, these setups fail at predicting the ALT+ rate for dilution series PM22 and PM23. Using only the implemented halo effect as data

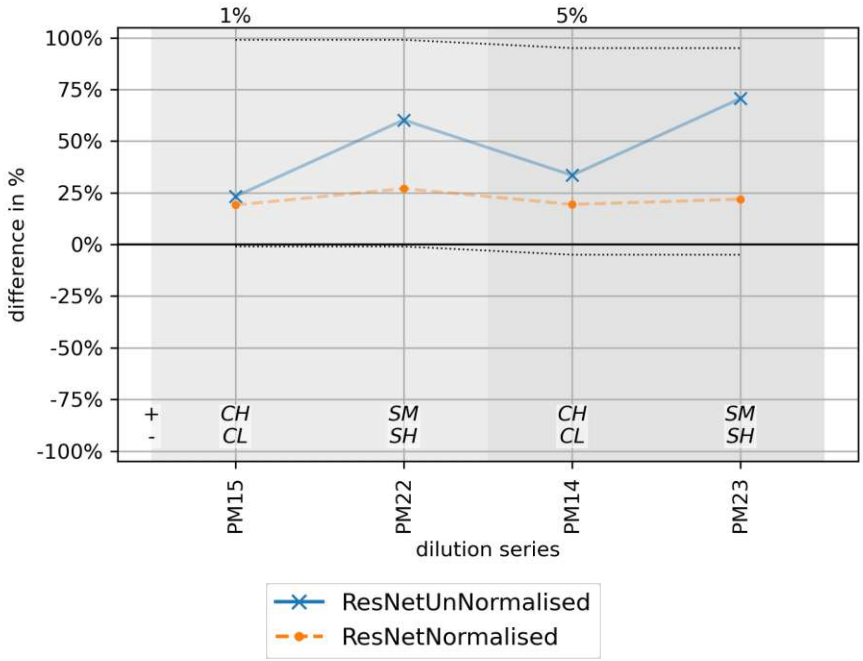


Figure 8.2: Differences of predicted ALT+ rate and true ALT+ ratio of dilution series in $\mathcal{D}_{sub}$. The Figure shows the differences for ResNet using image normalisation and the hyperparameters $(\alpha, \lambda, S) = (0.001, 0.005, 32)$, as well as ResNet without normalisation and $(\alpha, \lambda, S) = (0.001, 0.01, 32)$. The shaded background corresponds to the true ALT+ ratio, which is also indicated on top of the figure. At the bottom of the figure, we find the cell lines of each dilution series in two separate rows with the following encoding: the top row (+) records the ALT+ cell lines SK-N-MM (SM) or CHLA90 (CH). In the bottom row (-), we find the ALT− cell lines SK-N-SH (SH) or CLB-MA (CL).

augmentation does not provide more robust MyNet models on new data, as setup 2. provides the worst results. Using only batch normalisation, the simple setup 1. gives decent results on $\mathcal{T}_{test}$ as well as accurate predictions on $\mathcal{D}_{sub}$. We note that this setup provides slightly worse results on $\mathcal{T}_{test}$ than the optimal MyNet model of Section 8.2 without robustification, see Table 8.1, but it gives much better predictions on $\mathcal{D}_{sub}$. Hence, for the final implementation of MyNet, we use the robustification setup 1. and the hyperparameters as discussed in Section 8.2.

Table 8.3b shows the corresponding results for ResNet. We find that setup 2. improves prediction on $\mathcal{D}_{sub}$ compared to the optimal ResNet model of Section 8.2 without robustification, see Table 8.1. However, using all three data augmentation techniques of halo effect, random blurring and sharpening according to setup 3. provides the best predictions both on $\mathcal{T}_{test}$ and particularly on $\mathcal{D}_{sub}$. In particular, employing the robustification methods in setup 3. improves the results on $\mathcal{D}_{sub}$ of the optimal ResNet model of Section 8.2. Hence, for Chapter 9, we finally implement ResNet by using the

Setup	Epochs Until Early Stopping	Accuracy on $\mathcal{T}_{test}$	Absolute Difference Between Predicted and True ALT+ Ratio on $\mathcal{D}_{sub}$			
			PM14	PM15	PM22	PM23
1.	48	0.924	0.004	0.021	0.043	0.001
2.	44	0.905	0.138	0.163	0.369	0.337
3.	50	0.926	0.036	0.074	0.186	0.101
4.	34	0.933	0.062	0.099	0.236	0.163
5.	52	0.929	0.044	0.082	0.212	0.156

(a) MyNet

Setup	Epochs Until Early Stopping	Accuracy on $\mathcal{T}_{test}$	Absolute Difference Between Predicted and True ALT+ Ratio on $\mathcal{D}_{sub}$			
			PM14	PM15	PM22	PM23
2.	69	0.940	0.125	0.125	0.188	0.131
3.	82	0.941	0.078	0.084	0.134	0.076

(b) ResNet

Table 8.3: MyNet and ResNet trained using optimal hyperparameters and different robustification setups. As discussed in Section 8.2, for MyNet and ResNet we both use normalisation as well as the hyperparameters $(\alpha, \lambda, S) = (0.0001, 0.005, 64)$ and $(\alpha, \lambda, S) = (0.001, 0.005, 32)$, respectively. The table highlights the accuracy score on $\mathcal{T}_{test}$ as well as absolute differences between predicted and actual ALT+ rates on $\mathcal{D}_{sub}$. For better reference, we highlight the chosen setups for the final models in bold font.

robustification setup 3. and the hyperparameters as discussed in Section 8.2.

In summary, we find that rather simple robustification techniques such as halo effect, batch normalisation and random blurring or sharpening provide most predictive models on $\mathcal{T}_{test}$ as well as $\mathcal{D}_{sub}$. While for MyNet batch normalisation was sufficient, ResNet required all of the afore-mentioned techniques to provide decent performance.

8.4 Summary

In this chapter, we discussed three technical research questions to setup the IBA models based on our methodology of Section 6.7. We found that image normalisation considerably improves the model predictions of MyNet and ResNet and identified optimal hyperparameters of learning rate, batch size and regularisation parameter λ of `torch.optim.AdamW`. Furthermore, we established that ResNet performed best on previously unseen testing data when training it with the data augmentation techniques halo effect, random blurring and sharpening as well as using batch normalisation. Conversely, for MyNet, batch normalisation was sufficient to get best results.

CHAPTER 9

Results

FGA or IBA models have not yet been used for ALT classification in the literature. Based on our two main research questions RQ1 and RQ2, we want to discuss in this thesis how well our final FGA and IBA models allow for classifying the ALT status of neuroblastoma cells. Our research questions cover two main aspects to assess the quality of ALT classification:

1. How well do the models predict the ALT status on previously unseen data (RQ1)? This aspect requires us to analyse the models' performances on testing data $\mathcal{T}_{test}$ as well as on dilution series of $\mathcal{D}$. Classifying cells on $\mathcal{D}$ is particularly important as one predicts the ALT status in practice for image series that are not part of the training data. Hence, the model results on $\mathcal{D}$ provide best indications how the models will perform for practical purposes. In that respect, it is also important to analyse how confident the models are with their predictions.

2. How do the models predict the ALT status and which cell properties are important (RQ2)? Finding these image-derived criteria to describe the ALT status of cells is important to explain the models' decisions. This will help users to understand why the models predict a given result and may also foster the users' confidence in the models' decisions. As opposed to "black box" methods, we can explain why the models arrived at a specific prediction.

Answering our two main research questions RQ1 and RQ2 allows us to evaluate whether our FGA and IBA models may serve as candidates for new state of the art methods in ALT classification. Certainly, as outlined in the last chapter of this thesis, we find our results in a very controlled setting and further research is necessary to corroborate them for image series of more diverse origin and recording situations.

Our findings in Chapters 7 and 8 are numerous and corroborate how we set up the data samples and train the FGA and IBA models. The first Section 9.1 summarises these setups for easier reference. For the first part of research question RQ1, Section 9.2 outlines the model performances on the training sample $\mathcal{T}_{test}$. Afterwards, Sections 9.3 and 9.4 clarify which FGA features and image properties are important when assigning the ALT status in FGA and IBA models to answer research question RQ2. For the second part of research question RQ1, Section 9.5 discusses in detail how the trained models perform on the dilution series of $\mathcal{D}$, which we did not use in the training samples. Finally, Section 9.6 analyses how confident the models are when predicting the ALT status in Section 9.5, which is also related to research question RQ1.

9.1 Final Data Preparation Steps and Pipelines

In this section, we summarise how we finally setup the samples $\mathcal{T}, \mathcal{T}_{IBA}, \mathcal{T}_{test}, \mathcal{V}$ to train nuclear and serial FGA models as well as the IBA models on nucleus level. Furthermore, we state any preprocessing steps that we apply according to the results of Chapters 7 and 8 and based on our methodology of Chapter 6.

We choose the training, validation and testing samples in line with our methodology of Section 6.5 and findings of Chapter 7. In particular, we build $\mathcal{T}, \mathcal{T}_{test}$ based only on nuclei of the pure dilution series P3∼A, P6, P8, P9WH, P10, P11, P12, P13 if they feature at least two telomere spots, see Sections 6.5, 7.1 and 7.2.1. We ensure using balanced cell lines and targets (i.e. 50% ALT+ nuclei), resize $\mathcal{T}$ to 20,000 nuclei and split the data into $\mathcal{T}$ and $\mathcal{T}_{test}$ according to an 80%-20% split on field of view level, see Section 7.1. As mentioned in Section 6.5, we then partition $\mathcal{T}$ into $\mathcal{T} = \mathcal{T}_{IBA} \cup \mathcal{V}$ according to another 80%-20% split, where we again ensure that $\mathcal{T}_{IBA}$ and $\mathcal{V}$ feature the same ALT+ ratio (i.e. 50%).

For the nuclear and serial FGA models, we do not normalise the nuclear images before we extract the 14 robust features that we will use for training, see Sections 7.2.2 and 7.2.3. Afterwards, we scale features according to Section 7.1. The nuclear FGA models use the optimal hyperparameters of Section 7.2.4 and we employ the logistic regression model of Section 7.2.5 with a limited number of six features. We apply the Wasserstein distance model as given in Section 7.2.6.

For the IBA models, we preprocess images by normalising the nuclear images according to Section 8.2 before taking them as inputs of the deep learning networks of Section 6.7. Furthermore, for training, we apply the optimal hyperparameters of MyNet and ResNet as discussed in Section 8.2. Similarly, we use the best robustification techniques as given in Section 8.3. As discussed in Sections 6.5, 6.7 and 8.1, we will use only the telPNA channel to train and apply MyNet, while we use all three channels (telPNA, nuclear segmentation mask, DAPI channel) for ResNet.

For the dilution series of $\mathcal{D} = \{\mathcal{D}_{P4\sim A}, \mathcal{D}_{P7}, \mathcal{D}_{PM1}, \mathcal{D}_{PM3}, \ldots, \mathcal{D}_{PM24}\}$, we use the same sampling and preprocessing steps as discussed above for $\mathcal{T}, \mathcal{T}_{test}, \mathcal{T}_{IBA}, \mathcal{V}$. In particular,

similarly to $\mathcal{T}$, we only include nuclei with a minimum number of two telomere spots and reduce the number of nuclei of each dilution series in $\mathcal{D}$ to 4,000. Furthermore, we only include nuclei if the post-processing model of Section 7.3 predicts that they are correctly segmented. In addition, we use the scaler that we estimated on $\mathcal{T}$ to also scale the FGA features of all dilution series in $\mathcal{D}$. For the IBA models, we normalise the nuclear images before using them as inputs for the networks.

9.2 Performance on $\mathcal{T}_{test}$

In line with research question RQ1, we want to assess how well the trained nuclear and serial FGA models as well as the IBA models predict the ALT status of nuclei on the training sample $\mathcal{T}_{test}$. In Section 6.4.1, we refer to this problem as ALT classification on nucleus level. Note that $\mathcal{T}_{test}$ includes nuclei of the same dilution series as the training sample $\mathcal{T}$ but $\mathcal{T}_{test}$ is disjoint of $\mathcal{T}$, see Section 9.1. In line with our decision criteria of Section 6.8, we determine accuracy scores on $\mathcal{T}_{test}$ for all models and also assess the accuracy on $\mathcal{T}_{test}$ for each of the four cell lines SK-N-SH, SK-N-MM, CHLA90, CLB-MA, separately. Accuracy scores on cell line level allow us to also infer false positive and false negative rates. Note that research question RQ1 also includes performance on $\mathcal{D}$, which we cover in Section 9.5.

Determining the ALT status of nuclei on $\mathcal{T}_{test}$ is ALT classification on nucleus level, which is infeasible for the Wasserstein distance model. Still, we can apply the Wasserstein distance model on $\mathcal{T}_{test}$ to determine the overall ALT+ rate, which should ideally be close to 50% according to Section 9.1. To come up with an accuracy score for the Wasserstein distance model, we first apply the prediction algorithm of the Wasserstein distance models in Section 6.6.3 on $\mathcal{AS}_{fine}$ instead of $\mathcal{AS}$. This approach allows us to predict ALT+ rates on $\mathcal{T}_{test}$ and on cell line level on a much more granular scale (e.g. by predicting an ALT share of 68% instead of 75% ALT+). We can then determine accuracy scores of the Wasserstein distance model on cell line level, as the true ALT+ rate is either 0% or 100% depending on whether the cell line is ALT+ or ALT−. By using the accuracy scores for separate cell lines, we can easily infer the overall accuracy score on $\mathcal{T}$ by taking their average. Note that this accuracy score is only a surrogate for the Wasserstein distance model, as this model does not work on nucleus level but with feature distributions on level of the *whole* sample of $\mathcal{T}_{test}$. Generally speaking, the more nuclei contribute to the feature distribution, the more accurate the distribution and the predictions of the Wasserstein distance model are. Hence, figuratively, the model will be better on $\mathcal{T}_{test}$ than the sum of its parts on cell line level. As a result, we note that comparing this score with the accuracy scores of the other models has to be taken with a grain of salt, since our assumptions favour lower score values.

Table 9.1 summarises accuracy scores on $\mathcal{T}_{test}$ and on cell line level of $\mathcal{T}_{test}$ for all FGA and IBA models. Note that $\mathcal{T}_{test}$ is balanced. Hence, to put the accuracy scores in perspective, we see that the baseline score would be 50%. We find that ResNet and the Wasserstein distance model gives the most accurate results on $\mathcal{T}_{test}$ and that the

IBA models predict in general better than the nuclear FGA models. When looking at the results on cell line level, we find that the Wasserstein distance model gives perfect predictions for the cell lines SK-N-MM, SK-N-SH and CLB-MA, but performs worst for CHLA-90. In general, we note that all models struggle most with CHLA-90 compared to the other cell lines. According to experts of the CCRI, this can be due to lower signal intensity and sharpness among the nuclear images of CHLA-90 cells in $\mathcal{T}_{test}$. Furthermore, we see that SK-N-SH seems to be harder to classify correctly as ALT− than CLB-MA. After analysing the images of the corresponding SK-N-SH dilution series, experts of the CCRI think that stronger cytoplasmic background in the telPNA channel may cause these problems, particularly for dilution series P12, see also Section 6.3. Hence, in Section 9.5 below, we can already expect most models to overestimate the ALT+ rate for dilution series that contain high ratios of SK-N-SH cells.

Looking at the predictions on cell line level, we note that FGA models exhibit a false positive rate of up to 16%-22% and a false negative rate of up to 19%-23%. IBA models show considerably lower false positive and false negative rates of 7%-9% and 8%-12%, respectively. The Wasserstein distance model excels with a false positive rate of 0%, but also shows a relatively high false negative rate of up to 28%.

Model	**Accuracy on $\mathcal{T}_{test}$**	**Accuracy on Cell Line**			
		SM	**SH**	**CH**	**CL**
RandomForestClassifier	0.862	0.870	0.809	0.805	0.962
LogisticRegression	0.838	0.825	0.777	0.774	0.973
XGBClassifier	0.860	0.864	0.812	0.800	0.962
SVC	0.852	0.841	0.838	0.768	0.961
Wasserstein Distance Model	**0.93***	**1.00**	**1.00**	0.72	**1.00**
MyNet	0.924	0.931	0.909	0.876	0.984
ResNet	**0.941**	0.931	0.932	**0.922**	0.983

Table 9.1: Accuracy scores of FGA and IBA models on $\mathcal{T}_{test}$ as well as on level of cell lines SK-N-MM (SM), SK-N-SH (SH), CHLA90 (CH) and CLB-MA (CL) on $\mathcal{T}_{test}$. The Wasserstein distance model predicts an ALT+ share of 48% on $\mathcal{T}_{test}$ and we find an accuracy score of 93% under the assumptions mentioned in Section 9.2. To highlight that comparing this score with the scores of the other models has to be taken with a grain of salt, we mark it with an asterisk *. We also marked best results in bold font.

9.3 Feature Importance

As discussed in Section 7.2.2, there are 14 features that have proven to be stable across specific pure and actual dilution series of $\mathcal{D}_{sub}$ and $\mathcal{T}$. It yet remains analysing how well each feature explains the ALT status of a nucleus in line with research question RQ2. In this section, we want to discuss the feature importance scores for FGA models. We start by determining how we can define feature importance scores for nuclear FGA models according to Section 3.4:

- *Logistic Regression*: as we have scaled the features in $\mathcal{T}$, we can take the β coefficient of the logistic regression as indicator for how important the model judges the features. If $|\beta_i|$ is big, the corresponding i-th feature is important. If $|\beta_i| \approx 0$, the relevant feature is unimportant.
- *Support Vector Machine*: we can use so-called permutation importance to determine how important a feature is to classify the ALT status [Bre01]. For a specific feature, we permute its values in $\mathcal{T}$ and keep the other features and the ALT status label unchanged. We then apply the resulting data to the trained support vector machine. For important features, we expect a significant drop of the accuracy score, while unimportant features will merely affect it. When repeating this algorithm for all features and multiple permutations, we can determine how important a feature is.
- *Random Forest*: as discussed in Section 3.4, building the decision trees of a random forest amounts to finding optimal splits of features that minimise Gini impurity. We use `sklearn`'s feature importance method that is based on the impurity decrease of a feature averaged over all trees [PVG+11].
- *Gradient Boosting*: similarly to random forests, we calculate the variable importance of each variable (that defines certain nodes) in a single gradient boosting tree by the amount that each node split improves a performance measure (in `XGBoost` by default information gain), weighted by the number of observations that the node covers. One then averages the feature importances across all decision trees within the gradient boosting model, see also Chapter 10.13 of [HTF09].

Using above's methodology, we can calculate the importance of each of the 14 features in the nuclear FGA models and rank them for each model separately. Figure 9.1 depicts these ranks for all nuclear FGA models of Section 3.4.

To simplify the following discussion, we look at the top 4 ranks of each model. Hence, the most important features are the following:

1. *cluster shade* and *cluster prominence*: for all nuclear FGA models, cluster shade and cluster prominence are very important for predicting the ALT status. Indeed, ALT+ nuclei in $\mathcal{T}$ show on average greater values for cluster shade and cluster prominence than ALT− nuclei. Biologically, this is because ALT+ nuclei usually show greater, brighter and more heterogeneous spots in the telPNA channel, see Section 2.2. Note that cluster shade and cluster prominence are highly positive correlated according to Figure 7.4.
2. *kurtosis* and *skewness*: for all nuclear FGA models, at least one covariate of the highly correlated features kurtosis and skewness shows high feature importance. ALT+ nuclei in $\mathcal{T}$ exhibit on average greater values for kurtosis and skewness than ALT− nuclei, which corresponds to longer tails of the telPNA intensity distribution. Biologically, this makes sense since we expect long tailed intensity distributions of ALT+ nuclei due to more intense (e.g. ultra-bright) spots, see Section 2.2.

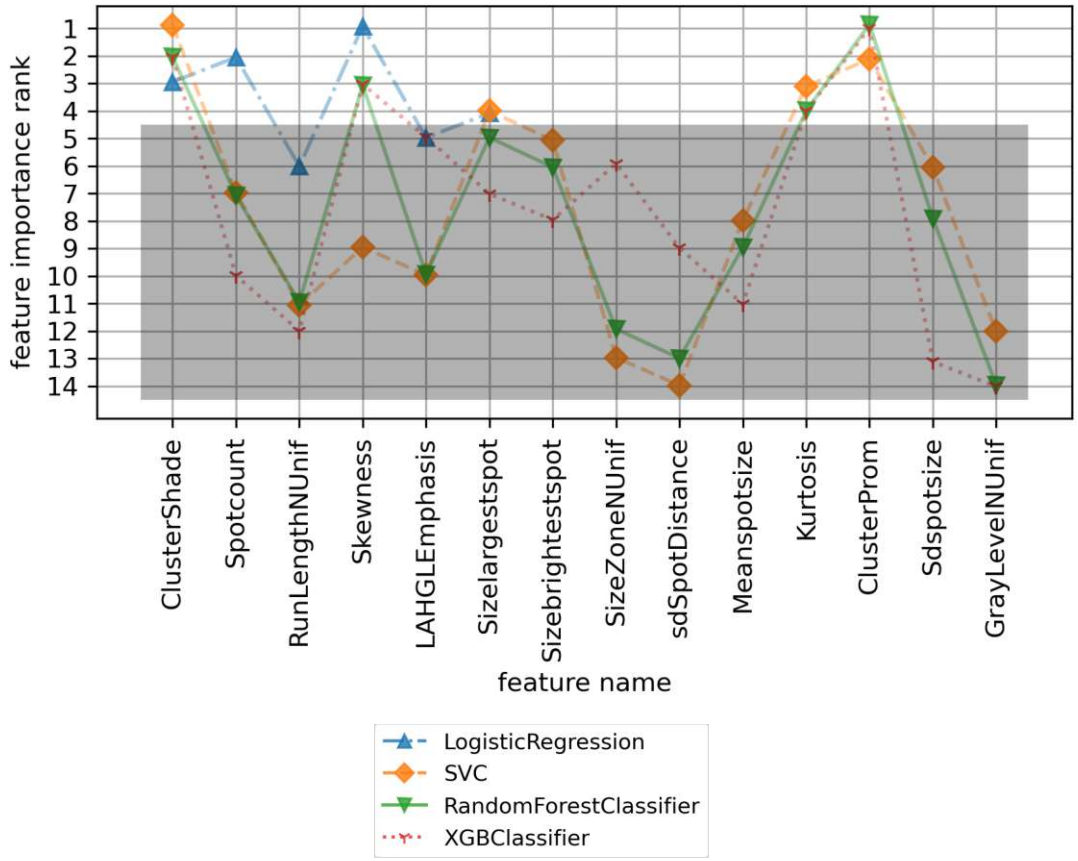


Figure 9.1: Robust features of Section 7.2.2 ranked according to their importance in the nuclear FGA models from 1 (high importance) to 14 (low importance). Note that the logistic regression model uses only the six features on the left of the figure according to Section 7.2.5. For easier reference, ranks below 4 are grayed out and feature names are abbreviated.

3. *size largest spot*: for the support vector machine and logistic regression model, the size of the largest spot of a nucleus is important to classify the ALT status. As discussed in Section 2.2, this is in line with our expectations, as big (usually ultra-bright) spots are more common in ALT+ cells.

4. *spot count*: for the logistic regression model, the number of spots is important to classify the ALT status. Biologically, the number of spots does not relate to the ALT status, see also Section 7.2.1. However, one can argue that the spot segmentation model of Section 6.2 identifies more spots if they are brighter and more pronounced, which is more likely for ALT+ cells. Hence, the logistic regression model possibly confounds *higher number of spots* and *brighter and more pronounced spots* in the telPNA channel, which may explain this spurious association. This observation

corroborates that our decision in Section 7.2.1 of taking the least possible minimum number of two required spots per nucleus was in several aspects correct. Otherwise, we could have unwittingly biased the ALT+ ratio of dilution series in $\mathcal{D}$.

In addition to the afore-mentioned important features for nuclear FGA models, we note that we have selected *cluster prominence* and *standard deviation of spot size* as the best (i.e. most predictive) pair of features in the Wasserstein distance model for ALT classification on series level, see Section 7.2.6. While we discussed the biological relevance for cluster prominence above, we note that ALT+ nuclei exhibit on average greater standard deviations of spot sizes than ALT− nuclei. This can be attributed to bigger ultra-bright spots and in general more heterogeneous telPNA images of ALT+ cells, see Section 2.2.

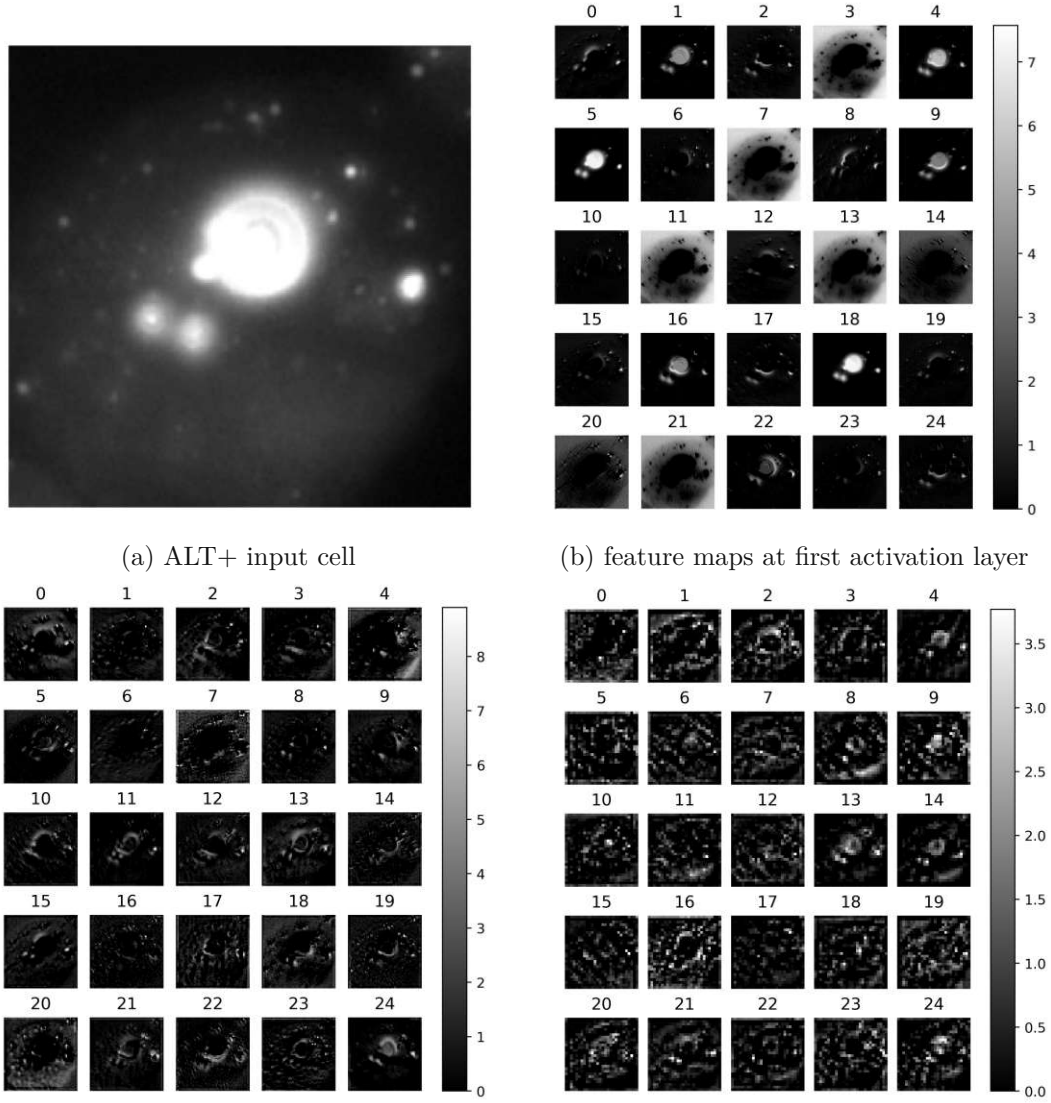
9.4 Visualised Feature Extraction of MyNet

For deep learning models, it is usually not clear how the models came up with a certain prediction. However, as discussed in Section 4.2, CNN use convolutions that preserve the spatial relationship of the input image when extracting features. When applying the CNN to a given image, we can therefore visualise the two dimensional feature maps at each stage of the CNN as images to analyse what kind of image properties the model extracts. It makes most sense to visualise these feature maps at activation layers to keep only the most relevant features.

In line with research question RQ2, this section analyses the extracted features of MyNet for an ALT+ and an ALT− cell in $\mathcal{T}_{test}$ that MyNet correctly classified as ALT+ and ALT−, respectively. Figure 6.4 shows the architecture of MyNet and that it consists in total of nine activation functions. As features become more and more abstract when passing through the network, we will only consider the feature maps at the first, third and fifth ReLU activation layer. Furthermore, for each activation layer, we will randomly select 25 feature maps to simplify the discussion.

For an ALT+ cell of the dilution series P11, Figure 9.2 shows the input image as well as 25 randomly selected feature maps at the first, third and fifth activation layer.

The input image shows several strongly pronounced ultra-bright telomere spots that we can easily identify in multiple feature maps of the first activation layer (e.g. feature map 5). The first activation layer also shows feature maps that seem to focus on the background of the cell (e.g. feature map 3) or less bright spots at different illumination levels (e.g. feature map 15). Feature maps at the third activation layer already incorporate three convolutional layers and therefore include features of a greater receptive view, see Figure 6.4. In the third activation layer, we can again identify the ultra-bright spots as roundish contour lines in the feature maps (e.g. feature map 13) as well as less bright and less pronounced spots (e.g. feature map 6). Again, we note feature maps that partly incorporate background information (e.g. feature map 4). In the fifth activation layer, the features become more abstract as expected. Still, we can easily see the signals of the



(a) ALT+ input cell

(b) feature maps at first activation layer

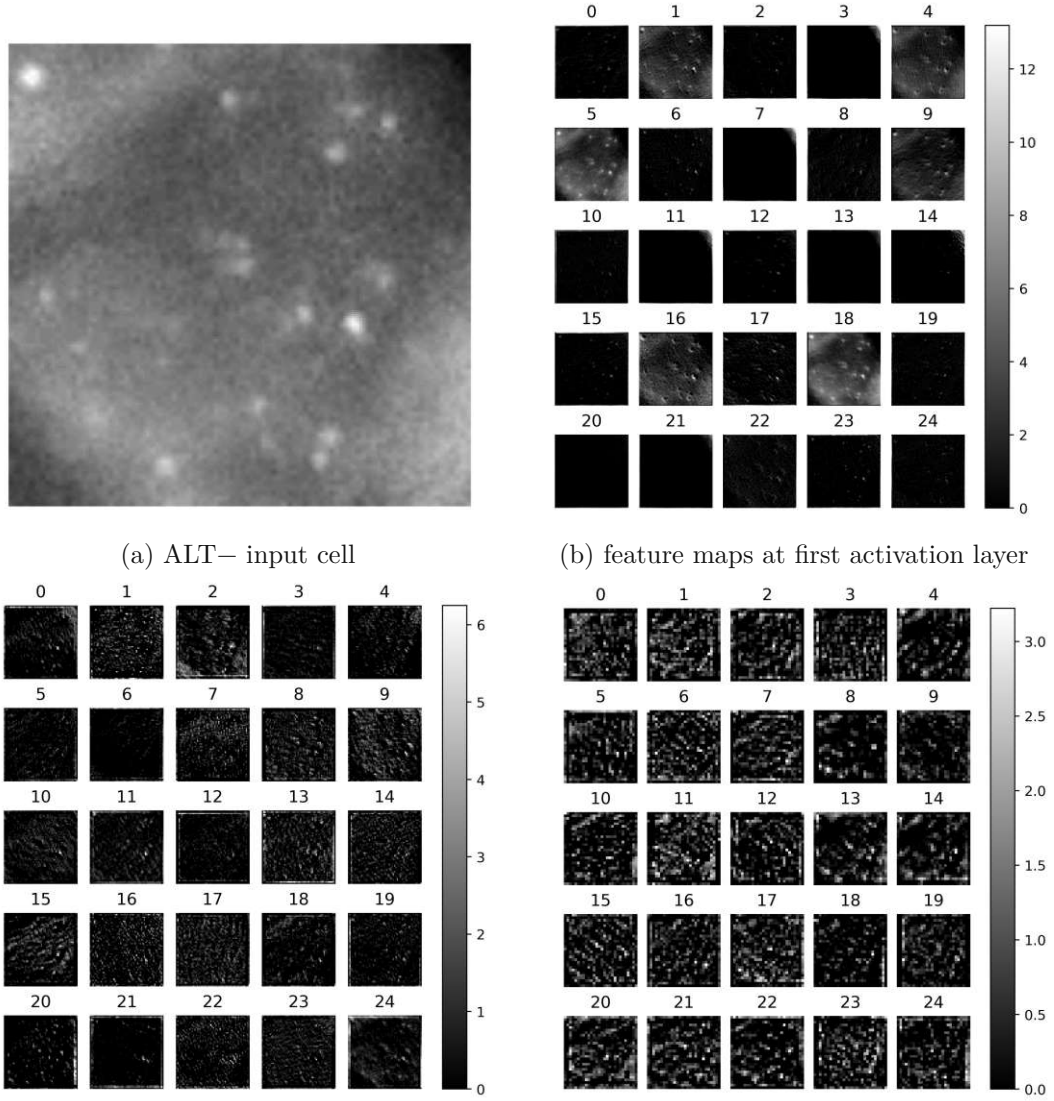
(c) feature maps at third activation layer

(d) feature maps at fifth activation layer

Figure 9.2: Feature maps of the first, third and fifth activation layer of MyNet for an ALT+ cell of dilution series P11 (field of view 507, nucleus 37, cell line SK-N-MM). For each activation layer, we randomly selected 25 feature maps. Note that Figure 9.3 shows features of the same layers.

ultra-bright telomere spots as contour lines (e.g. feature map 13), and clusters of less pronounced telomere spots (e.g. feature map 19).

Figure 9.3 shows the input image of an ALT− cell in the dilution series P9WH and the same 25 randomly selected feature maps of the first, third an fifth activation layer as in Figure 9.2. While we note that the illumination scales partly differ across the images,



(a) ALT− input cell

(b) feature maps at first activation layer

(c) feature maps at third activation layer

(d) feature maps at fifth activation layer

Figure 9.3: Feature maps of the first, third and fifth activation layer of MyNet for an ALT− cell of dilution series P9WH (field of view 487, nucleus 85, cell line SK-N-SH). For each activation layer, we randomly selected 25 feature maps. Note that Figure 9.2 shows feature of the same layers.

we can still compare the feature maps of Figures 9.2 and 9.3. In Figure 9.3, we see that the ALT− cell does not show many telomere spots that are well in contrast to the background. For that reason, the feature maps at the first activation layer seem to mainly extract features at the same brightness intensity level (e.g. feature map 5). We recognise telomere spots in some feature maps (e.g. feature map 17). Compared to Figure 9.2, the

features of the third and fifth activation layer in Figure 9.3 appear less coherent. We can imagine some feature maps extracting telomere spots and contiguous telPNA clusters (e.g. feature maps 21 and 13 of the third and fifth activation layer, respectively). We note that the overall intensity at these activation layers is lower than in Figure 9.2.

Summarising, Figures 9.2 and 9.3 indicate that MyNet uses features that identify greater roundish structures such as pixel clusters in the telPNA channel at varying illumination levels. In particular, some features incorporate the size and intensity of spots that are well in contrast to the background. When combining multiple features of spots at different brightness intensity levels, the model is also able to capture the heterogeneity of the telPNA channel. We can therefore see that MyNet extracts features that appear similar to the most relevant FGA features *cluster prominence* and *cluster shade*, *size of the largest spot*, *kurtosis* and *skewness* as well as *standard deviation of spot size*, which we discuss in Section 9.3.

9.5 Prediction Results on $\mathcal{D}$

As part of research question RQ2, we apply in this section the FGA and IBA models on the dilution series of $\mathcal{D}$ to compare predicted and actual ALT+ rates. In Section 6.4.2, we referred to this problem as ALT classification on series level. For nuclear FGA models and IBA models, we apply the method discussed at the beginning of Section 6.6.3 to determine the ALT+ ratio of a dilution series. While none of the dilution series in $\mathcal{D}$ is part of $\mathcal{T}, \mathcal{T}_{test}, \mathcal{T}_{IBA}, \mathcal{V}$, note that the sub-family $\mathcal{D}_{sub} = \{PM14, PM15, PM22, PM23\} \subseteq \mathcal{D}$ is strictly speaking not completely out-of-sample. This is because we used $\mathcal{D}_{sub}$ to select stable features of the FGA models in Section 7.2.2 and to choose hyperparameters and robustification methods for the IBA models in Sections 8.2 and 8.3.

In line with our decision criteria of Section 6.8, Figure 9.4 shows differences between the predicted and actual ALT+ rate for each dilution series of $\mathcal{D}$ and each model. Figure 9.5 illustrates the corresponding differences on the level of predicted ALT+ classes $\mathcal{G} = \{0\%, 1\%, 5\%, \ldots, 75\%, 100\%\}$.

For both figures, we split the information in two separate plots of identical scaling to avoid cluttered illustrations. For Figure 9.4 we used the predictions of the Wasserstein distance model on $\mathcal{AS}_{fine}$, as discussed in Section 9.2, while for Figure 9.5 we use the predictions on $\mathcal{AS}$. Moreover, to avoid misunderstandings, we emphasise that Section 8.3 already shows the results of the IBA models on $\mathcal{D}_{sub}$. To summarise the information of Figures 9.4 and 9.5, Table 9.2 shows the sum of absolute differences in predicted ALT+ rates and in predicted ALT+ groups across $\mathcal{D}$ for all models separately.

Based on Figures 9.4 and 9.5 as well as Table 9.2, we see that the Wasserstein distance model predicts the ALT+ rates and ALT+ classes $\mathcal{G}$ by far the best. For all dilution series in $\mathcal{D}$, the Wasserstein distance model predicts ALT+ rates that are at most one class off from the true ALT+ rates. We also see that the Wasserstein distance model

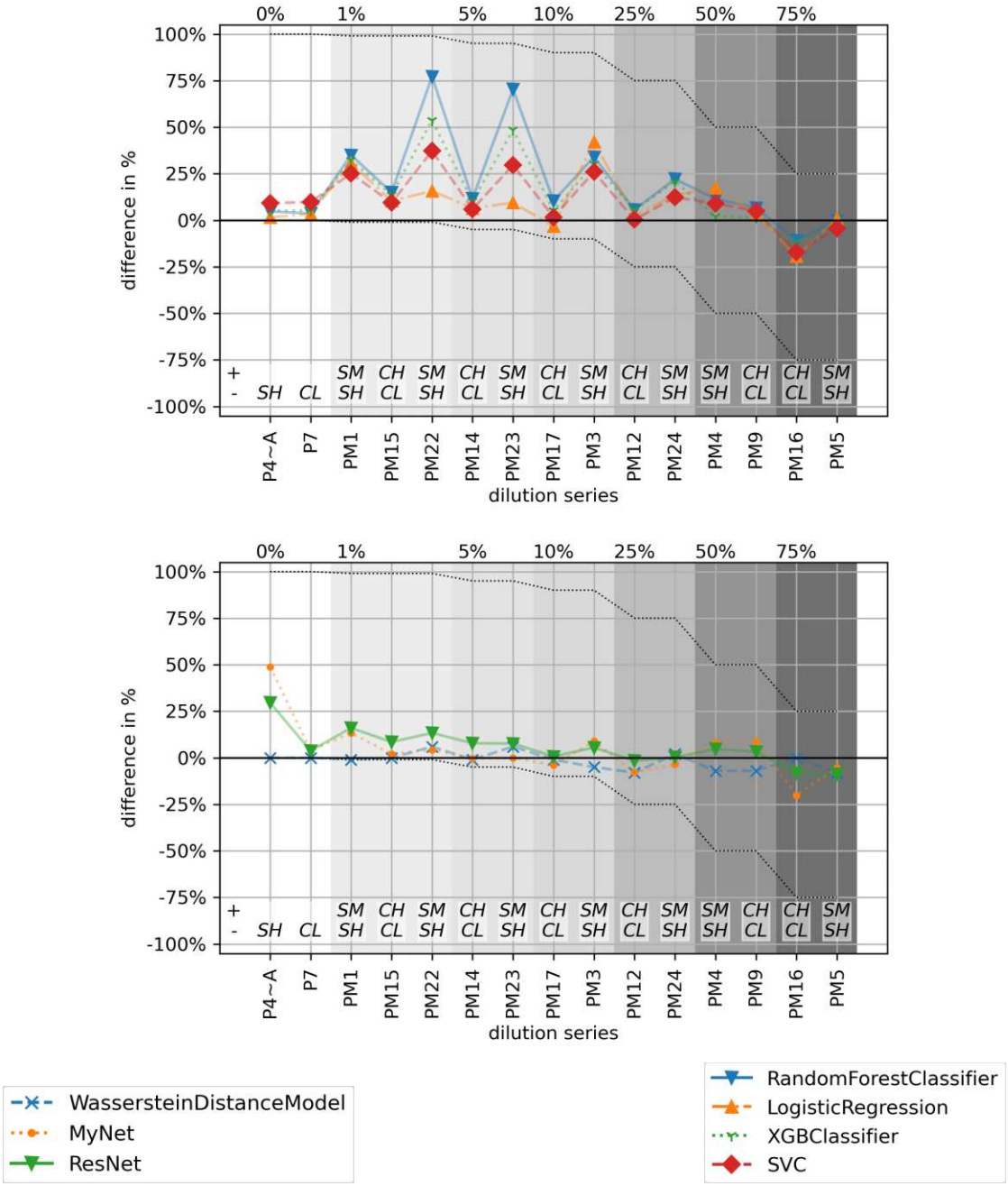


Figure 9.4: Differences of predicted and true ALT+ ratio of dilution series in $\mathcal{D}$ for nuclear FGA models (above) as well as IBA models and Wasserstein distance model (below). The dotted lines on top and at the bottom show the worst possible differences for each dilution series. The shaded background corresponds to the true ALT+ ratio, which is also indicated on top of the figure. At the bottom of the figure, we find the cell lines of each dilution series in two separate rows with the following encoding: the top row (+) records the ALT+ cell lines SK-N-MM (SM) or CHLA90 (CH). In the bottom row (-), we find the ALT− cell lines SK-N-SH (SH) or CLB-MA (CL).

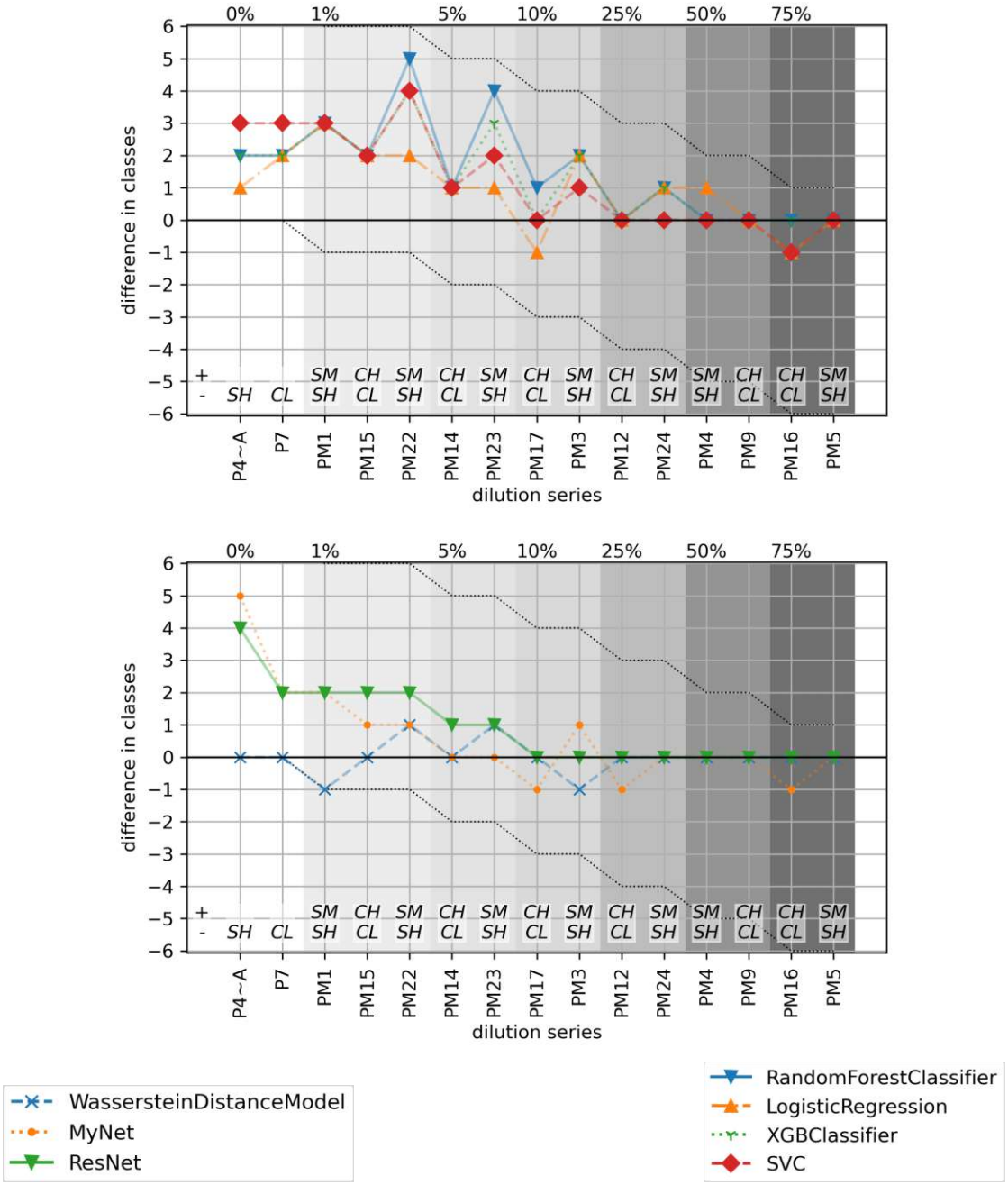


Figure 9.5: Differences of predicted and true ALT+ class of $\mathcal{G}$ for dilution series of $\mathcal{D}$ separately for nuclear FGA models (above) and IBA models and Wasserstein distance model (below). The gray dotted lines indicate the worst possible deviations in terms of number of classes of $\mathcal{G}$. The dotted lines on top and at the bottom show the worst possible differences for each dilution series. The shaded background corresponds to the true ALT+ ratio, which is also indicated on top of the figure. At the bottom of the figure, we find the cell lines of each dilution series in two separate rows with the following encoding: the top row (+) records the ALT+ cell lines SK-N-MM (SM) or CHLA90 (CH). In the bottom row (-), we find the ALT− cell lines SK-N-SH (SH) or CLB-MA (CL).

correctly identified the ALT+ rate of 0% for the dilution series P4∼A and P7, which confirms the false positive rate of 0% from Section 9.2.

With the exception of PM22 and PM23, the nuclear FGA models predict ALT+ rates quite similarly. Overall the logistic regression model predicts ALT+ rates and classes best. We also note that the nuclear FGA models provide in general worse results for dilution series of the cell lines SK-N-SH and SK-N-MM (such as PM22 and PM23). This is in line with our observation in Section 9.2 regarding lower accuracy scores for SK-N-SH cells. We ascertain false positive rates of more than 5% on P4∼A and P7 and note stable predictions of at most one ALT+ class difference for dilution series with ALT+ ratios of more than 10%.

For the IBA models, we see that they perform better than the FGA models and that ResNet gives slightly more stable predictions than MyNet. We also note that both ResNet and MyNet failed in predicting the ALT+ rate of 0% in the dilution series P4∼A. Experts of the CCRI validated that this dilution series is of worse quality as compared to the other dilution series, since the telPNA channel is mostly out of focus with weak signals. Still, we note that all nuclear FGA models and the Wasserstein distance model are able to cope better with this worse image quality and predict more accurately on P4∼A than the IBA models. Both MyNet and ResNet give stable predictions of at most one ALT+ class difference for dilution series with ALT+ ratios greater than 1%.

Summarising, we find that the Wasserstein distance model excels in predicting the ALT+ rates for dilution series in $\mathcal{D}$, including series of worse image quality and varying fluorescence staining (e.g. P4∼A, PM23). Furthermore, we established that IBA models mostly give better results than the other nuclear FGA models. We believe that the Wasserstein distance model outperforms the other approaches, as it is based on feature distributions of all nuclei in the relevant dilution series, which appears to stabilise the predictions.

Model	$\sum$ **Abs. Differences ALT+ rates**	$\sum$ **Abs. Differences ALT+ classes**
RandomForestClassifier	3.19	23
LogisticRegression	1.79	18
XGBClassifier	2.48	20
SVC	2.02	20
Wasserstein Distance Model	**0.15**	**4**
MyNet	1.40	15
ResNet	1.20	14

Table 9.2: Summary statistics of Figures 9.4 and 9.5 for each model. The second column sums the absolute differences of ALT+ rates, which we see in Figure 9.4, across all dilution series in $\mathcal{D}$ and for each model separately. The third column sums the absolute differences of ALT+ classes in $\mathcal{G}$, which we see in Figure 9.5, across all dilution series in $\mathcal{D}$ and for each model separately. We marked the best results in bold font.

9.6 Prediction Confidence

When considering the results of Section 9.5, we are usually interested in the models' confidence when predicting the ALT status on nucleus or series level. This confidence relates to research question RQ1, as it allows us to judge the prediction quality. In this section we want to discuss the prediction confidence for all FGA and IBA models.

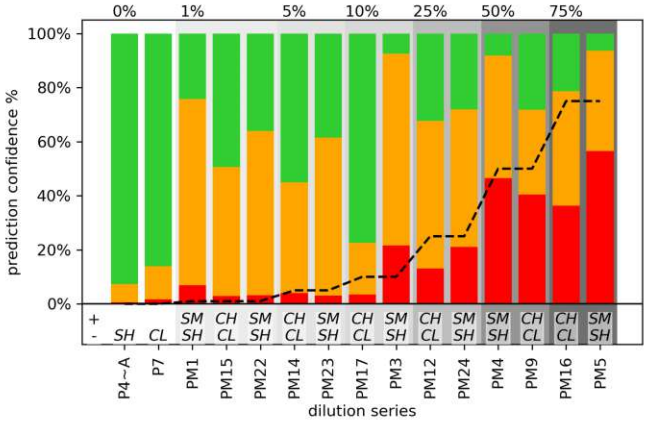
For the nuclear FGA models and IBA models, Figures 9.6 and 9.7 show the distribution of sure ALT+, sure ALT− and unsure assignments across $\mathcal{D}$ according to the criteria of Section 6.8.1. We generally note that the IBA models ResNet and particularly MyNet are much more confident in their predictions than FGA models. This could be because the image-derived features in MyNet and ResNet are more descriptive than the manual features of the FGA models. For support vector machines, random forests and gradient boosting, we observe lower confidence when predicting the ALT status of nuclei in dilution series of cell lines SK-N-MM and SK-N-SH. Indeed, the models predict the ALT+ ratio in these dilution series poorly, see Section 9.5. The logistic regression model is mostly less sure about its predictions than the other FGA models. Still, we note that its rate of sure ALT+ assignments corresponds quite well to the actual ALT+ ratio. This also holds true for ResNet, which is less sure about the (overall wrongly predicted) ALT+ rate in P4∼A than MyNet.

Based on the criteria to determine prediction confidence of Wasserstein models in Section 6.8.1, we find in Figure 9.8 that the Wasserstein distance model is mostly confident when assigning the ALT+ rate. Only for four dilution series, the model is unconfident. For two of these dilution series, the Wasserstein distance model wrongly predicted the ALT+ rate, and the remaining two series consist of cell lines CHLA-90 and CLB-MA, for which the model predicts the ALT+ ratio less accurately, see Section 9.2. Furthermore, with the exception of one dilution series (PM22), we also note that the first and second closest predicted classes π_1, π_2 are at most one class off from the true ALT+ class. Still, we also see that the confidence ratios for the dilution series PM3 and PM23 are high, although the predicted ALT+ class is in fact not correct.

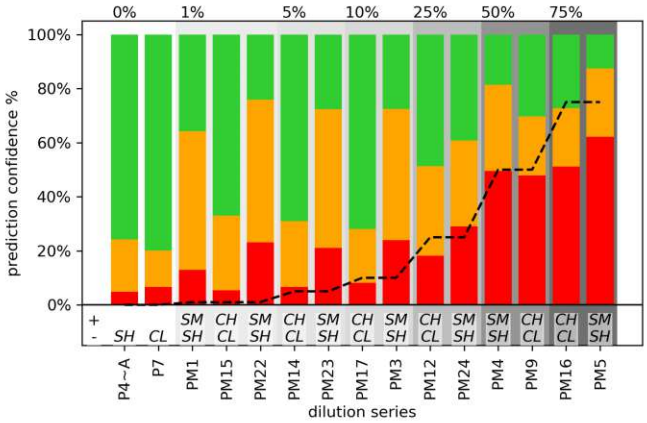
The results underline that considering the models' prediction confidence is valuable for analysing how trustworthy the model outputs are. This holds particularly for the logistic regression model, the Wasserstein distance model and ResNet.

9.7 Summary

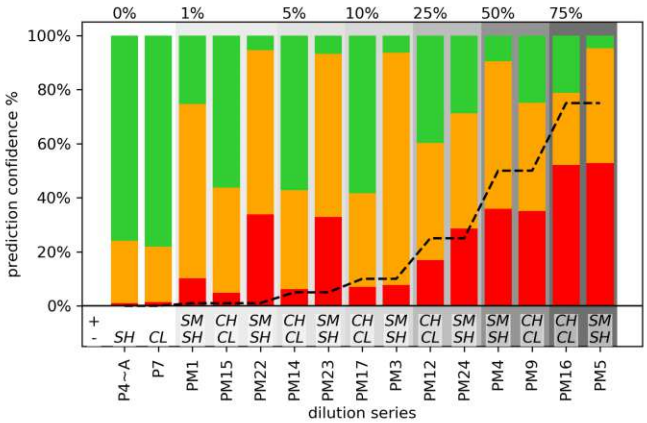
In this section, we evaluated the final FGA and IBA models on the testing samples $\mathcal{T}_{test}$ and $\mathcal{D}$ to answer our main research questions RQ1 and RQ2. We evaluated the model performances on $\mathcal{T}_{test}$ and learned that both ResNet and the Wasserstein distance model provided the best overall accuracy results. On cell line level, the IBA models show low false positive and false negative rates, while the Wasserstein Distance model gave a perfect false positive rate of 0%, but a relatively high false negative rate.



(a) Logistic Regression

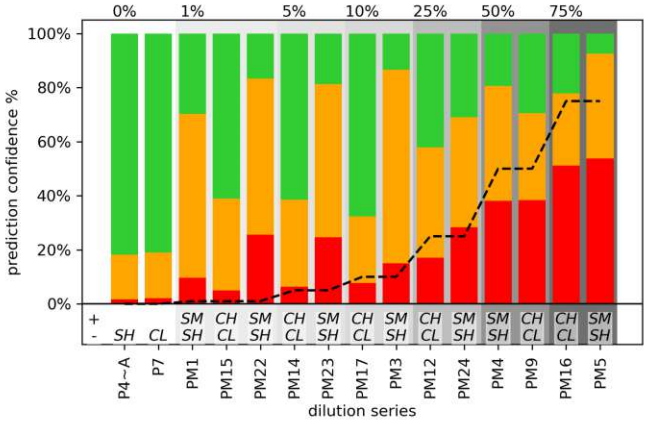


(b) Support Vector Machine

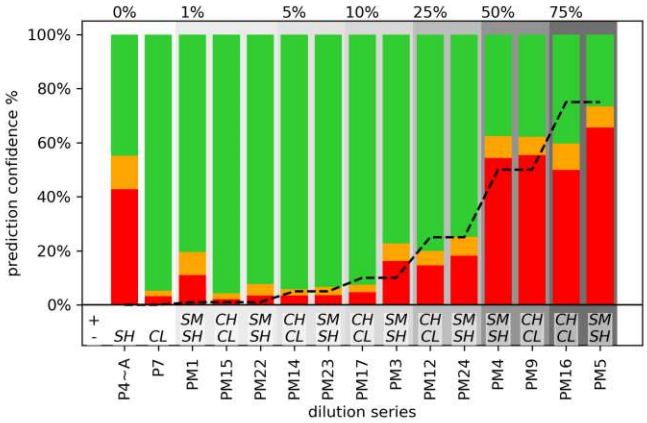


(c) Random Forest

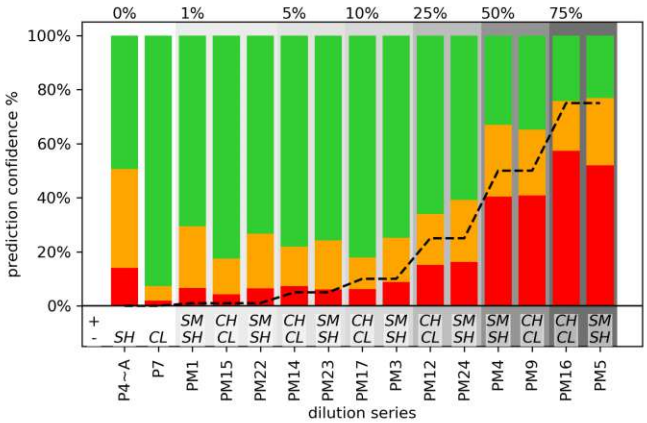
Figure 9.6: Prediction confidence across dilution series in $\mathcal{D}$. For each dilution series, the figures display the share of sure ALT+ (bottom, red), sure ALT− (top, green) and unsure assignments (middle, orange) according to Section 9.6. Black dashed lines refer to the actual ALT+ rate for easier reference. See Figure 7.2 for a description of the remaining elements.



(a) Gradient Boosting, XGB



(b) MyNet



(c) ResNet

Figure 9.7: Prediction confidence across dilution series in $\mathcal{D}$. For each dilution series, the figures display the share of sure ALT+ (bottom, red), sure ALT− (top, green) and unsure assignments (middle, orange) according to Section 9.6. Black dashed lines refer to the actual ALT+ rate for easier reference. See Figure 7.2 for a description of the remaining elements.

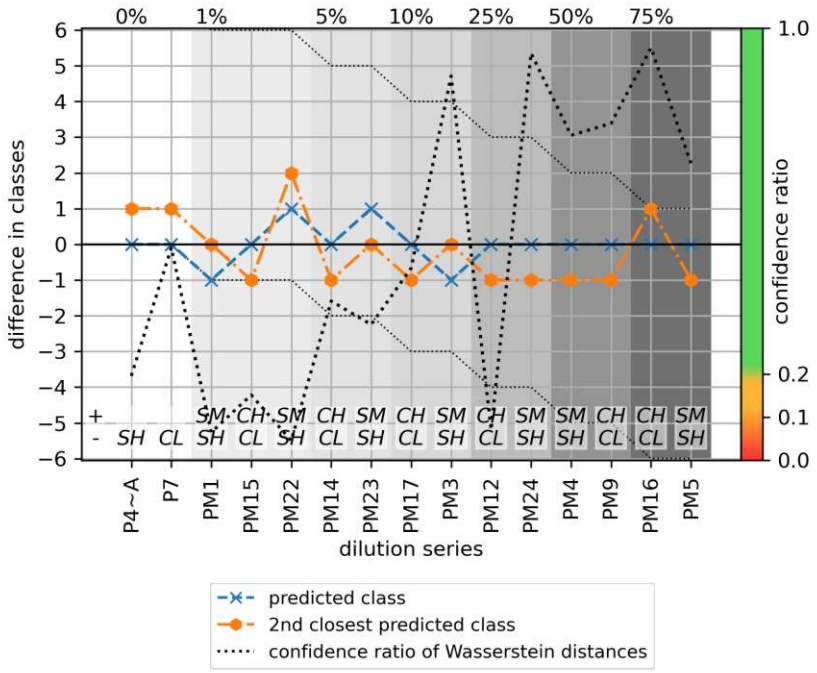


Figure 9.8: Confidence ratio γ of the Wasserstein distance model for all dilution series in $\mathcal{D}$. The figure displays differences of the predicted class (dashed, blue line) and of the second closest ALT+ class (dash-dotted, orange) as well as the confidence ratio γ (dotted, black). The left axis refers to the number of class differences according to the predicted class and second closest class. The right axis displays the confidence ratio and a colour scheme according to the criteria of Section 9.6. The shaded background corresponds to the true ALT+ ratio, which is also indicated on top of the figure. At the bottom of the figure, we find the cell lines of each dilution series in two separate rows with the following encoding: the top row (+) records the ALT+ cell lines SK-N-MM (SM) or CHLA90 (CH). In the bottom row (-), we find the ALT− cell lines SK-N-SH (SH) or CLB-MA (CL).

For the FGA models, we found that they mostly agree on the important features to predict the ALT status. They comprise cluster shade and cluster prominence, kurtosis, skewness, the size of the nuclei' largest spot and the spot count. By investigating the feature maps at activation layers, we ascertained that MyNet extracts features that appear similar to these most relevant FGA features.

The results for predicting the ALT+ ratios on $\mathcal{D}$ show that the nuclear FGA models perform considerably worse than the IBA models. Still, the Wasserstein distance model outperforms all other models by providing most accurate predictions that are at most one ALT+ class away from the actual ALT+ ratios.

We also analysed how confident the models are when predicting the ALT+ status on $\mathcal{D}$. We generally note that the IBA models ResNet and particularly MyNet are much

more confident in their predictions than FGA models. For the logistic regression model, we learned that its rate of sure ALT+ assignments corresponds quite well to the actual ALT+ ratio. The Wasserstein distance model is mostly confident when assigning the ALT+ rate. For half of the cases when the Wasserstein distance model wrongly predicted the ALT+ rate, the Wasserstein distance model was in fact unconfident. The results underline that considering the models' prediction confidence is valuable for analysing how trustworthy the model outputs are.

CHAPTER 10

Summary and Conclusions

This thesis poses two main research questions RQ1 and RQ2 to asses how well our chosen FGA and IBA models allow for classifying the ALT status of neuroblastoma cells. Our research questions cover two main aspects to assess the quality of ALT classification: prediction quality and confidence (RQ1), as well as main drivers of model decisions (RQ2).

For research question RQ1 of Section 6.8, our results show that the Wasserstein distance model provides most accurate predictions on $\mathcal{T}_{test}$ and $\mathcal{D}$. The confidence ratio γ that we introduce in Section 9.6 is a reliable measure for determining the model's confidence when predicting the ALT ratio of a given dilution series. The remaining nuclear FGA and IBA models predict the ALT status on nucleus level and provide overall worse results on $\mathcal{T}_{test}$ and $\mathcal{D}$ than the Wasserstein distance model. Among these models, ResNet, MyNet and the logistic regression model predict the ALT status best on $\mathcal{D}$. When considering only sure ALT+ assignments according to Section 9.6, the predictions of ResNet and the logistic regression model become more accurate. We believe that the Wasserstein distance model outperforms the other approaches, as it is based on feature distributions of multiple nuclei, which appears to stabilise predictions.

For research question RQ2, we have seen that FGA models mostly agree on features that are able to predict the ALT status well. The most relevant features focus on the presence of clusters as well as the skewness and kurtosis of the intensity distribution in the telPNA channel, or consider spot sizes. We have found that MyNet appears to extract image properties that are similar to these FGA features.

In summary, we recommend using the Wasserstein distance model when we are interested in ALT classification on series level based on images of multiple nuclei. If we want to classify the ALT status of an individual cell, we recommend using ResNet. If, in addition, we want to explain and reconstruct how and why the model predicted the ALT status of an individual cell, we recommend using the logistic regression model.

The results of this thesis may in the future impact clinical diagnostics of ALT. In a controlled setting, both the FGA and IBA models have managed to predict the ALT status on cell line and series level with high confidence, and assessing predictions of multiple models at once (such as the Wasserstein distance model, ResNet and the logistic regression model) may even further foster confidence. While in any case experts have to verify the predictions on whether a sample is ALT+ or ALT−and if the predicted ALT+ ratio is sound, models allow for fast processing thousands of cells in a sample. Hence, they may support experts in diagnosing the ALT status for clinical reports.

There are several directions for paths of future research. It appears essential to apply and, if necessary, also train the models on data of new cell types and also on tissue segments to assess how stable the selected features and CNNs are. Similarly, it is worth assessing how well the models predict the ALT status under different recording situations than the standardised setup that we used for our data. Furthermore, for the Wasserstein distance model, we note that $\mathcal{T}_{test}$ and the individual dilution series of $\mathcal{D}$ consist each of 4,000 - 5,000 nuclei. Analysing how this number of nuclei affects the prediction quality is an interesting question for future research. Similarly, if much more than 4,000 nuclei per dilution series and sufficient hardware are available, it appears worth assessing three-dimensional Wasserstein distance models, which use the distribution of three instead of two features discussed in Section 6.6.3.

Glossary

ALT alternative lenghtening of telomeres. 1

CCRI children's cancer research institute. 2

CNN convolutional neural network. 3

CV cross-validation. 15

DNA deoxyribonucleic acid. 7

FGA feature generation approach. 3

FISH fluorescence in-situ hybridisation. 1

GLCM gray level co-occurance matrices. 16, 50

GLCM gray level run length matrices. 16, 50

GLSZM gray level size zone matrices. 16, 50

HEp-2 human elliptical 2. 4

IBA image based approach. 3

SVC, SVM support vector machine. 66

telPNA telomere PNA. 1

WGS whole genome sequencing. 36

XGB extreme gradient boosting. 66

Bibliography

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[ACH+18] Sandra Ackermann, Maria Cartolano, Barbara Hero, Anne Welte, Yvonne Kahlert, Andrea Roderwieser, Christoph Bartenhagen, Esther Walter, Judith Gecht, Laura Kerschke, et al. A mechanistic classification of clinical phenotypes in neuroblastoma. *Science*, 362(6419):1165–1170, 2018.

[AKTO15] Morteza Moradi Amin, Saeed Kermani, Ardeshir Talebi, and Mostafa Ghelich Oghli. Recognition of acute lymphoblastic leukemia cells in microscopic images using k-means clustering and support vector machine classifier. *Journal of medical signals and sensors*, 5(1):49, 2015.

[BEDP+97] Tracy M Bryan, Anna Englezou, Luciano Dalla-Pozza, Melissa A Dunham, and Roger R Reddel. Evidence for an alternative mechanism for maintaining telomere length in human tumors and tumor-derived cell lines. *Nature medicine*, 3(11):1271–1274, 1997.

[BEG+95] Tracy M Bryan, Anna Englezou, Jyothi Gupta, Silvia Bacchetti, and Roger R Reddel. Telomere elongation in immortal human cells without detectable telomerase activity. *The EMBO journal*, 14(17):4240–8, 1995.

[Bis06] Christopher M Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Bun08] Fred Bunz. *Principles of cancer genetics*, volume 1. Springer, 2008.

[CD14] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.

[CG16] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[CGGS13] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International conference on medical image computing and computer-assisted intervention*, pages 411–418. Springer, 2013.

[Cut13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[DMC15] Dev Kumar Das, Asok K. Maiti, and Chandan Chakraborty. Automated system for characterization and classification of malaria-infected stages using light microscopic images of thin blood smears. *Journal of microscopy*, 257(3):238–252, 2015.

[DRCF+03] Fabrizio D'Adda Di Fagagna, Philip M Reaper, Lorena Clay-Farrace, Heike Fiegler, Philippa Carr, Thomas Von Zglinicki, Gabriele Saretzki, Nigel P Carter, and Stephen P Jackson. A DNA damage checkpoint response in telomere-initiated senescence. *Nature*, 426(6963):194–198, 2003.

[EVC+19] Théo Estienne, Maria Vakalopoulou, Stergios Christodoulidis, Enzo Battistela, Marvin Lerousseau, Alexandre Carre, Guillaume Klausner, Roger Sun, Charlotte Robert, Stavroula Mougiakakou, et al. U-resnet: Ultimate coupling of registration and segmentation with deep nets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 310–319. Springer, 2019.

[FCG+21] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie TH Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[FMS+18] Carlos A Ferreira, Tânia Melo, Patrick Sousa, Maria Inês Meyer, Elham Shakibapour, Pedro Costa, and Aurélio Campilho. Classification of breast cancer histology images through transfer learning using a pre-trained inception resnet v2. In *International conference image analysis and recognition*, pages 763–770. Springer, 2018.

[FRM+22] Lukas Frank, Anne Rademacher, Norbert Mücke, Stephan M Tirier, Emma Koeleman, Caroline Knotz, Sabrina Schumacher, Sabine A Stainczyk, Frank Westermann, Stefan Fröhling, et al. Alt-fish quantifies alternative lengthening of telomeres activity by imaging of single-stranded repeats. *Nucleic Acids Research*, 2022.

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

[GKH16] Robert J Gillies, Paul E Kinahan, and Hedvig Hricak. Radiomics: images are more than pictures, they are data. *Radiology*, 278(2):563, 2016.

[GWZZ16] Zhimin Gao, Lei Wang, Luping Zhou, and Jianjia Zhang. Hep-2 cell image classification with deep convolutional neural networks. *IEEE journal of biomedical and health informatics*, 21(2):416–428, 2016.

[HCH+09] Jeremy D Henson, Ying Cao, Lily I Huschtscha, Andy C Chang, Amy YM Au, Hilda A Pickett, and Roger R Reddel. Dna c-circles are specific and quantifiable markers of alternative-lengthening-of-telomeres activity. *Nature biotechnology*, 27(12):1181–1185, 2009.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[HGH+96] Jeff C Hoehner, Carolina Gestblom, Fredrik Hedborg, Bengt Sandstedt, Leif Olsen, and S Påhlman. A developmental model of neuroblastoma: differentiating stroma-poor tumors' progress along an extra-adrenal chromaffin lineage. *Laboratory investigation; a journal of technical methods and pathology*, 75(5):659–675, 1996.

[HM04] Kai Huang and Robert F Murphy. From quantitative microscopy to automated image understanding. *Journal of biomedical optics*, 9(5):893–912, 2004.

[HR10] Jeremy D Henson and Roger R Reddel. Assaying and investigating alternative lengthening of telomeres activity in human cells and cancers. *FEBS letters*, 584(17):3800–3811, 2010.

[HSH+11] Christopher M Heaphy, Andrea P Subhawong, Seung-Mo Hong, Michael G Goggins, Elizabeth A Montgomery, Edward Gabrielson, George J Netto, Jonathan I Epstein, Tamara L Lotan, William H Westra, et al. Prevalence of the alternative lengthening of telomeres telomere maintenance mechanism in human cancer subtypes. *The American journal of pathology*, 179(4):1608–1615, 2011.

[HSK+12] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[HSNH+21] Sabine A Hartlieb, Lina Sieverling, Michal Nadler-Holly, Matthias Ziehm, Umut H Toprak, Carl Herrmann, Naveed Ishaque, Konstantin Okonechnikov, Moritz Gartlgruber, Young-Gyu Park, et al. Alternative lengthening of telomeres in childhood neuroblastoma from genome to proteome. *Nature communications*, 12(1):1–18, 2021.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[JWHT13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[Kan42] Leonid V Kantorovich. On the translocation of masses, cr dokl. *Acad. Sci. URSS*, 37:191–201, 1942.

[KFB+21] Florian Kromp, Lukas Fischer, Eva Bozsaky, Inge M Ambros, Wolfgang Dörr, Klaus Beiske, Peter F Ambros, Allan Hanbury, and Sabine Taschner-Mandl. Evaluation of deep learning architectures for complex immunofluorescence nuclear image segmentation. *IEEE Transactions on Medical Imaging*, 40(7):1934–1949, 2021.

[KGB+12] Virendra Kumar, Yuhua Gu, Satrajit Basu, Anders Berglund, Steven A Eschrich, Matthew B Schabath, Kenneth Forster, Hugo JWL Aerts, Andre Dekker, David Fenstermacher, et al. Radiomics: the process and the challenges. *Magnetic resonance imaging*, 30(9):1234–1248, 2012.

[Kle13] Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[KPMR18] Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[KVRKB+15] Verena Kaynig, Amelio Vazquez-Reina, Seymour Knowles-Barley, Mike Roberts, Thouis R Jones, Narayanan Kasthuri, Eric Miller, Jeff Lichtman, and Hanspeter Pfister. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. *Medical image analysis*, 22(1):77–88, 2015.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LST+16] Geert Litjens, Clara I Sánchez, Nadya Timofeeva, Meyke Hermsen, Iris Nagtegaal, Iringo Kovacs, Christina Hulsbergen-Van De Kaa, Peter Bult, Bram Van Ginneken, and Jeroen Van Der Laak. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Scientific reports*, 6(1):1–11, 2016.

[LTH+18] Michael Lee, Erdahl T Teber, Oliver Holmes, Katia Nones, Ann-Marie Patch, Rebecca A Dagg, Loretta M S Lau, Joyce H Lee, Christine E Napier, Jonathan W Arthur, et al. Telomere sequence content can be used to determine alt activity in tumours. *Nucleic acids research*, 46(10):4903–4918, 2018.

[Mar10] John M Maris. Recent advances in neuroblastoma. *New England Journal of Medicine*, 362(23):2202–2211, 2010.

[Mon81] Gaspard Monge. Mémoire sur la théorie des déblais et remblais, mémoires acad. *Royale Sci. Paris*, 3:1781, 1781.

[MSR+20] Li Ma, Renjun Shuai, Xuming Ran, Wenjia Liu, and Chao Ye. Combining dc-gan with resnet for blood cell image classification. *Medical & biological engineering & computing*, 58(6):1251–1264, 2020.

[MSS+13] Firas Mualla, Simon Schöll, Björn Sommerfeldt, Andreas Maier, and Joachim Hornegger. Automatic cell detection in bright-field microscope images using sift, random forests, and hierarchical clustering. *IEEE transactions on medical imaging*, 32(12):2274–2286, 2013.

[MWT+21] Danny MacKenzie, Andrea K Watters, Julie T To, Melody W Young, Jonathan Muratori, Marni H Wilkoff, Rita G Abraham, Maria M Plummer, and Dong Zhang. Alt positivity in human cancers: Prevalence and clinical insights. *Cancers*, 13(10):2384, 2021.

[oAp] ICAP Internationcal Consensus on ANA patterns. Nomenclature and classification trees. `https://anapatterns.org/trees-full.php`. Accessed: 2021-09-12.

[OD12] Sarah Osterwald and Katharina Deeg. The beginnig of an ALTernate ending? *Landes Bioscience*, 3(3):263–275, 2012.

[Pau21] Christiane Paukner. Etablierung der quantitativen fluoreszenz-in-situ-hybridisierung als diagnostisches verfahren. Fachhochschule Wiener Neustadt, Johannes-Gutenberg-Straße 3, A-2700 Wiener Neustadt, 5 2021. Bachelor's Thesis.

[PBH+03] Sven Perner, Silke Brüderlein, Cornelia Hasel, Irena Waibel, Alexandra Holdenried, Neslisah Ciloglu, Heiko Chopurian, Kirsten Vang Nielsen, Andreas Plesch, Josef Högel, and Möller. Quantifying Telomere Lengths of Human Individual. *Am J Pathol. 2003 Nov; 163(5): 1751–1756. doi: 10.1016/S0002-9440(10)63534-1*, 163(5):1751–1756, 2003.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[Pla99] John C Platt. Probabilistic outputs for svms and comparisons to regularized likehood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[PPG+15] Annalisa Pezzolo, Angela Pistorio, Claudio Gambini, Riccardo Haupt, Manuela Ferraro, Giovanni Erminio, Bruno De Bernardi, Alberto Garaventa, and Vito Pistoia. Intratumoral diversity of telomere length in individual neuroblastoma tumors. *Oncotarget*, 6(10):7493, 2015.

[PVG+11] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Mattieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RNM17] Larissa Ferreira Rodrigues, Murilo Coelho Naldi, and Joao Fernando Mari. Exploiting convolutional neural networks and preprocessing techniques for hep-2 cell classification in immunofluorescence images. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 170–177. IEEE, 2017.

[RNM20] Larissa Ferreira Rodrigues, Murilo Coelho Naldi, and Joao Fernando Mari. Comparing convolutional neural networks and preprocessing techniques

for hep-2 cell classification in immunofluorescence images. *Computers in biology and medicine*, 116:103542, 2020.

[RSIK18] Alexander Rakhlin, Alexey Shvets, Vladimir Iglovikov, and Alexandr A Kalinin. Deep convolutional neural networks for breast cancer histology image analysis. In *international conference image analysis and recognition*, pages 737–744. Springer, 2018.

[RSW+19] Andrea Roderwieser, Frederik Sand, Esther Walter, Janina Fischer, Judith Gecht, Christoph Bartenhagen, Sandra Ackermann, Felix Otte, Anne Welte, Yvonne Kahlert, et al. Telomerase is a prognostic marker of poor outcome and a therapeutic target in neuroblastoma. *JCO Precision Oncology*, 3:1–20, 2019.

[RWSZ20] Saimunur Rahman, Lei Wang, Changming Sun, and Luping Zhou. Deep learning based hep-2 image classification: A comprehensive review. *Medical Image Analysis*, page 101764, 2020.

[SRT+16] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging*, 35(5):1196–1206, 2016.

[Sto74] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

[SWMP21] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021.

[VCX18] Yeeleng S Vang, Zhen Chen, and Xiaohui Xie. Deep learning framework for multi-class breast cancer histology image classification. In *International conference image analysis and recognition*, pages 914–922. Springer, 2018.

[vGFP+17] Joost JM van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo JWL Aerts. Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer Research*, 77(21):e104–e107, 10 2017.

[VGO+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert

Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[Vil09] Cédric Villani. *The Wasserstein distances*, pages 93–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[WSC+19] Yu Wang, Fuqian Shi, Luying Cao, Nilanjan Dey, Qun Wu, Amira S Ashour, Robert Simon Sherratt, Venkatesan Rajinikanth, and Lijun Wu. Morphological segmentation analysis and texture-based support vector machines classification on mice liver fibrosis microscopic images. *Current Bioinformatics*, 14(4):282–294, 2019.

[XXS+17] Fuyong Xing, Yuanpu Xie, Hai Su, Fujun Liu, and Lin Yang. Deep learning in microscopy image analysis: A survey. *IEEE transactions on neural networks and learning systems*, 29(10):4550–4568, 2017.

[YLP+17] Stephen SF Yip, Ying Liu, Chintan Parmar, Qian Li, Shichang Liu, Fangyuan Qu, Zhaoxiang Ye, Robert J Gillies, and Hugo JWL Aerts. Associations between radiologist-defined semantic and automatically computed radiomic features in non-small cell lung cancer. *Scientific reports*, 7(1):1–11, 2017.

[ZLH+19] Ye Zhang, Mingchao Li, Shuai Han, Qiubing Ren, and Jonathan Shi. Intelligent identification for rock-mineral microscopic images using ensemble machine learning algorithms. *Sensors*, 19(18):3914, 2019.