

U-Net basierte Klassifikation von Durchflusszytometrie-Daten

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Data Science

eingereicht von

Tímea Tóth, Bsc

Matrikelnummer 01502072

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Senior Lecturer Dipl.-Ing. Dr.techn. Michael Reiter

Wien, 27. September 2023



Tímea Tóth

Michael Reiter

U-Net based Classification of Flow Cytometry Data

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Data Science

by

Tímea Tóth, Bsc


Registration Number 01502072

to the Faculty of Informatics

at the TU Wien

Advisor: Senior Lecturer Dipl.-Ing. Dr.techn. Michael Reiter

Vienna, 27th September, 2023


Tímea Tóth


Michael Reiter

Erklärung zur Verfassung der Arbeit

Tímea Tóth, Bsc

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, 27. September 2023



Tímea Tóth

Acknowledgements

Firstly, I would like to thank my supervisor, Michael Reiter, as well as Florian Kowarsch from the Computer Vision Lab for the support and guidance throughout this thesis, for the continuous feedback, and for awaking my interest in computer vision.

Special thanks to my family and friends for their continuous support during my studies and for their tolerance in difficult times. I would like to thank Marina and András for making our studies so enjoyable, studying with you was so much fun and even helped me through the hardest exams. Special thanks to Bella, you have constantly supported me both professionally and personally.

Last but not least, I would like to thank Xaver, without your advice, patience, and encouragement, I would not have been able to finish my studies.

Danksagung

Zunächst möchte ich mich bei meinem Betreuer Michael Reiter und bei Florian Kowarsch vom Computer Vision Lab für die Unterstützung und Anleitung während dieser Arbeit, für das kontinuierliche Feedback und dafür, dass sie mein Interesse an der Computer Vision geweckt haben, bedanken.

Besonderer Dank gilt meiner Familie und meinen Freunden für die kontinuierliche Unterstützung während des Studiums und dass ihr mich auch in schwierigen Zeiten ausgehalten habt. Ich möchte mich bei Marina und András dafür bedanken, dass ihr unser Studium so unterhaltsam gemacht habt. Das Lernen mit euch hat mir so viel Spaß gemacht und mir auch durch die härtesten Prüfungen geholfen. Besonderen Dank an Bella, du hast mich sowohl beruflich als auch persönlich immer unterstützt.

Zu guter Letzt möchte ich mich bei Xaver bedanken, ohne deine Ratschläge, deine Geduld und deinen Zuspruch hätte ich mein Studium nicht beenden können.

Kurzfassung

Die akute lymphoblastische Leukämie (ALL) ist die häufigste Krebserkrankung bei Kindern. Bei bösartigen Erkrankungen können sich abnorme Zellen unkontrolliert teilen und auf umliegende Zellen übergreifen. Die Chemotherapie ist die häufigste Behandlung der ALL. Da jeder Patient anders auf die Therapie anspricht, muss sie kontrolliert und individuell angepasst werden. Die minimale Restkrankheit (Minimal Residual Disease, MRD) gibt Aufschluss über das Ansprechen des Patienten auf die Behandlung. Anhand dieses Wertes kann die Intensität oder Dauer der Behandlung angepasst werden. Die Durchflusszytometrie ist eine laserbasierte Methode, mit der die MRD nachgewiesen werden kann.

Es gibt mehrere Methoden zur automatischen Erkennung von Krebszellpopulationen auf der Grundlage von Durchflusszytometriedaten mit beeindruckenden Ergebnissen, die jedoch einen entscheidenden Nachteil haben: die mangelnde Interpretierbarkeit der Verfahren. In dieser Arbeit wenden wir eine Bildsegmentierungsmethode, das U-Net Modell, an, um dieses Problem auf visuelle Weise zu lösen, so dass alle Schritte des Gating-Verfahrens leicht nachvollzogen werden können.

Die Ergebnisse entsprechen dem Resultaten der State-of-the-art mit dem zusätzlichen Vorteil, dass die visuelle Darstellung des Gating-Verfahrens leicht nachvollziehbar ist.

Abstract

Acute lymphoblastic leukemia (ALL) is the most common type of malignant disease among children. In the case of malignant diseases, abnormal cells can divide uncontrollably and spread to surrounding cells. Chemotherapy is the most common treatment for ALL. Since every patient responds differently to the therapy, it needs to be controlled and individualised. Minimal Residual Disease (MRD) indicates the patient's response to the treatment. Based on this value the intensity or length of treatment can be modified. Flow cytometry is a laser-based method which can detect MRD.

There are several methods for the automatic detection of cancer cell populations based on flow cytometry data with astonishing results but have only one striking disadvantage: the lack of interpretability of the processes. In this thesis, we apply an image segmentation method, the U-Net model, to solve this problem in a visual manner so that all steps of the gating procedure can be easily comprehended.

The results meet the performance of the state-of-the-art methods with the additional advantage of easy tractability through the visual representation of the gating procedure.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
1.1 Research Questions	3
1.2 Contribution and Overview of the Thesis	4
2 Clinical Background	7
2.1 Flow Cytometry	7
2.2 Data	10
2.3 Gating procedure	10
3 Related Work	13
3.1 State-of-the-art Methods	13
3.2 Evaluation of the state-of-the-art Methods	15
4 Method	17
4.1 Image segmentation	17
4.2 Biomedical image segmentation	18
4.3 Fundamentals of Neural Networks	18
4.4 Convolutional Neural Networks	20
4.5 Training a neural network	24
4.6 Regularisation of a Neural Network	25
4.7 Optimisation of a Neural Network	27
4.8 U-Net Architecture	28
5 Application of the U-Net for cancer cell detection in flow cytometry	
data	33
5.1 Workflow for implementing the automation of the gating procedure . .	33
5.2 Training of the network	38
5.3 Experiments	43
	xv

5.4 Evaluation of the model performance	44
6 Results	47
6.1 Descriptive analysis	47
6.2 Overall evaluation	54
7 Conclusion	57
7.1 Conclusion of research questions	57
7.2 Limitations	59
7.3 Contributions and Future Work	60
List of Figures	61
List of Tables	63
Acronyms	65
Bibliography	67

CHAPTER 1

Introduction

The second most frequent cause of death worldwide is cancer [HD17]. Cancer research is key to understanding the mechanisms of cancer and developing new treatments, tests, and prevention measures for cancer [CRU]. Many different types of cancer exist. 3% of cancer diagnoses are leukemia cases, however, this rate is significantly higher (33%) for children under 15 years of age [PRL08]. Childhood leukemia cases are curable in 80% of the cases [PRL08] therefore early detection and treatment are essential.

Leukemia is a cancer of the blood cells which starts in the bone marrow. In leukemia patients, there is an uncontrolled growth of abnormal, immature white blood cells, called blasts, which flood the bone marrow and prevent the production of other vital cells such as red blood cells and platelets, which are essential for survival [Bai17]. Leukemia can be divided into different types based on the type of white blood cells are affected and the speed of progress: chronic lymphocytic (CLL), chronic myeloid (CML), acute myeloid (AML), acute lymphocytic (ALL), monocytic (ML) as well as other types [HSLB12]. In this thesis, we will focus on acute lymphocytic leukemia (ALL), which occurs when too many stem cells become abnormal lymphocytes [Bai17].

Every patient responds differently to the therapy; therefore, it is crucial to control and individualise it. Based on the Minimal Residual Disease (MRD) value of the patient, treatment length and intensity can be determined [Cam09]. MRD is defined as the number of remaining cancer cells measured in bone marrow samples after chemotherapy. Evaluating MRD has become a proven diagnostic tool for treatment monitoring [BVV⁺09].

There are several different methods for MRD monitoring, such as cell-culture systems, fluorescent in situ hybridisation, southern blotting, immunophenotyping, and polymerase chain reaction (PCR) techniques. Unfortunately, most of these methods have either low sensitivity, specificity, or applicability. PCR-based methods and multiparameter flow-cytometric immunophenotyping reach a high sensitivity and are generally applicable [SOvdV⁺01]. In this thesis, the focus lies on flow-cytometric immunophenotyping.

Flow cytometry is a technique during which light is beamed onto a sample flowing in a liquid stream and the scattered light signals are then measured. This technique is frequently utilized in the analysis of blood and bone marrow samples. Flow cytometry is applied in many fields such as cancer biology, monitoring of infectious diseases, immunology, and virology, and is the basis of immunophenotyping, a technique to mark protein expressions of a cell population by applying fluorescent compounds to categorize the cell [HM22]. Flow cytometry will be described in Section 2.1 in more detail.

The data collected by flow cytometry can then be used to identify cancer cell populations. By a process called *gating*, medical experts draw multiple polygons between different cell populations on 2D images to hierarchically sub-select and detect cancer cell populations. This process is repeated until the filtered data consists only of cancer cells. These annotations are used as a basis for training automatic prediction models [RRK⁺16].

This task is highly dependent on the medical expert since it requires an advanced understanding of the properties of the flow cytometry cells and samples. Additionally, the gating procedure is very time-consuming. In light of these challenges, there is an emerging need for the automation of the procedure. Through automation, the process can be accelerated, the resources of medical experts can be optimized, and the possibility of human error can be reduced. Therefore, many methods have been developed with one goal in mind: to detect cancer cell populations in an automated way.

In this thesis, we will use two-dimensional projections of the data space as they are employed by medical experts in the gating procedure. The reasoning behind this approach is to recreate the gating procedure and hence the way the ground truth is generated since the gating procedure relies only on two-dimensional projections as well. We will use the U-Net architecture for segmentation in order to detect cancer cell populations in 2D images. A description of the U-Net architecture and an explanation of why the U-Net architecture is suitable for this problem will be outlined in Section 4.8.

This thesis aims to automatically identify cancer cells using flow cytometry data of acute lymphoblastic leukemia (ALL) patients based on 2D plots created as part of the gating procedure. Based on the labelled data from medical experts, a model can be trained by hierarchically dividing the cancer cell populations. The gating procedure performed by medical experts uses a fixed sequence of 2D images to detect blast cells. This process will be replicated using our Neural Network model.

There are several automated methods that have been proposed to detect cancer cells in flow cytometry data. Most of them work with multidimensional data and assign a class label to each event of the sample, but do not provide additional information as to why a particular cell was classified as a cancer cell [RRK⁺16] [WRW⁺22] [ANHB11] which would be essential information in order to gain trust and transparency.

The proposed approach offers the advantage that by using two-dimensional projections, medical experts are enabled to directly follow the process of cancer cell detection in the gating procedure. Moreover, the use of two-dimensional projections, as opposed to multidimensional alternatives, leads to a significant increase in process efficiency by

reducing computational complexity. Based on this visual approach using the U-Net architecture, cancer cells can be traced in a supervised manner. By making this method interpretable through the visualization of each step of the process, additional information regarding the automated gating procedure can be obtained.

We aim to achieve this goal by developing a Deep Convolutional Neural Network. Deep Convolutional Neural Networks in particular have proven to be highly effective on spatial data like images or audio. Experts manually identify cancer cell populations in a flow cytometry sample using a combination of 2D plots. Therefore, it would seem natural to process flow cytometry samples as 2D images for automated analysis as well.

One property of CNNs is translation invariance, which is not suitable in our case because the location of each pixel is crucial. Each pixel may contain cells that are either selected for the next gating step or not. Hence, we need an extension of the ordinary CNN which considers the location of each pixel, such as the U-Net architecture. The output of this network architecture is a segmentation mask of the input image. These masks contain information on whether a pixel contains the cells of interest. This type of problem can be considered a classification task, in which every pixel of the image will be assigned to a class [RFB15]. The U-Net is capable of automatically identifying cancer cell populations in the given flow cytometry data with the additional advantage of a visual representation of the process that is easy to interpret.

We compare the proposed U-Net method for automated cancer cell detection with the transformer architecture proposed by Wödlinger et al. [WRW⁺22], which serves as the baseline method in our experiments. It works in higher dimensional data spaces whereas the proposed method is restricted to sequentially classifying 2D images.

The main advantage of our method lies in the interpretability of the output of the network. The result of the proposed approach is similar to the gating procedure and is therefore comprehensible for medical experts. Working with two-dimensional projections could offer the additional advantage of reducing the amount of training data required, thanks to the smaller number of dimensions and the reduced model complexity. Since medical data is difficult to obtain, this could be a great benefit for further research.

1.1 Research Questions

The scope of this thesis is the implementation, application, and evaluation of the U-Net architecture for the automated identification of cancer cells based on flow cytometry data. The following research questions will be addressed:

1. What is the optimal U-Net architecture in terms of layers and kernel parameters for automating the gating procedure in flow cytometry data?

What preprocessing steps are essential to prepare the input for the U-Net architecture's gating procedure replication? This question investigates the efficacy of the U-Net architecture in addressing the specified task and identifies the most suitable

architectural parameters. Furthermore, it explores the application of segmentation masks generated by the U-Net for data filtering and cancer cell detection. Additionally, it assesses the impact of employing separate models for each hierarchical level against using a single model across all hierarchy levels, examining whether false predictions are easier to detect.

2. What distinct advantages does the proposed method offer in comparison to state-of-the-art techniques for identifying cancer cell populations?

The main objective of this thesis is to evaluate whether we need multidimensional flow cytometry data to identify cancer cells or if two-dimensional projections can also provide accurate results. This question will highlight both the strengths and limitations of the proposed architecture. Performance evaluation will compare the proposed U-Net architecture against established methods, using metrics such as F1 score, precision, and recall. Furthermore, we will ascertain whether the proposed method outperforms alternatives when dealing with varying proportions of cancer cells (MRD) and outline situations where this is the case.

1.2 Contribution and Overview of the Thesis

The key contribution of this thesis is the adaptation of the U-Net architecture to the sequential 2D images generated from flow cytometry data, the implementation of the hierarchical sub-selection of the sub-populations and the evaluation and comparison of the results with the current state of the literature in automated cancer cell detection using flow cytometry data.

While several methods have already been proposed to automatically detect cancer cells, the main difference between the existing methods and the proposed method is the reduced complexity due to the use of 2D image data rather than multidimensional datasets and the explainability of the results. The proposed method provides additional insights into related research on cancer cell detection. Additionally, we would like to investigate, whether a Convolutional Neural Network-based approach is capable of identifying cancer cell populations meeting state-of-the-art performance.

This thesis has the following structure: In Chapter 2, the clinical background will be elaborated upon. It begins with an introduction to flow cytometry, its history and applications, followed by the gating procedure. In the last part of the chapter, the state-of-the-art methods for automated cancer cell detection using flow cytometry data will be introduced and evaluated.

Chapter 3 introduces the related work in the automated identification of cancer cells using flow cytometry data for the detection and monitoring of **MRD** of Acute Lymphoblastic Leukemia patients.

Chapter 4 summarises image segmentation methods, the fundamentals of Convolutional Neural Networks, and the U-Net architecture.

In Chapter 5, the implementation of the automated gating procedure using the U-Net architecture will be presented. In this part of the thesis, the data preparation, generation

of input images, as well as the creation and usage of segmentation masks will be described. Additionally, the network architecture, the model training itself, the implementation and challenges of the hierarchical approach, as well as the experiments conducted will be further elaborated.

In Chapter 6, the evaluation methods and metrics will be introduced, followed by additional insights gathered from the experiments.

In Chapter 7 the proposed research questions will be answered and the limitations will be addressed.

CHAPTER 2

Clinical Background

Acute Lymphoblastic Leukemia (ALL) is the most common type of malignant disease among children, hence the detection of ALL became a vital and emerging research area. Currently, 80% of childhood ALL can be successfully cured [PRL08] with intensive combination chemotherapy regimens, which in some cases need to be combined with radiotherapy and/or hematopoietic stem cell transplantation [SS09].

The risk of early relapse of leukemia patients significantly correlates with the number of detected residual leukemia cells [CvdWTBS⁺98]. Minimal Residual Disease (MRD) indicates the response of the patient to the treatment even in early stages and additionally enables the recognition of relapse (see Figure 2.1). Further strategies for treatment (modifying intensity or length) are based on the number of remaining leukemia cells that can be detected in the bone marrow or blood cells of the patients. A well-established method for detecting MRD is flow cytometry [Cam09].

There are other methods introduced for the detection of Minimal Residual Disease, for instance PCR-based methods and sequential flow cytometry analysis using the gating method [ch913]. There are various studies (eg. [VdVBVWVD04]) that try to separate patients into different risk groups: MRD low risk, MRD high risk and MRD intermediate risk. Based on these studies within 8 to 15 days of treatment, high MRD risk patients (more than 10% MRD) or low-risk patients (less than 0.01 %) can be detected.

In the following sections, flow cytometry, a technology used to analyse individual cells in order to detect leukemia cells will be introduced (Section 2.1) and the gating method performed manually by medical experts will be explained (Section 2.3).

2.1 Flow Cytometry

Flow Cytometry is a comprehensive technology that biologists have at their disposal to study cell populations with high precision. Its main benefits are the statistical aspects as

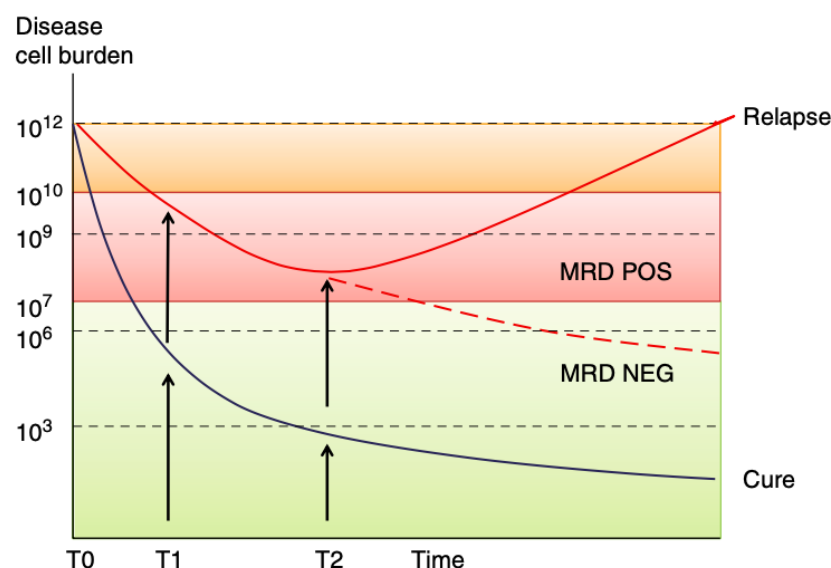


Figure 2.1: Disease burden based on number of MRD [ch913]

well as the opportunity to separate sub-populations [PGLVKB12].

The foundation of flow cytometry is not new; its history leads back to the seventeenth century when microscopes were used to analyse cells and tissues. In the early twentieth century, stains were developed in order to examine various cellular components. For the first time, fluorescence microscopy was applied for the detection of malignant cells in the middle of the twentieth century. Around the 1980's, it was proven that flow cytometry had had a great impact on clinical diagnostics and Flow Cytometers have been already used in numerous hospitals. One of the main reasons for the popularity of this technique is its simplicity because it can be maintained with little human effort and the results can be read without extensive knowledge of flow cytometry technology and without expertise in data analysis [Giv13]. It can be used for numerous applications such as fluorescent protein, cell counting, or MRD detection. Flow cytometers measure the scattered light and fluorescence emissions as the cells pass through the laser beams [PGLVKB12].

A Flow Cytometer is comprised of three key parts:

The Fluidics System (1) consists of the cell-by-cell application of the input sample to the laser.

The primary goal of the Optical System (2) is to extract information from the scattered light. It contains two parts: excitation optics with focusing lenses as well as prisms and collection optics where the scattered light is collected and gathered by special optical detectors. When the cell reaches the interrogation point, the scattered light of the laser beam is collected by several specific detectors, thereby gaining information about the physical properties of the cell such as size and granularity. In this process, the Forward Scattered Light (FSC) and Side Scattered Light (SSC) are collected, which

allows differentiation between heterogeneous populations (See [2.2](#)). In order to gather

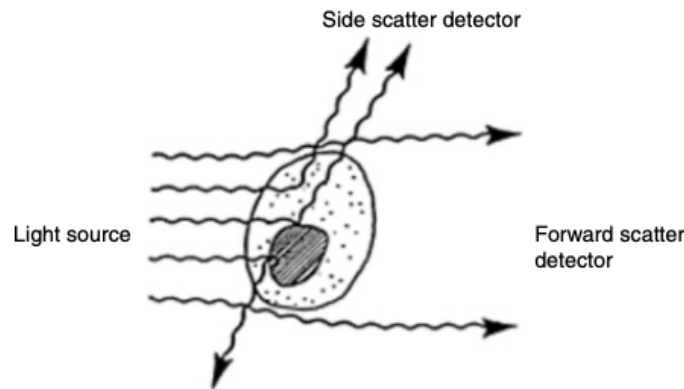


Figure 2.2: Forward Scatter and Side Scatter Light [\[ch213\]](#)

information on the biochemical properties of the cell, dyes of fluorochromes are used and the brightness of the fluorescence intensity is recorded.

The Computer/Electronic System (3) converts the light into numerical data. This data is stored and displayed in numerous ways, for example with histograms or scatter plots [\[ch213\]](#).

With its rapidly increasing applications, flow cytometry has been used for decades to support the diagnosis of haematological diseases. This technology is widely used for the detection of leukemia, characterisation and prognostics for children, and even during therapy. The previous approach for identifying leukemia was based on the leukemia's morphology. The problem with this method is that its success depends heavily on the haematologist as well as the fact that it is very difficult to distinguish between leukemic cells and normal lympho-haematopoietic progenitor cells, which in turn has a major impact on the treatment required. Given the above limitations of morphology, immunophenotyping of cancer cells is gaining importance and is frequently used in research nowadays. This technology examines antigens on the cell surface in order to identify the cell type and the stage of differentiation. Leukemia detection includes the detection of white blood cells in blood or bone marrow. With a flow cytometer, the labelled cells can be analysed in a short time and it is now possible to detect even a small number of cells, which would not be feasible using solely morphology [\[Wan14\]](#).

The use of flow cytometry has been shown to be extraordinarily important for the detection of Minimal Residual Disease among post-therapy patients. The major idea for [MRD](#) detection by flow cytometry is based on the identification of antigens which are expressed differently between leukemic populations and their normal counterparts in the bone marrow [\[Woo13\]](#).

2.2 Data

The light signals generated by the Flow Cytometer are read by detectors. After being converted into electronic signals, they can be analysed by computers and saved in a standard format for flow cytometry (FCS file format) [Giv01].

In this thesis, we use bone marrow samples for detecting residual leukemia cells analysed by Flow Cytometers. These samples were collected from leukemia (B-ALL) patients 15 days after induction therapy in three different countries: Austria, Germany, and Argentina [JS12].

- Vienna: 519 samples were collected in the St. Anna Children's Cancer Research Institute between 2009 and 2020.

These samples were separated into two datasets:

1. `vie14` contains samples gathered between 2009 and 2014 (200 samples)
 2. `vie20` holds data from the years 2015-2020 (319 samples)
- Berlin: 79 samples were gathered at Charité Berlin in 2016.
 - Buenos Aires: 65 samples were obtained in the years 2016 and 2017 at the Garrahan Hospital in Buenos Aires.

Each sample within these datasets contains approximately 300.000 events. The samples were manually labelled by at least two medical experts (so as to provide a reliable ground truth), using the gating procedure [RDS⁺19]. Three out of the four datasets are accessible at the [FlowRepository website](#). The website does not provide access to the samples that were collected in Vienna between 2016 and 2020.

The resulting data was further filtered by the selection of certain cell populations based on their biological features. Properties such as size (light scattering at the forward angle) and internal complexity (right-angle scattering) can distinguish between specific cell populations [BW00]. Additionally, the cluster of differentiation (CD) values, which summarize the surface characteristics of the different cells, are given. For example, CD19 and CD20 are B-cell markers [Giv01].

2.3 Gating procedure

The main goal of the gating procedure is to separate single populations in a heterogeneous sample. It allows the analysis of sub-populations to hierarchically trace cancer cells [ch213].

Generally, the gating procedure is performed manually by medical experts based on a hierarchical sequence of 2D images. This procedure has several limitations, as it strongly depends on the scientist's knowledge and is extremely time-consuming [LRD⁺18].

Figure 2.3 illustrates the hierarchy of the manual gating procedure. The blue dots are

cells within the polygon drawn by medical experts. These cells will be used for the next 2D representation of the data. The grey cells (outside of the polygon) will not be taken into consideration for the next gating stages.

The first gate is the *Syto+* gate which contains all events corresponding to cells. In the image of the first gate, Front Scatter Area (FSC-A), values of each cell are displayed on the x-axis and Syto 41 values on the y-axis. In this step, the events which are neither cells nor debris are rejected. The second gate is the *Singlets* gate, where cells, which are too large are discarded. The third gate, the *Intact* gate, is where dead cells (small cells) are dismissed. The *CD 19* gate selects the cells which are not B cells, and the final gate is the *Blasts* gate, where the leukemic cells are selected. Three different images are used in order to detect blasts since many cells overlap (cancer cells cannot be clearly separated from non-cancer cells). The intersection of the detected cells on these three images is labelled as cancer cells at the end of the gating procedure [RRK⁺16].

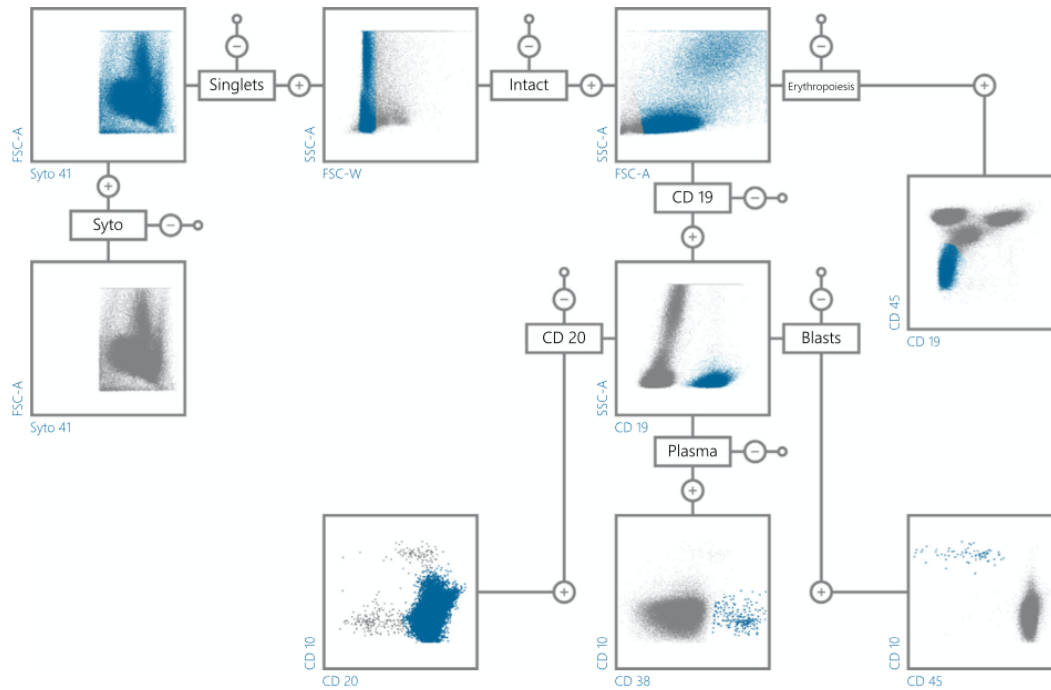
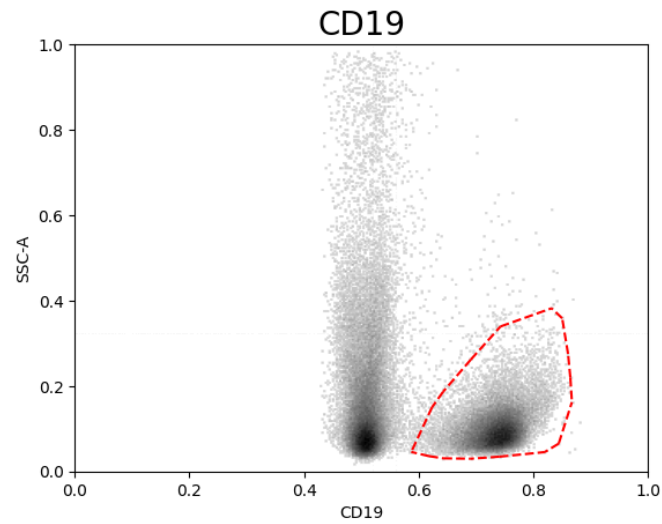


Figure 2.3: Manual gating procedure [RRK⁺16]

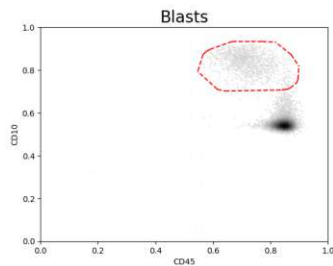
Following the gating procedure, the percent of Minimal Residual Disease is calculated in the samples in the following way [ch913]:

$$\text{MRD}\% = \frac{\text{Number of cancer cells}}{\text{Total number of cells}} \times 100 \quad (2.1)$$

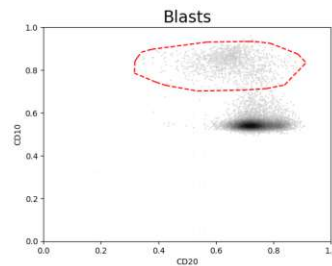
The goal of this thesis is not only to implement the proposed algorithm, but also to compare it with the existing state-of-the-art methods (see Chapter 3). In order to allow



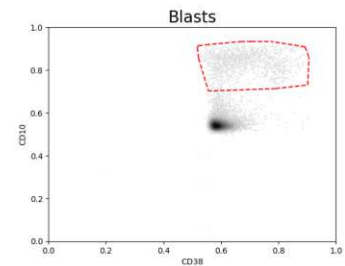
(a) Gate 4: Selection of B cells.



(b) Gate 5.1: Selection of blast cells.



(c) Gate 5.2: Selection of blast cells.



(d) Gate 5.3: Selection of blast cells.

Figure 2.4: Visualisation of the last steps of the manual gating procedure. Image a shows the 4th gate while images b, c, and d are parts of the 5th gate. The images are two-dimensional projections of a sample [JS12] gathered in Vienna. The cells within the red dashed polygons are the cells, labelled by medical experts.

a fair comparison with the performance of these methods, we used all events from the Intact gate (third gate) in the same way as it was described in the studies. This means that only the last two gates (see Figure 2.4) will be taken into consideration for the overall evaluation of the proposed method.

CHAPTER 3

Related Work

To address the limitations of manual gating by medical experts, various approaches have been proposed that allow for automated identification of cell populations in flow cytometry data. This chapter gives an overview of the state-of-the-art methods relevant for this thesis. The focus lies on methods for the automated identification of cancer cells using flow cytometry data of Acute Lymphoblastic Leukemia patients. Common advantages and limitations of the previous research will be pointed out in order to enable a good comparison between the proposed method as well as the state-of-the-art methods.

Recent research papers suggest numerous unsupervised and supervised approaches for this task. In comparison to the manual gating procedure, these methods are applied directly to the multidimensional flow cytometry data [WRW⁺22].

The main challenges of the previously applied methods are the large sample sizes (approximately 300.000 cells per sample and more than 10 features) and the unbalanced design seeing as the number of cancer cells is below 0.01% in some samples [RRK⁺16]. The datasets collected in Berlin and Buenos Aires, as described in Section 2.2, both contain less than 80 samples, which is a relatively small sample size, hence the risk of overfitting is high. Due to this, the application of machine learning methods is not straightforward.

3.1 State-of-the-art Methods

There are various flow cytometry data analysis methods that have shown good results for the automatic identification of cancer cell populations. Manual gating works with the two-dimensional projections of the flow cytometry data, but the proposed automated methods use the entire parameter space. These methods attempt to assign a label (blast or non-blast cell) for each event of the multidimensional dataset without any specific explanation or traceability of the results. These methods will be separated into two groups

based on the type of machine learning approach applied: unsupervised or supervised methods.

3.1.1 Unsupervised Methods

Examples of unsupervised methods, that were used in the state-of-the-art literature, are clustering and density estimation. These methods do not require any labelled data; they function based on the similarity of the different events corresponding to the cells.

K-Means clustering for flow cytometry data (FlowMeans) models the data as a single population which contains various clusters with mode detection using kernel density estimation. To determine the number of clusters, a change point detection algorithm is used [ANHB11]. FlowClust [LBG08] and FlowMerge [FBBG09] are both model-based clustering algorithms which use the Box-Cox transformation combined with the Expectation Maximisation (EM) algorithm with the distinction that FlowMerge additionally applies a cluster merging algorithm. These methods face major challenges when applied to new datasets, since cluster size, shape, and position can vary largely across various laboratories [RRK⁺16].

3.1.2 Supervised Methods

Supervised methods require data labelled by medical experts which serves as ground truth for training the algorithms. There are many different approaches such as Support Vector Machines, Gaussian Mixture Models, or Bayesian approaches.

The main goal of Support Vector Machines is to find a hyperplane that separates cancer cells from non-cancer cells [RRK⁺16]. Gaussian Mixture Models (GMM) can be used in combination with the Expectation Maximisation algorithm [RRK⁺16] where each sample is represented by a GMM. The parameters of this GMM are determined by a linear combination of multiple reference GMM based on the training samples. Furthermore, Reiter et al. [RDS⁺19] proposed another GMM-based approach where the combination of a Gaussian Mixture Model with a parametric density model is used for predicting cancer cell populations. The aim is to find the weights of a linear combination of several GMMs to represent new samples by interpolation of the stored samples. A hierarchical Bayesian model [JWF16] was developed to classify cancer cells using latent modelling. The main advantage of this model is that expert knowledge can be added through priors.

A transformer architecture (a supervised classification technique that directly identifies blast cells of a sample) has been proposed [WRW⁺22]. While it is capable of capturing global information, this entails increased model complexity in terms of memory and time. A transformer is an Encoder-decoder based architecture which uses a stacked self-attention layer and pointwise fully-connected layers [VSP⁺17]. Transformer models are very successful and are commonly used in computer vision tasks, such as object detection or segmentation, as well as Natural Language Processing (NLP) [HWC⁺22].

3.2 Evaluation of the state-of-the-art Methods

The transformer model by Wödinger et al. [WRW⁺22] outperformed existing methods in terms of median F1-scores using the four datasets introduced in Section 2.2. For all experiments performed (except two edge cases), a median F1-score of ≥ 0.86 was obtained, which constitutes quite a breakthrough as a stable performance across all experiments was reached.

The main advantage of the transformer model is the ability to classify all cells of a sample at once relating each cell to all other cells of the sample, owing to the attention mechanism. For further explanation of the transformer architecture, see [WRW⁺22]. The attention mechanism is one of the major differences compared to other deep learning based approaches which use the input data in a sequential manner [VSP⁺17]. This in turn leads to an improvement in performance in terms of computational complexity and runtime.

The results of the best performing state-of-the-art methods are displayed in Table 3.1. In the columns of the table, the precision (p), recall (r), average F1-score (avg F1), and median F1-scores (med F1) are presented using the transformer model proposed by Wödinger et al. [WRW⁺22] compared to the results using Gaussian Mixture Models by Reiter et al. [RDS⁺19]. The first two columns contain information as to which training and test sets were used for the experiment. The remaining two datasets were used as validation sets for the evaluation.

train	test	p	r	avg F ₁	med F ₁	med F ₁ [RDS ⁺ 19]
vie	vie	0.81	0.83	0.81	0.94	=
bln	bue	0.63	0.84	0.66	0.87	0.68
bln	vie14	0.77	0.83	0.77	0.90	0.35
bln	vie20	0.79	0.77	0.74	0.87	0.48
bue	bln	0.56	0.92	0.62	0.77	0.5
bue	vie14	0.76	0.88	0.79	0.90	0.84
bue	vie20	0.79	0.74	0.72	0.88	0.86
vie14	bln	0.78	0.82	0.75	0.9	0.81
vie14	bue	0.82	0.81	0.78	0.95	0.84
vie14	vie20	0.81	0.74	0.73	0.89	0.86
vie20	bln	0.64	0.87	0.66	0.81	0.25
vie20	bue	0.82	0.69	0.71	0.86	0.81
vie20	vie14	0.82	0.69	0.71	0.86	0.89

Table 3.1: Results of the state-of-the-art literature [WRW⁺22]

The GMM model showed very reliable results with median F1-score > 0.5 in more than 95% of the samples; nevertheless, the transformer model outperformed the GMM

approach in terms of the median F1 score in all but one case. In the cases, where the Buenos Aires or Berlin dataset (smallest dataset) were used for training, the results are worse compared to when the Vienna dataset was used.

The transformer model achieved higher performance for samples with a higher MRD fraction, for samples with a very low MRD fraction, the predictions were overestimated rather than underestimated.

3.2.1 Summary

Several different approaches were proposed in the literature for the automated identification of cancer cells in flow cytometry data. There are supervised as well as unsupervised methods; the common characteristic of these models is that they are all applied directly to the multidimensional data.

In conclusion, the proposed methods reached a high performance based on the median F1-score in automated cancer cell detection where the Transformer model proposed by Wödinger et al. [WRW⁺22] outperformed the previously introduced methods. The lack of explanatory ability of the models is a defining characteristic of these approaches. By replicating the manual gating procedure we want to close this gap in the literature.

The next chapter introduces the basics of image segmentation methods and convolutional neural networks and concludes with the U-Net architecture itself.

CHAPTER 4

Method

This chapter introduces the model we used in our approach to automate the gating which is based on the U-Net architecture. The U-Net architecture is a convolutional neural network that was proposed by Ronnenberger et al. [RFB15] for biomedical image segmentation. First, image segmentation (Section 4.1), especially biomedical image segmentation methods (Section 4.2) will be summarized. Next, the foundation of neural networks can be found in Section 4.3, followed by an overview of the convolutional neural networks 4.4. Afterward, the next sections cover the training 4.5, regularisation 4.6 and optimisation 4.7 of neural networks. This chapter will close with a detailed explanation of the U-Net architecture (Section 4.8).

4.1 Image segmentation

Image segmentation is a fundamental field within computer vision. The main objective of this process is to divide images into several segments and then add semantic labels to each pixel of the image.

Image segmentation methods can be split into different categories, e.g. semantic segmentation, instance segmentation, and panoptic segmentation. In case of semantic segmentation, pixels are classified using semantic labels yet this method does not discriminate between distinct instances within the same category. In turn, instance segmentation detects individual objects on an image (e.g. each person). Panoptic segmentation combines image segmentation with semantic segmentation, i.e. in addition to assigning class labels to each pixel, it also identifies which specific instance it belongs to [MBP⁺21].

Image segmentation can be defined as an image processing technique that divides images into meaningful regions or segments. It can additionally be considered as a way of defining boundaries between various semantic units in an image [GDDM19]. It allows a better understanding of the image and identification of objects.

Image segmentation methods can be useful for numerous applications, some of the most

common applications being understanding scenes, medical image analysis, robot perception, augmented reality [MBP⁺21], security monitoring, and remote sensing [KKR14]. There are various methods for image segmentation, e.g. thresholding, edge detection, region-based methods, or clustering methods such as k-Means Clustering. In recent years, several deep-learning-based image segmentation methods have been developed and gained a lot of attention due to their breakthrough performance. Some well-known examples are fully Convolutional Networks [LSD15], Encoder-decoder based models [BKC17], Region-based Convolutional Neural Networks (R-CNN) (for instance segmentation) [RHGS15], [HGDG17] and Recurrent Neural Network-based models [VCR⁺16].

4.2 Biomedical image segmentation

Healthcare is a broad and vital application area for image segmentation. One of the most difficult tasks in medical image analysis is the segmentation of medical images and the identification of pixels of organs or lesions from medical images, such as CT or MRI images, so as to provide important information regarding the form and volume of these organs. The main challenge in this field is the variation of image quality and the unavailability of a vast amount of labelled data for some diseases [GDDM19]. Biomedical image segmentation methods are widely used in various fields such as diagnostic, localisation of tumours or other pathologies, planning of various treatments, and computer-integrated surgery [PD13]. The most frequent application is the segmentation of tissues in order to, among other things, count or detect cancer cells [GDDM19].

Doctors typically examine the medical images manually, which is not only time-consuming, but also subjective [PD13]. In order to overcome these limitations, various image segmentation techniques have been introduced to enable the automatic analysis of a large number of medical images.

In recent years, Artificial Neural Networks and Deep Learning have attracted more and more attention and achieved many significant improvements and breakthroughs in this field [MBP⁺21].

Over the past years, the U-Net architecture has achieved remarkable results in a wide range of biomedical image segmentation areas, such as the application of U-Net in X-ray imaging [BDVDGS⁺18], ultrasound imaging [KWG⁺18], or CT imaging [TLC⁺18]. Further explanation of the U-Net architecture can be found in Section 4.8. Based on the promising results of the U-Net architecture in biomedical image segmentation, we presume that it may work well on flow cytometry data.

In order to understand the essential building blocks of the method, the basics of Neural Networks are presented in the next chapter.

4.3 Fundamentals of Neural Networks

This section gives an overview of the background of neural networks, starting with the basics of feed forward neural networks.

The history of neural networks leads back to 1958 when Rosenblatt [Ros58] implemented the perceptron, which is a single layer neural network. The book of Minsky et al. [MP69] demonstrated the computational limitations of the single-layer perceptron. As a linear model, the perceptron can learn limited classes of non-linear mappings by using non-linear feature transformations that are fixed prior to training. However, it is not able to learn general non-linear mappings from training data, because the dimensionality of the feature space would grow too fast with the number of training data (curse of dimensionality). In order to address this shortcoming, the Multi Layer Perceptron (MLP, or feed forward neural network), which is capable of learning non-linear functions, was introduced. In 1989, Hornik et al. [HSW89] proved with the Universal Approximation Theorem, that any *Borel measurable* function can be approximated with an MLP.

We refer to these models as feed forward seeing as information passes through the function of x , through the intermediate computations used to define f , and finally to the output y [GBC16]. These networks contain an input layer, hidden layers, and an output layer (see Figure 4.3).

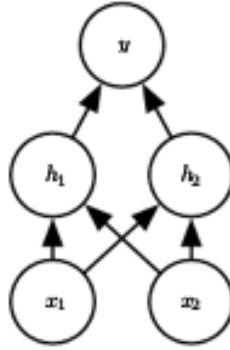


Figure 4.1: A simple feed forward network with one hidden layer containing two units [GBC16]

For the output of each unit of a feed-forward network, an activation function is applied. Activation functions usually map an input to an output using a non-linear function [AZH⁺21]. Non-linear activation functions are commonly used if the underlying data structure is complex or difficult to learn. An important factor in choosing an activation function is that it has to be differentiable so as to allow the weights to be optimised using gradient descent [SSA17].

Some frequently used activation functions include:

- Sigmoid: When the sigmoid activation function is used, the output is an S-shaped function between 0 and 1.

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4.1)$$

- ReLU: The ReLU (rectified linear unit) activation function is defined as follows:

$$\text{ReLU}(x) = \max(0, x) \quad (4.2)$$

The outputs of the function are positive values. A benefit of the ReLU function is that it only activates some neurons and not all at once since negative values are set to 0 [JDV⁺17]. An additional benefit of the ReLU function is, that it is partly linear, which means it is easy to optimize with gradient based methods [GBC16]

- Softmax: Multiple sigmoid functions combined to solve classification problems with multiple classes:

$$\text{Softmax}(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K \quad (4.3)$$

It provides for each data point the probability that it belongs to a particular class. [JDV⁺17]

4.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have attracted a lot of interest over the last decade due to their breakthrough performance [AMAZ17]. The history of CNNs leads back to the 1980s when Kuniyiko Fukushima introduced the convolutional and downsampling layers [FM82]. The name refers to the fact that the network implements a mathematical operation called *convolution*, a special type of linear operation [GBC16]. One major assumption of a CNN is that the features do not have spatial dependencies [AMAZ17]. A Convolutional Neural Network is a feed forward network.

There are numerous reasons as to why Convolutional Neural Networks are beneficial for solving image classification tasks. The three most important characteristics, which may in turn help to improve a machine learning system, are sparse interactions, parameter sharing (or weight sharing), and equivariant representations [GBC16]. CNNs share these weights, i.e. they use them more than once, which reduces the number of parameters as well as the computational complexity. Sparse interactions indicate, that the kernel size used has smaller dimensions than the input image. CNNs have the property of being translation equivariant, i.e. if the input changes, the output changes correspondingly [GBC16]. An additional advantage is spatial invariance; the location of an object is often irrelevant for image classification tasks given that we are solely interested in detecting the object [ZLLS21].

4.4.1 Layers

Convolutional Neural Networks consist of multiple layers with different functionalities, i.e. convolutional layers (see Section 4.4.1), pooling layers (Section 4.4.1) and fully connected layers (Section 4.4.1). In this section, all 3 layers will be elaborated upon.

Convolutional layer

In Convolutional Neural Networks, the convolutional layer is the main building block consisting of kernels (also referred to as convolutional filters) [AZH⁺21].

To generate a feature map, a kernel, consisting of learnable weights, is shifted over the input vector. This operation is called a convolution operation. We will focus on the convolutional operation for 2D images since this will be used in our implementation. The 2D convolutional operation can be written as:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n). \quad (4.4)$$

where I is a 2D input and K is a 2D kernel [GBC16].

Figure 4.2 illustrates the first step of a 2D convolutional operation, where the kernel strides through the input matrix and generates a feature map.

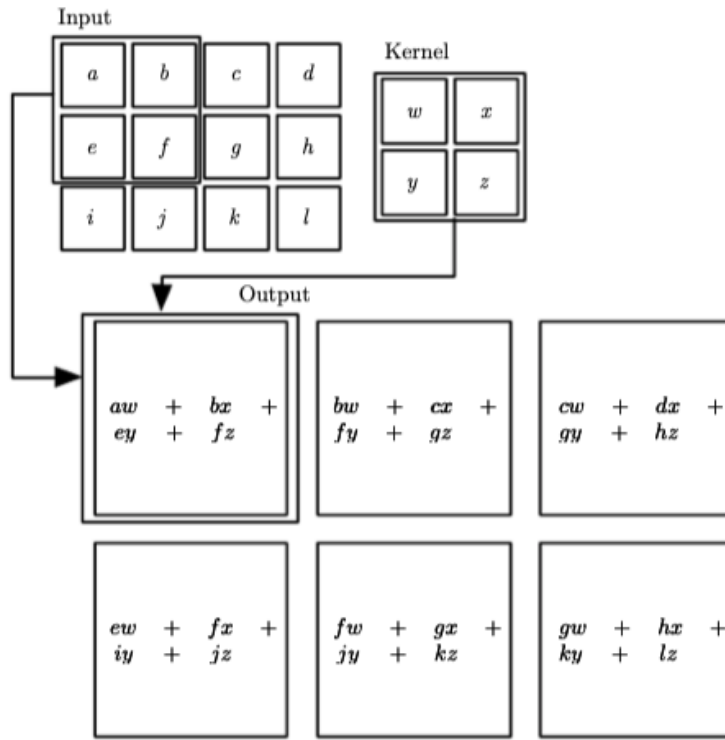


Figure 4.2: Visual representation of a 2D convolution [GBC16].

The output of a convolutional layer can be calculated as follows:

$$a_{ij} = \sigma((K * X)_{ij} + b) \quad (4.5)$$

where a_{ij} is the output for location (i,j) , K is a kernel which slides over the input (X) , b is the bias, $*$ represents the convolution operation and σ is the activation function (see Figure 4.3) [IGMA18].

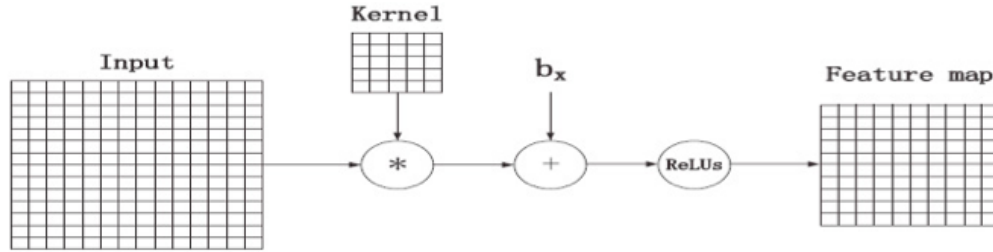


Figure 4.3: Visual representation of a convolutional layer [ELL⁺17]

Note that pixels located close to the border of an image may be neglected when applying the convolutional operation. A solution for the problem is padding, which means new irrelevant pixels (usually zeros) are added to the edge of the image. By doing so, all the important information will be preserved. In addition, including padding changes the number of pixels in the output, which is commonly used to receive an output image of the same size as the input image. [ZLLS21].

When calculating the convolutional operation, the kernel moves over the input matrix starting from the top left corner. In each step, the kernel moves x pixels horizontally. When the kernel cannot move any further to the right, it jumps y pixels vertically and back to the first column. x and y are referred to as stride parameters. Not only the parameter padding but also stride changes the size of the output image [ZLLS21]. The default value for stride is usually 1. An example with a vertical and horizontal stride of 1 and padding of 0 is illustrated in Figure 4.4.

Convolutional Neural Networks can be divided into three parts: feature extraction, which is performed by the convolutional layer and the pooling layer, and classification, where the fully connected layer converts the features into the desired output [YNDT18].

Pooling layer

The main task of the pooling layer is dimensionality reduction [AMAZ17], which is accomplished by combining the feature maps [AZH⁺21]. In comparison to the convolutional layer, the pooling layer replaces the output with some summary statistics of the nearby outputs [GBC16], it has neither a kernel nor parameters [ZLLS21]. Well-known pooling methods are average-pooling, min-pooling, and max-pooling, where max-pooling is the most frequently used method [IGMA18].

It divides the image into sub-region squares and returns only the maximum value within

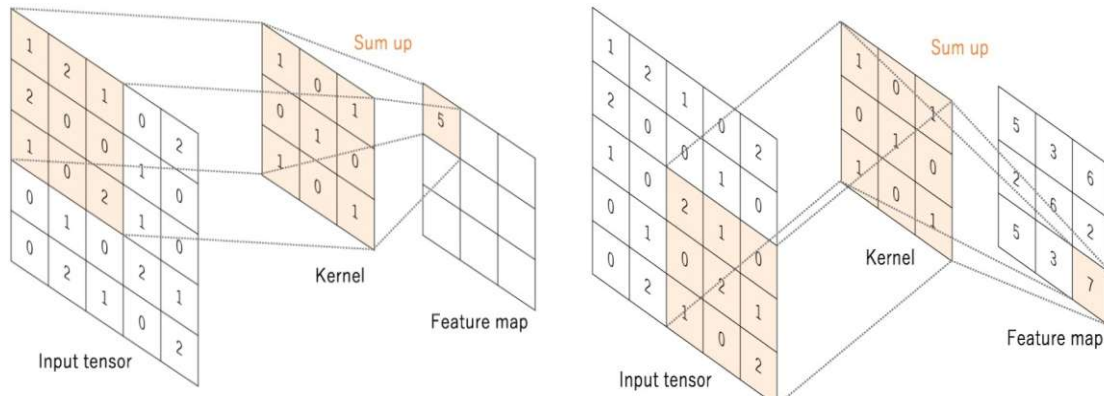


Figure 4.4: Convolution operation using kernel size 3×3 , no padding, and stride 1 [YNDT18]

that sub-region [AMAZ17]. The main idea of max pooling is downsampling to reduce the dimensionality (i.e. the resolution) of the images.

Fully connected layer

The previous layers can be considered as the feature extraction part of the [CNN], where the fully connected layer in combination with a softmax activation function is responsible for the classification. The objective of the fully connected layer is to learn the mapping between the feature maps and the class probabilities. One (or more) fully connected layer is applied to the features which were detected by the convolutional and pooling layer, in order to calculate the class probabilities or any other downstream tasks [WBAK20].

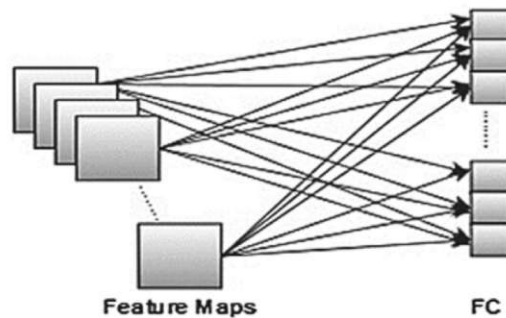


Figure 4.5: Fully Connected layers [WBAK20]

The output of the last convolutional layer is connected to each hidden unit of the first fully connected layer (see Figure 4.5). It should be noted that a major disadvantage of fully connected layers is that they are prone to overfitting.

4.5 Training a neural network

When training a network, the goal is to find kernels and weights that minimise the given loss function. A frequently used training procedure is gradient descent, an optimisation algorithm. It updates the weights in such a way that the loss function (cost function) $E(\mathbf{x})$ is minimised [YNDT18].

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n\|^2, \quad (4.6)$$

where x_n is the input data and t_n is the corresponding target variable. The primary goal of the gradient descent algorithm is to find a weight vector (\mathbf{w}), so that the loss function is minimized. If we vary the weight vector by a step-size $\mathbf{w} + \delta\mathbf{w}$, then the error function changes by $\delta E \simeq \delta\mathbf{w}^T \nabla E(\mathbf{w})$ where $\nabla E(\mathbf{w})$ (the gradient of the loss function) points in the direction of the greatest increase [BN06].

Since $E(\mathbf{w})$ is a smooth continuous function of \mathbf{w} , it will reach its minimum at the point $\nabla E(\mathbf{w}) = 0$.

An iterative procedure is used for solving this equation, which starts with an initial value \mathbf{w}_0 and updates \mathbf{w} the following way:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \Delta\mathbf{w}^{(\tau)}, \quad (4.7)$$

where τ stands for an iteration step.

Gradient descent uses the parameter η ($\eta > 0$), which is the learning rate:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}) \quad (4.8)$$

In such an iterative way, a new weight vector will be calculated in each step. This method is called gradient descent (steepest descent) seeing as the weight vector is moving in the direction of the largest rate of decrease of the error function [BN06].

The gradient of the error function $E(\mathbf{w})$ still needs to be evaluated with a method called backpropagation. The first step (propagation of errors backward through the network) is to calculate the derivatives of the error function with respect to the weight vector. In the second phase (weight adjustment), the derivatives are then applied to compute the adjustments to be made to the weights [BN06].

The network training can be summarized in the following steps:

- Step 1: Provide input vector to the network.
- Step 2: Perform convolution using filters in order to obtain a feature map, use an activation function and pooling operations
- Step 3: Before classification, a fully connected layer is applied on the feature map

Step 4: This resulting output is passed to a classifier (eg. softmax).

Step 5: Compute loss function and calculate gradient

Step 6: Backpropagate the error component and update the parameters.

Step 7: Perform the forward pass and repeat Steps 2 to 6 using updated parameters until the network converges [IGMA18].

In the case of a differentiable function, gradient descent works efficiently. In many machine learning applications, the objective function consists of a sum of sub-functions, each of which is evaluated for a subsample of the training data. For such stochastic objective functions, stochastic gradient descent can be more efficient by performing gradient steps with respect to the individual subfunctions [KB14].

4.6 Regularisation of a Neural Network

A common challenge in machine learning is to train models which not only work well on the given data but also generalise for unseen data. There are several strategies that are used to decrease the test loss, even if it partially increase the training loss. These methods are called regularisation methods [GBC16].

Two main challenges in machine learning are overfitting and underfitting. Overfitting means that the method performs well on the training data but performs poorly on the test data, whereas underfitting occurs when the method cannot reach a good performance on the training set [GBC16].

The best strategy to make a neural network generalise better is to use more training data, which can be achieved by data augmentation methods.

4.6.1 Data Augmentation

The performance of machine learning and deep learning models depends largely on the proper quality and quantity of training data. In several tasks, such as biomedical applications, it is very difficult or even impossible to collect enough data for training. If the amount of data used for training is not sufficient, the trained model may very well fit properly the data used, but not perform well on new, unseen data, hence overfitting.

To address this problem, several methods have been introduced which can generate new images with the help of "simple" transformations using the given labelled data [MG18]. These methods are called Data Augmentation methods. The extended training set (including the generated transformed images) can help the model to learn less specific characteristics of the data, so that the trained model is more generalised and performs better on the independent test set [CMV⁺21]. In computer vision, data augmentation is essential so as to reduce the generalisation error and achieve sufficient results, given that CNNs commonly contain millions of parameters [HGYR17].

Traditional, straightforward methods of data augmentation such as rotations and shifts are frequently used in computer vision tasks. Affine transformations such as scaling, rotation, and colouring have proven to perform reliably in generating augmented training images with high quality [MG18].

Ronnenberger et al. [RFB15] suggest using random elastic transformations to generate annotated images. The main idea is to use smooth distortions with random displacement vectors (on a 3×3 grid). The displacements are obtained from a Gaussian distribution with a standard deviation of 10 pixels.

The choice of data augmentation methods heavily depends on the use case and therefore it requires additional domain knowledge and will be further described in Chapter 5.

4.6.2 Early stopping

In many cases, the training error decreases steadily over time, but after a few epochs, the validation error starts to increase again (see figure 4.6). This indicates that the model with the lowest training error does not necessarily perform well in the validation and testing set. Furthermore, continuing training after the point where the validation loss once again increases does not significantly improve model performance. Instead, an early stopping criterion can be used, which saves the model with the lowest validation loss [GBC16]. Another advantage of early stopping is that it can drastically reduce training time compared to simply terminating after a certain number of epochs. An example of an early stopping criterion would be stopping when the validation loss does not improve for a predefined number of epochs, the so-called patience.

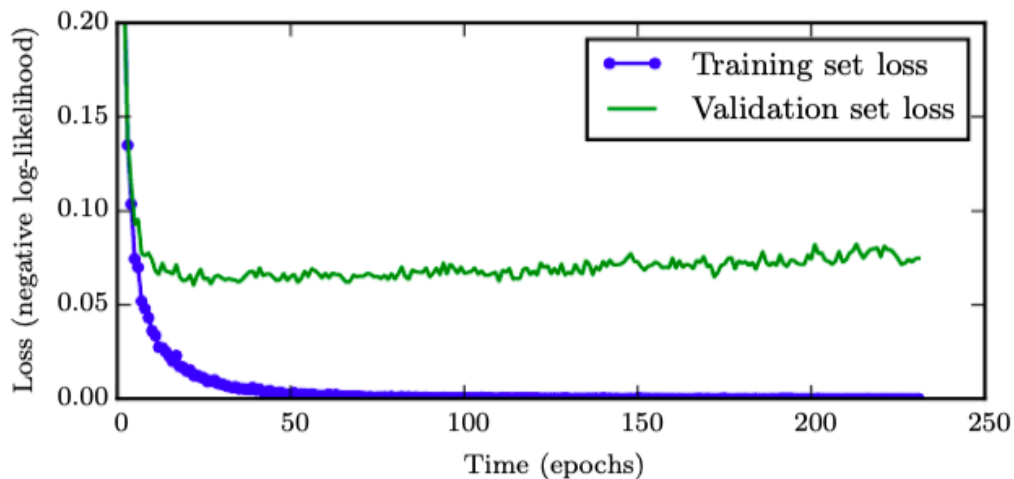


Figure 4.6: Illustration of the need for early stopping. The negative log-likelihood loss was visualised for the training and a validation set [GBC16].

4.7 Optimisation of a Neural Network

In order to find the most suited model, the empirical risk which is the expected training error needs to be minimized. The choice of the different loss functions and the search for the best model parameters will be discussed below.

4.7.1 Loss function

A Loss function (cost function) measures the discrepancy between target values and network output. The main goal is to have predictions which lie as close as possible to the ground truth, which means the expected loss function needs to be minimised [BN06]. During the training, the training loss and its derivative (gradient) are calculated and then used for the propagation part of the training, followed by the update of the weights with their respective gradients [WBAK20].

The choice of a loss function highly affects the model performance, and the choice of the "best" loss function highly depends on the data itself [Jad20]. The most commonly used loss function for regression tasks is the Mean Squared Error.

$$E = \frac{1}{n} \sum_{i=1}^n e_i^2 \quad (4.9)$$

where e_i represents the difference between the target output and the prediction [WBAK20]. Cross Entropy Loss is a commonly used loss function for classification tasks and can be calculated in the following way:

$$CrossEntropyLoss = - \sum_i y_i \log(\hat{y}_i) \quad (4.10)$$

where y_i is the true label and \hat{y}_i is the estimation of class i .

Image segmentation tasks can be considered as a classification problem, but instead of classifying the entire image, the prediction is performed for each pixel individually. According to a survey by Shruti Jadon [Jad20], binary cross entropy is best suited for image segmentation tasks when the classes are balanced.

The evaluation metric Dice Similarity is a frequently used evaluation metric for solving image segmentation problems. The Dice Loss is a Loss function based on this evaluation metric [MCN⁺21].

$$DiceLoss = 1 - \frac{\text{Intersection of the predicted mask and the ground truth}}{\text{Union of the predicted mask and the ground truth}} \quad (4.11)$$

The minimum of the dice loss is 0, which indicates a perfect fit since the predicted mask perfectly matches the ground truth.

Cross Entropy Loss and Dice Loss are well suited for the U-Net architecture. Both are capable of measuring the discrepancy between a target segmentation mask and a predicted segmentation mask.

4.7.2 Parameter tuning

There are numerous parameters in the case of training an artificial neural network which highly affect the performance of the model. Learning rate, batch size, depth of the network, and the number of hidden units are among the parameters that must be carefully selected.

The learning rate is an important parameter to optimise when training a neural network, it determines the step size in each iteration [Mur12].

Using adaptive learning rates, the model performance can be further improved. ADAM (Adaptive Moment estimation) is a commonly used algorithm for stochastic optimisation due to its efficiency when working on a large amount of data, a vast amount of parameters, and low memory demand. It computes an adaptive learning rate for various parameters using estimates of the first and second moments of the gradients [KB14].

Optimization algorithms for machine learning commonly update the parameters based on an expected value of the cost function. Calculating this value for all samples is usually computationally expensive and therefore only a subset of the samples will be used for this calculation. Batch size indicates the number of training instances used before the weights are adjusted. Algorithms that use all the training sets at once are called batch methods, only part of the samples minibatch methods and only one sample stochastic methods [GBC16]. The higher the batch size, the more accurate the results in each step, although it entails higher computational complexity.

The choice of batch size, the initial learning rate, as well as architectural parameters such as the depth of the network and the number of hidden units, are selected before the actual training process of a model begins. It is common practice to explore the performance of the model using different combinations of those parameters. This process is called parameter tuning. There are several methods used to tune the parameters, e.g. grid search or random search. When using Grid search for parameter tuning, all combinations of each parameter in a given range will be explored and compared. The model with the lowest loss function will be used as the final model [Ben12]. Random search uses random variations of the parameters in order to reduce computational complexity.

4.8 U-Net Architecture

In this Section, the main characteristics of the U-Net architecture will be introduced. The main motivation behind the choice of the method is the goal to implement the hierarchy of the gating procedure (see Section 2.3). In order to select the remaining data for the next level of the hierarchy, the cell's pixel-wise location is crucial. Therefore, a common CNN is not well suited for our task.

We decided on a method that showed great results for solving biomedical image segmentation problems (see Section 4.2), the U-Net architecture. Due to the downsampling and upsampling part of the U-Net, the pixel-wise location of information will be preserved, which is essential for the implementation of the hierarchical gating procedure. The

input of a U-Net architecture is an image and segmentation mask pair. Given the flow cytometry data labelled by medical experts, introduced in Section 2.2, the scatterplots and corresponding segmentation masks can be easily generated. It serves as the ground truth for the model. The prediction of the model is a segmentation mask for a given input image, which allows for the partitioning of the data for the next level of the hierarchy.

Convolutional neural networks are frequently applied to image classification tasks. With some minor modifications, they are also well suited for image segmentation tasks.

To solve a classification task, the required output is a probability distribution over the classes. To achieve this, the images are flattened, which is not the appropriate approach for image segmentation tasks. Through flattening, the images lose their spatial relationships in the image. To solve this problem, fully Convolutional Layers can be used [GDDM19]. Fully Convolutional Networks (FCN) are a special type of CNN which preserve the spatial relationships between the image pixels, which makes it suitable for solving image segmentation tasks [JDV⁺17].

FCNs compared to CNN have upsampling layers to restore the spatial resolution of the input image in the output layer. Since max pooling layers lead to resolution loss, skip connections will be used between the upsampling and downsampling part of the network. This uses stored information from the downsampling path in the upsampling path [JDV⁺17].

Since the aim of this thesis is to solve an image segmentation task in order to detect cancer cells, a fully Convolutional Network, the U-Net, developed by Ronnenberger et al. [RFB15] will be introduced in this section. See Chapter 5 for a detailed description of the U-Net implementation for flow cytometry data.

4.8.1 The Network Architecture

The U-Net architecture is an encoder-decoder-based network. In [RFB15], the architecture was described as follows: *The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localisation.* The encoder part compresses the input image into a latent spatial representation (capturing the semantic information of the input image) and the decoder part predicts the output using the compressed representation [MBP⁺21]. The left part of the architecture, shown in Figure 4.7 is the contracting path (encoding), which can be considered as a normal CNN. It consists of convolutional layers (3x3) with a ReLU activation function followed by max pooling (2x2) as a downsampling operation with stride 2, which are repeated numerous times [SPED21].

The right side of Figure 4.7 corresponds to the expansive path (decoding), where each step consists of an upsampling convolution (2x2) in combination with convolutional layers (3x3) with a ReLU activation function [SPED21]. The upsampling convolution (transposed convolution, see Figure 4.8) is used to reverse the effect of the convolutional operation used in the encoding part of the network [GDDM19].

Figure 4.9 illustrates how a 2x2 dimensional feature map will be upsampled to a 3x3

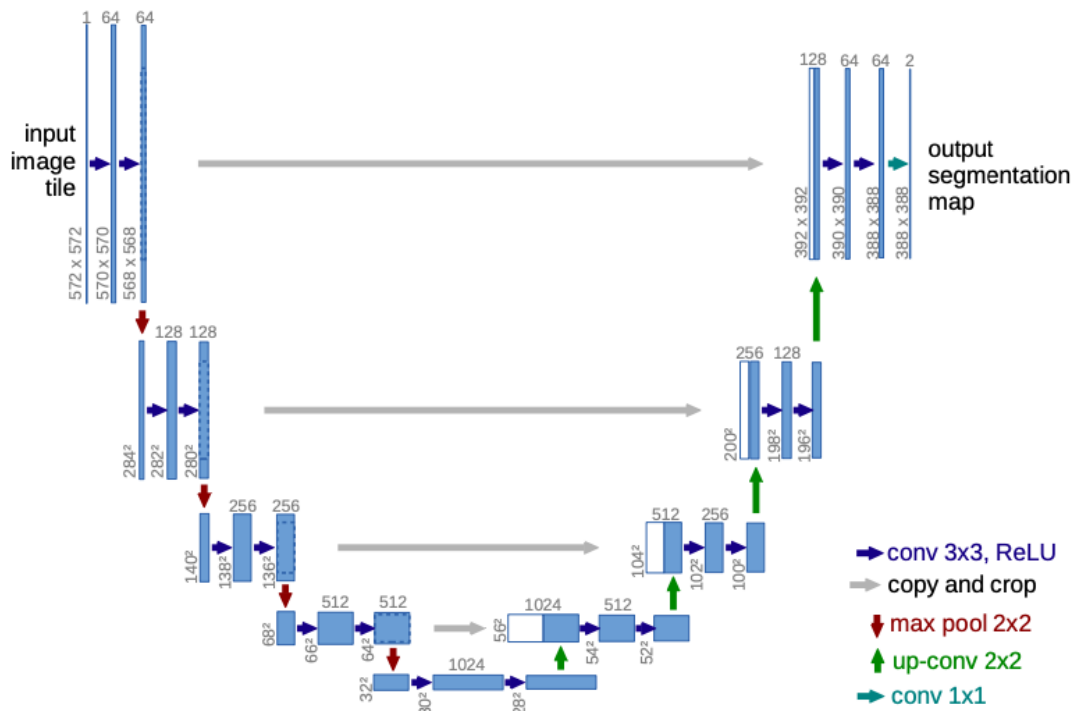


Figure 4.7: U-Net architecture [RFB15]

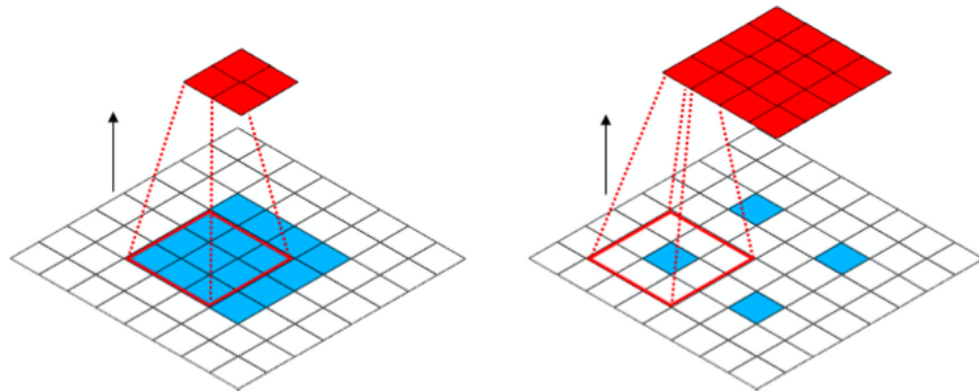


Figure 4.8: Comparison of a convolution and transposed convolution [GDDM19]

dimensional feature map. Each element of the feature map will be multiplied with the kernel and added up, resulting in a 3x3 feature map.

Finally, a 1x1 convolution (i.e. convolutional operation with kernel size 1x1) is applied to the final layer in order to obtain the correct dimensions. By doing so, a segmentation map as output will be generated [SPED21].

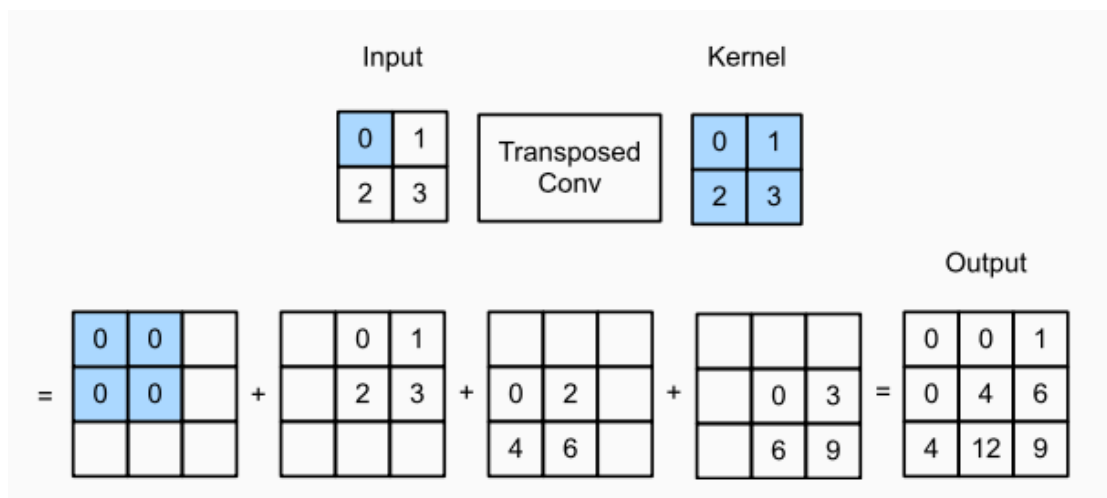


Figure 4.9: 2x2 dimensional example for the upsampling convolution, or with other words, transposed convolution [ZLLS21]

Pooling operations are frequently used to reduce the dimensions (see Section 4.4.1). These operations are used in the encoding part of the network. We now apply an operation called max unpooling. It reverses the previous effect of dimensionality reduction by decompressing the given pixel into more pixels (decoder part of the network). For example, 2x2 unpooling generates four pixels of one (see Figure 4.10) [GDDM19]. Max unpooling uses the information saved by max pooling, i.e. the indices of the pixel with the maximal value of each pooling operation. In the case of max unpooling, the elements will be placed on these indices, while all other values will be replaced by 0.

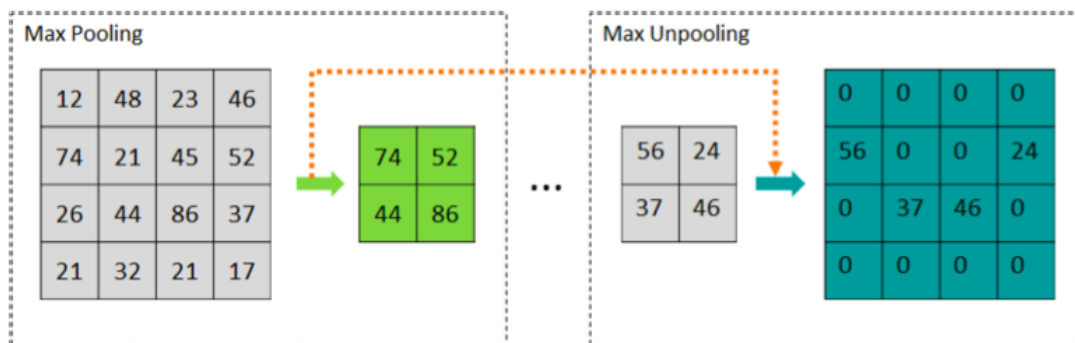


Figure 4.10: Max Pooling and Max Unpooling [GDDM19]

U-Net works with skip-connections, in order to take different levels of abstractions into account and recover the information lost by downsampling [GDDM19] (see arrow copy and crop on Figure 4.7).

One of the main advantages of the U-Net architecture is that, due to context-based learn-

ing, the U-Net is also much quicker to train compared to the most common segmentation models [SPED21].

For our application of the U-Net architecture, we have different inputs, namely, the gates of the gating procedure described in Section 2.3. There are various possibilities regarding how we can deal with this challenge. We can train multiple models (one for each gate) or we can use only one model to learn all gates by applying an extension of the U-Net architecture, called conditioned U-Net.

4.8.2 Conditioned U-Net

A special control mechanism, as described in the work of Perez et al. [PSDV⁺18], allows the usage of a single U-net model for different sources. The inclusion of such contextual information enables the U-Net model to distinguish between samples from different sources.

Conditioned U-Nets use a one-hot encoded vector which contains the information of the source of the sample. FiLM (Feature-wise Linear Modulation) layers were introduced to condition the neural network with feature-wise affine transformations which can be adaptively learned [PSDV⁺18].

$$FiLM(x) = \gamma(z) \cdot x + \beta(z) \quad (4.12)$$

where x is the input, which will be scaled and shifted on the basis of z . z is a vector indicating the source of data. In our implementation, z denotes the gate from which the input image was generated (see Section 2.3). The parameters γ and β are learned when training the model [MBP19].

This linear conditioning method involves an additional domain-specific contextual information (metadata) into the model, which makes the model more robust and leads to an increased model performance as discussed in [LGV⁺21]. By training a single model for different sources, the amount of training data is automatically increased.

The forthcoming chapter details the description of the implementation of both the U-Net architecture and the conditioned U-Net architecture for the automation of the gating procedure.

Application of the U-Net for cancer cell detection in flow cytometry data

This chapter presents the application of the U-Net architecture for the automatic detection of cancer in cells in flow cytometry data based on a sequence of 2-dimensional scatter plots used by medical experts for the manual gating procedure. Firstly, the workflow of the proposed method for automated cancer cell detection will be described in detail in Section 5.1, including data preparation in Section 5.1.1 as well as the creation of 2-dimensional projections of the multidimensional flow cytometry data 5.1.2 and segmentation masks using the labelled data by medical experts in Section 5.1.3. In Section 5.2 the training process will be described, followed by the implementation of the network architecture 5.2.1. Section 5.2.2 describes the data augmentation methods used to increase the number of training samples, model parameter optimization (Section 5.2.4), and hierarchical sub-selection of flow cytometry data using the predicted segmentation masks (Section 5.2.5). The chapter ends with a description of the experiments (Section 5.3) used for the evaluation of the models and for the comparison with the state-of-the-art literature and with an elaboration of the evaluation methods (Section 5.4).

5.1 Workflow for implementing the automation of the gating procedure

The workflow of the implementation can be summarized in the steps shown in Figure 5.1. As mentioned in Section 2.3 we represent the input data by 2D projections along axes which are known to be informative and which are used in the manual gating procedure.

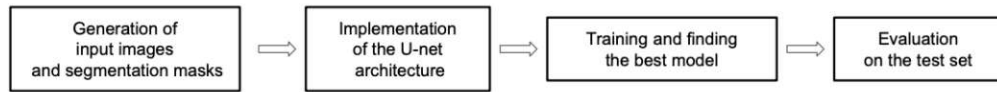


Figure 5.1: The implementation workflow

These representations are chosen a-priori and we must therefore define the image size, resolution, and plot type. Generating the input and target plots and choosing the optimal image size is crucial for the successful automatic detection of cancer cells. Using larger image sizes in Convolutional Neural Networks (CNNs) offers benefits such as capturing richer spatial information and context for improved feature extraction and generalization, but it also introduces challenges like increased computational complexity, memory usage, and potential overfitting risks.

Scatter plots could be used to represent these 2D projections, but we have observed that cells heavily overlap in some areas. A scatter plot displays overlapping cells as there where only one cell, in other words, we lose information regarding the density of cells. Thus, we chose a plot type that is able to capture the density of cell populations, namely the scatter density plot. The target plot represents the segmentation masks that capture all cells of the associated gating step which were selected by medical experts.

The second step is the actual implementation of the U-Net model architecture. The implementation is based on the Github repository [Per21] with adaptations regarding the number of channels of the images, the size of the feature maps, and the initial weights.

The implementation of the U-Net model architecture is followed by the training of the model. In this part, a U-Net model will be trained separately for each step of the gating procedure. There are several parameters (e.g. number of layers, size of the feature map, learning rate, loss function, and number of epochs) needed for the training which must be tuned to find the best possible model.

Once the models are trained, for each image in the test set a segmentation mask can be predicted analogous to the polygons outlined by medical experts in the gating procedure, which serves to exclude irrelevant cells for subsequent stages. Using the predicted segmentation masks this can be easily reproduced and the new (reduced) test image for the next gate can be generated. The last step is the evaluation which will be further elaborated upon in Chapter 6.

The implementation of the hierarchical approach to reproduce the gating procedure is summarized in Figure 5.2.

5.1.1 Data preparation

Bone marrow samples of leukemia patients can be found and downloaded from the FlowRepository [JS12] in fcs (Flow Cytometry Standard) form.

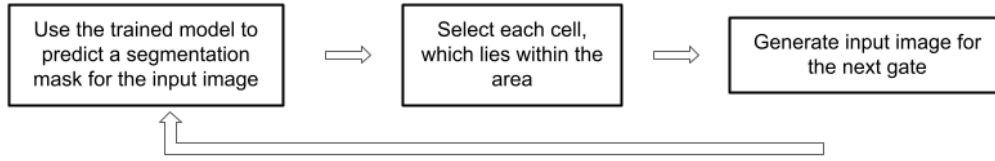


Figure 5.2: Implementation of the hierarchical approach for the automated gating procedure.

gate label	x	y
Syto	FSC-A	Syto41
Singlets	SSC-A	FSC-W
Intact	SSC-A	FSC-A
CD19	SSC-A	CD19
Blasts	CD10	CD45
Blasts	CD10	CD20
Blasts	CD10	CD38

Table 5.1: Features and gate labels used for the creation of input images in the various gating level.

As mentioned in Section 2.2, the FlowRepository is a database for flow cytometry experiments. The *flowmepy* [DWWK] python package was used to convert the data from fcs form into a pandas data frame. Each sample contains on average 300.000 events as well as the corresponding gates labelled by medical experts. Since the datasets come from different sources and were labelled by different doctors, the column names of the different datasets needed to be translated back to a consistent form.

The gathered flow cytometry samples were normalized feature-wise before the creation of the 2D scatterplots (which can be considered as the input of the U-Net model) with the min-max scaling method. In this case, variables with different ranges will be re-scaled into an interval of [0,1]. Normalising the data before creating the images is an important step as it allows the algorithms to reduce (irrelevant) variability of the training set and learn the pattern in flow cytometry faster. [TPX19]

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (5.1)$$

5.1.2 Input image preparation

For each level of the gating hierarchy, different input images will be generated. Table 5.1 contains the different x and y features for each gate.

The events (cells) of the 2-dimensional projections of the flow cytometry data were visualized via the *matplotlib* [Hum07] package in combination with the *mpl-scatter-density*

package [mpl20](#) using a scatter density plot. In this way, not only the x- and y-values but also the density can be displayed. In such a way it is easier to recognise the different clusters (cancer cell populations), see Figure [5.1.2](#).

The correct image size and resolution of the 2D input images is a crucial point for training the U-Net model. If the resolution is too high, the computational complexity and the time required for training explode. Images that are too small help to reduce the training time for the models, but can lead to individual pixels containing many target cells and non-target cells simultaneously, which would consequently make it difficult for the model to separate the two groups.

Since in many cases, several cancer cells are located at the edge of the image, data augmentation methods such as shifting or scaling could remove important cancer cells from the image. Therefore, the images were padded before saving.



Figure 5.3: Example of an input image of the Vienna 14 dataset. The image belongs to Gate 4; the x-axis represents the SSC-A and the y-axis the FSC-A of the given cells.

In Figure [5.3](#) it is easy to recognise that there are two sub-populations. To know which cells will be rejected at this stage of the gating procedure, the corresponding segmentation mask in Figure [5.4](#) needs to be considered.

5.1.3 Generating segmentation masks

The target image of a U-Net model is a (binary) segmentation mask, where pixels with a value of 1 indicate that a pixel belongs to cancer cells and pixels with a value of 0 belong to non-cancer cells. These masks can be considered as ground truth, where the background is coloured black and the foreground white [\[RFB15\]](#). These masks represent the smallest convex set that contains all the cells that have been labelled as cancer cells [\[BDH96\]](#).

The size of the input image and the size of the segmentation mask are the same, such that each pixel in the input image of the segmentation mask has a corresponding label.

Samples that do not contain cancer cells are a rare and special case ([bue](#) dataset 5 samples, [vie14](#) dataset 10 samples, [vie20](#) dataset 29 and the [bln](#) dataset contains no such samples). Since it is not possible to construct a convex hull in these cases and since they are relatively rare, we excluded these samples from both the training and evaluation phases of this work.



Figure 5.4: Example of a segmentation mask of the Vienna 14 dataset (it belongs to the input image in Figure [5.3](#)). The image belongs to Gate 4; the x-axis represents the SSC-A and the y-axis the FSC-A of the given cells.

The predicted segmentation masks of the U-Net are not binary, but take on values from the continuous interval from 0 to 1. Figure [5.5](#) shows an example of the raw predictions for Gate 4. Pixels shown in grey are pixels that cannot be clearly assigned to one of the two groups and therefore indicate the uncertainty of the prediction. One can see that most of the grey pixels are near the boundaries of the selected cells. We need a binary classification to specify cells to be extracted for the next step in the gating procedure. Hence, raw predictions below 0.5 are treated as 0, meaning a pixel that contains no cells of interest. Raw predictions greater than or equal to 0.5 are classified as 1, implying that the pixel belongs to the selected cell population.



Figure 5.5: Raw predictions for a segmentation mask in the continuous interval from 0 to 1.

5.2 Training of the network

The model is trained by a supervised learning approach using the generated 2D images (see Section 5.1.2) and the segmentation masks based on the data annotated by medical experts (see Section 5.1.3). The experiments were conducted using a fixed seed in order to allow the experiments to be reproducible. In order to find the best model for each gate,

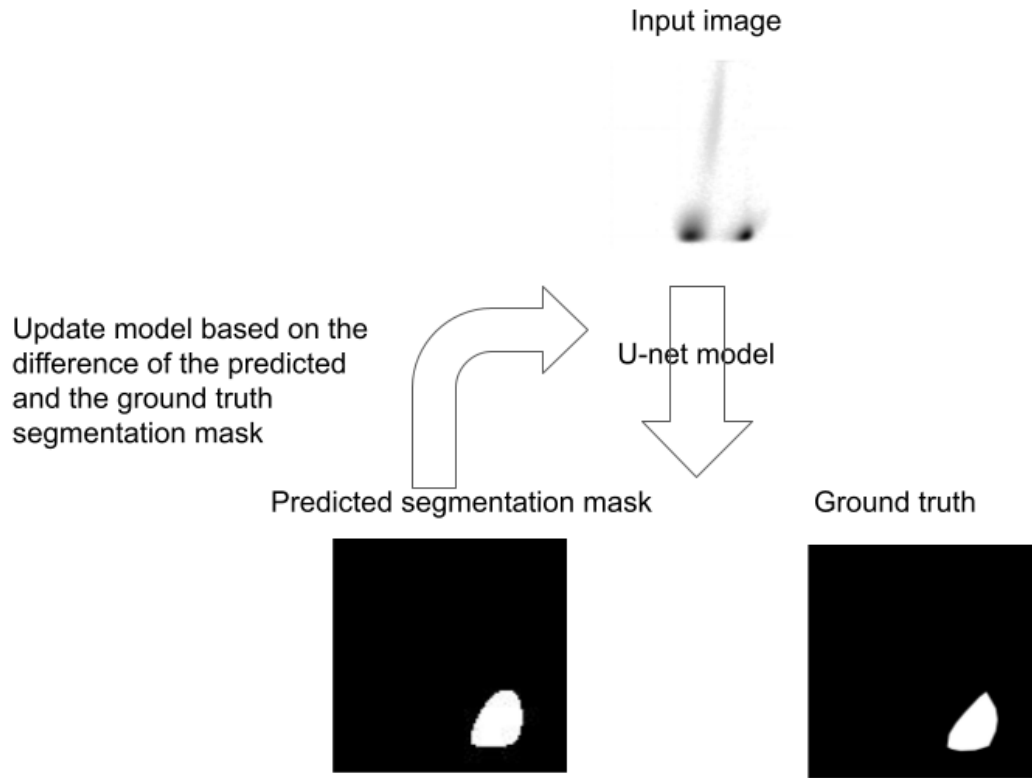


Figure 5.6: Training of the U-Net model.

the Dice loss (see Section 4.7.1) which is frequently used for solving image segmentation problems, was used as a similarity measure. The U-Net was implemented based on the original U-Net paper of Ronneberger et al. [RFB15] and its adaptation in the U-Net model implemented by Aladdin Persson [Per21]. For the entire training process, *Pytorch* [PGM⁺19] was used.

5.2.1 The Network Architecture

As introduced in Chapter 4.8, there are several versions of a U-Net architecture, it always depends on a specific problem to find the most suitable one. The depth of the network (number of layers), the kernel size, and the number of feature channels are the parameters that need to be explored.

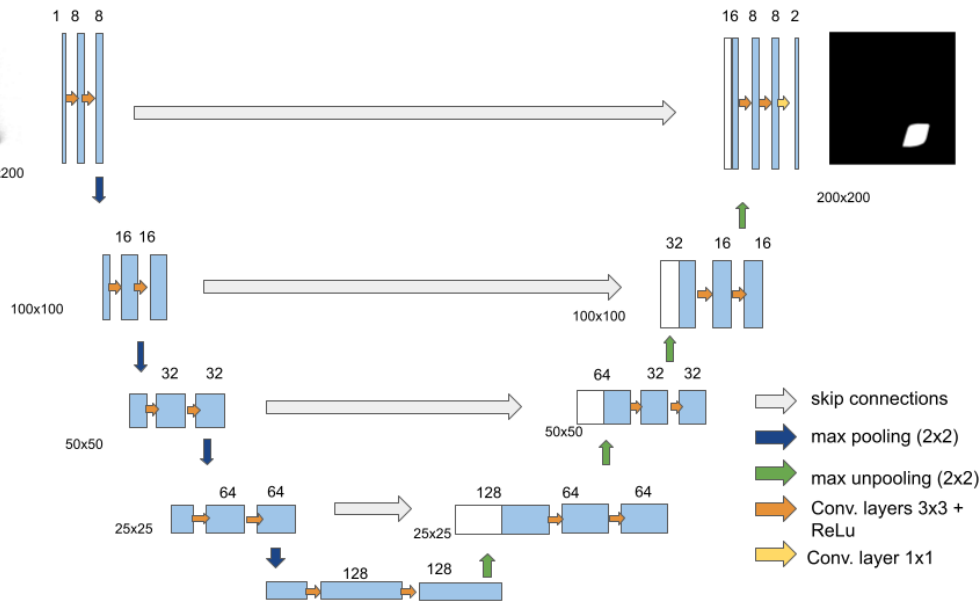


Figure 5.7: Modified version of the U-Net architecture proposed by Ronnenberger et al. [RFB15]

In Figure 5.7 the network architecture is presented. The network architecture has 2 paths, the contracting path (left) and the expanding path (right). The contracting path contains padded convolutions in order to keep the spatial dimensions of the input image (in the original U-Net paper, unpadded convolutions were used) followed by a ReLu activation function as well as a max pooling operation with stride 2 and kernel size 2. This max pooling step can be considered a downsampling step. The number of input and output channels is 1 since we only work with one colour (greyscale) in the segmentation mask. In order to find the most appropriate network architecture and the optimal values for the number of layers, feature channels, and kernel size, we perform a grid search using the parameters described in Table 5.2.

The parameters proposed by Ronnenberger et al. [RFB15] were used as initial values for the number of layers. We decreased the size of the feature channels due to the lower resolution of the input image. Since the complexity of the problem described in the original work is assumed to be different from the proposed problem, we perform experiments with increasing and decreasing these parameters (see Table 5.2).

The best-performing model architecture uses a kernel size of 3x3, number of layers 5 and feature channels starting with 8 and doubled at each level of the contracting path of the U-Net architecture and the respective inverted features on the expanding path, as described in Table 5.2. The size of the input image is resized to 200x200 (for the explanation and causes see Section 6.1.1), so the input image has the dimensions of 200x200x1. After the first two convolutional layers and max pooling operations, the

5. APPLICATION OF THE U-NET FOR CANCER CELL DETECTION IN FLOW CYTOMETRY DATA

kernel size	number of layers	feature channels	med F1-score	med precision	med recall
3x3	3	4	0.0	0.0	0.0
5x5	3	4	0.0	0.0	0.0
3x3	3	8	0.58	0.55	0.94
5x5	3	8	0.23	0.15	0.98
3x3	3	16	0.31	0.2	0.97
5x5	3	16	0.72	0.64	0.97
3x3	3	32	0.39	0.36	0.96
5x5	3	32	0.57	0.5	0.97
3x3	4	4	0.29	0.17	0.96
5x5	4	4	0.79	0.76	0.97
3x3	4	8	0.66	0.53	0.97
5x5	4	8	0.72	0.61	0.93
3x3	4	16	0.66	0.6	0.98
5x5	4	16	0.52	0.47	0.97
3x3	5	4	0.0	0.0	0.0
5x5	5	4	0.84	0.81	0.98
3x3	5	8	0.92	0.91	0.96
5x5	5	8	0.8	0.78	0.96
3x3	6	4	0.79	0.88	0.91
5x5	6	4	0.16	0.74	0.12

Table 5.2: Grid search for the best fitting model structure using the vie dataset.

image has a size of 100x100x8. On the bottom part of the architecture two convolutional layers are used but without any max pooling operation. In the expanding path, the downsampling path can be reversed using transposed convolutions. The result of the transposed convolutions and the result of the corresponding convolutional layer in the downsampling path is concatenated. The concatenated data is then used as input for the next layer in the expanding path. Finally, a 1 by 1 convolutional layer is used to change the number of channels to binary.

The input images will be used to iteratively improve the model performance with the stochastic gradient descent algorithm 4.5. For the training, the weights and the bias need to be initialised, which was done as proposed in the original U-Net paper of Ronneberger et al. [REB15]. The initial values for the weights were drawn from a normal distribution with a standard deviation of $\sqrt{\frac{2}{\text{number of input nodes}}}$.

Conditioned U-Net architecture

We used a Conditioned U-Net Architecture to investigate the question of whether one model for all gates or separate models for each gate are required to learn the detection of

cancer cells. The conditional U-Net architecture allows for a single model to be trained and still distinguish between the different gates by using the source of the sample as an one-hot encoded vector. We trained a conditioned U-Net model on the last steps of the gating procedure (as in Figure 2.4) in order to compare the results with state-of-the-art methods. We chose $[1,0,0,0]$ as the one-hot encoded vector for Gate 4, $[0,1,0,0]$ for Gate 5, and so on.

5.2.2 Data Augmentation

Data augmentation is an important part of the work in order to increase the amount of training data and improve regularization. This is particularly important for small datasets such as [bue](#) (65 flow cytometry samples) and [bln](#) (79 flow cytometry samples). The albumentations package [\[BIK⁺20\]](#) was used to create several augmentations on the training set with the probability parameter $p=0.5$, which means that the transformation was only applied on roughly 50% of the samples in each epoch. The augmentation methods were not applied to the validation or test set. The applied methods are:

- **Scale:** The selected cells usually form a cluster. The variance of the cluster is part of the patterns which we want to learn. We observed that the variance of the cluster is similar across instances. If we allow extremely large scaling, we would generate samples that do not represent the input data. Therefore, the variation of the cluster size should not differ too much from the original one, so we choose a scaling limit of 0.1 for the x and y-axis.
- **Shift:** Since the location of the selected cells is essential information for the model to learn, we do not want the augmentation to shift the cells out of a region, which we would normally consider a region of interest. We tried several parameters and found that values significantly above 0.1 push the cells too far away and values below 0.1 make no noticeable difference. We, therefore, chose a shift parameter of 0.1.

While scaling and shifts maintain the content of the original image, elastic transformations introduce localized deformations and alterations to the shape. Elastic transformations have proven to perform well on several image segmentation tasks, such as the MNIST dataset [\[SSP⁺03\]](#) or cell segmentation tasks performed by Ronnenberger et al. [\[RFB15\]](#). In our case, these distortions made the input images useless, due to the contextual disruption of the patterns that we want to learn. Therefore, we did not use the elastic transformation augmentation method for training the model.

5.2.3 Training implementation

Image and segmentation masks need to be considered as a pair in order to train the image segmentation model. The default data loader from pytorch does not allow loading pairs of images. We had to perform a slight modification of the loader class so that the

Hyperparameter	Description	Values
Batch size	The number of instances used before the model weights are adjusted.	5,30
Loss function	Name of the loss function.	CrossEntropyLoss DiceLoss
Learning rate	Learning rate for the optimization.	1e-3, 1e-4, 1e-5, 1e-6, 1e-07

Table 5.3: Values used for the hyperparameter tuning.

image and mask pair can be loaded and augmentations can be performed simultaneously. The input images have a size of 500x500, but they were resized during the preprocessing to a dimension of 200x200, since it is the most suitable size considering the trade-off of image size and run time (see Section 6.1.1). Therefore, the dimension of the output images is 200x200.

The choice of the loss function is an important part of the training. We used the *DiceLoss* function, which is often applied in image segmentation tasks. We also conducted trials with alternative loss functions, including *Binary Cross Entropy with Logits Loss*. However, the results did not match the performance achieved when utilizing *Dice Loss* as the selected loss function.

5.2.4 Optimizing model parameters

In order to explore how to achieve the best possible performance, the training image with the ground truth and predicted segmentation mask is continuously visualised (see Figure 5.2). In the initial diagram (epoch 0), the predicted segmentation mask is depicted through scattered individual points spread across the image. By the third epoch, the segmentation mask begins to exhibit a form that is akin to the ground truth. Remarkably, by the 35th epoch, the segmentation mask attains a shape closely resembling the ground truth.

As mentioned in Section 4.7.2, there are numerous parameters when training an artificial neural network which highly affect the performance of the model and there are several methods for the exploration of the parameters (see Section 4.7). We used the Grid search method in this thesis in order to find the best combination from our set of parameters.

The learning rate is an important parameter to optimise when training a neural network. The learning rate was used in combination with the Adam optimizer for the training, which is an adaptive learning rate optimization method [GBC16]. To avoid overfitting, a stopping criterion is used for the training. As discussed in Section 4.6.2, the training loss can continuously improve with each epoch, but the model is too tailored to the training set. To address this concern, we adopted a widely employed stopping criterion. This criterion involves monitoring the validation loss, and if there is no improvement observed after 20 iterations, the training process is stopped and the model associated with the

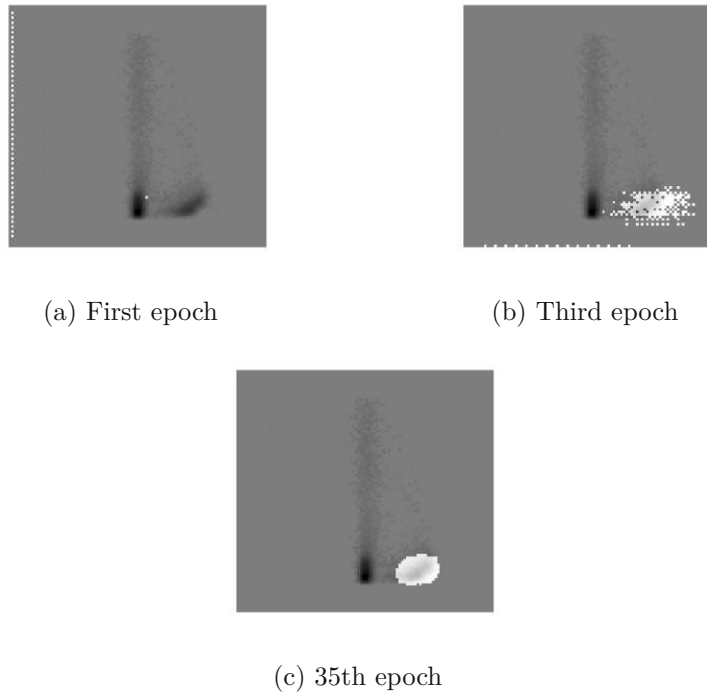


Figure 5.8: The input images and the predicted segmentation mask (white area) for three different epochs using the validation set of the vie dataset.

lowest validation loss is saved as the best model. We used an epoch size of 4000 to allow for a long learning process in case the validation loss still improves.

5.2.5 Partitioning via segmentation masks

The main goal of this thesis is to automate the gating procedure, and one part of the gating procedure that has not yet been addressed is the extraction of the cells between two gating steps. The result of our U-Net model is a segmentation mask that can be used to partition the data. The input data can be partitioned into as many bins as there are pixels in the output of the model. This allows us to determine for each pixel whether it contains cells of interest (labelled 1) or not (labelled 0). Cells located on pixels labelled 0 (black pixels) are discarded for the next step of the gating procedure. After partitioning, the input image for the next gating is generated from the remaining cells.

5.3 Experiments

To allow for a good comparison, the same experiments are used as in the state-of-the-art literature. As described in Section 3, Wödinger et al. [WRW⁺22] used a transformer model on the multidimensional flow cytometry samples.

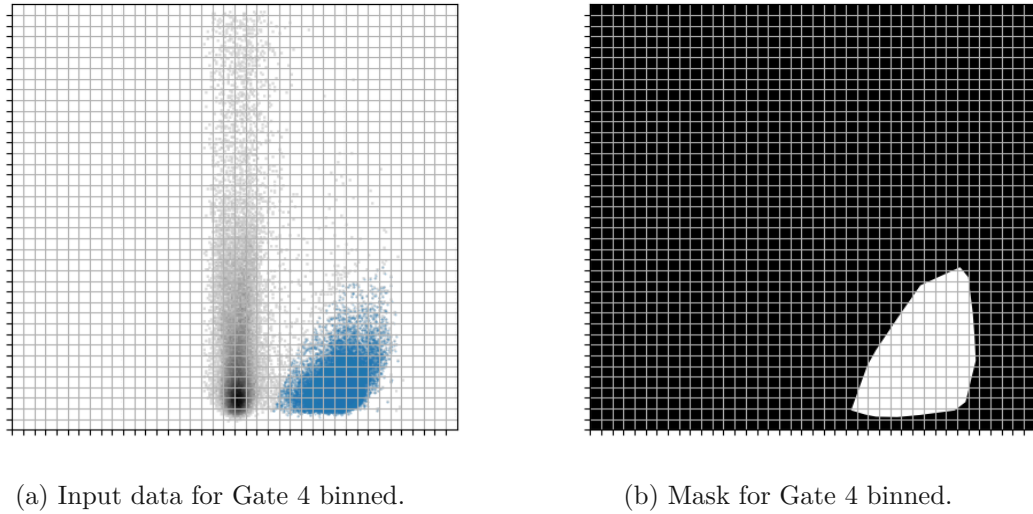


Figure 5.9: Imitation of the partitioning of the input data based on the predicted segmentation mask. The cells which lie on a black pixel on the right visualisation will be rejected for the next gate. The blue marked cells will be selected and used for the next gate. This example uses a 40x40 grid for visualisation purposes, in the experiment 200x200 pixels and grids will be used.

The four datasets containing bone marrow samples of leukemia patients 15 days after therapy (see Section 2.2) will be used. For the evaluation of the proposed method, these datasets are divided into the following parts: training set, validation set and test set (see Figure 5.10). The state-of-the-art methods use only the last two gates for the evaluation of the experiments. For comparison, we therefore only evaluate our experiments on the last two gates as well.

In this experiment, we have four datasets. One dataset will be used as training, two as validation and one as the test set. Each model trained with one of the available datasets will be evaluated against one of the other datasets, see Table 5.4. The only exception is the `vie` dataset, which contains samples from the `vie14` and `vie20` datasets. These samples will be randomly split into train, test and validation sets. In this case, the dataset will be considered new or unseen data without any labels. Once the model is trained, the labels of the test set will be used for the evaluation. For each experiment, the datasets were scaled by the variables of the training set (feature-wise) in order to achieve better performance and comparability.

5.4 Evaluation of the model performance

To identify the best fitting model based on the annotated flow cytometry data, a comparative analysis of the model performance using the median F1 score was performed.

Cancer cell population detection is fundamentally a classification problem. Nevertheless,

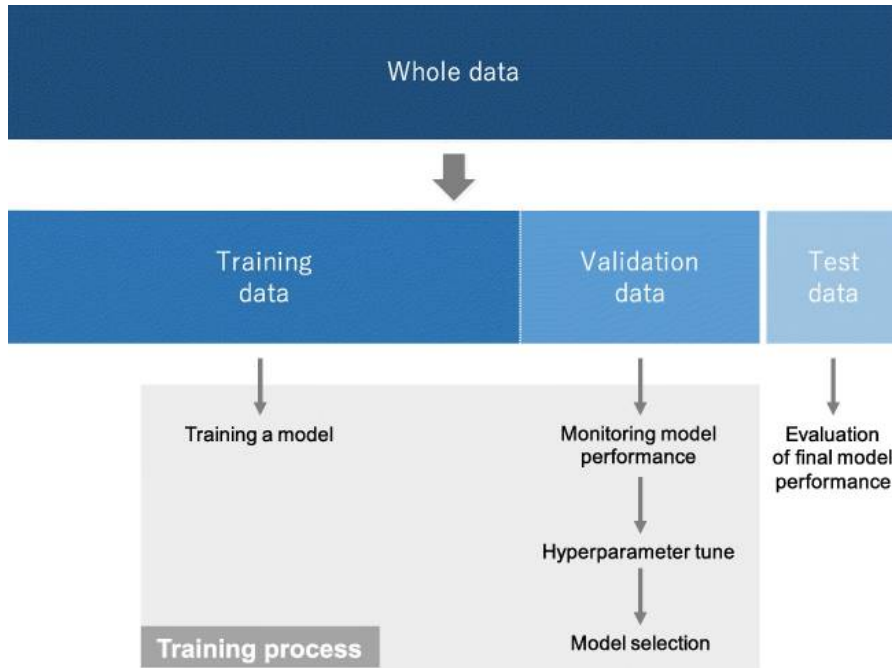


Figure 5.10: Training, validation and test data split [YNDT18]

our strategy employs image segmentation as the method of choice for addressing this task. Therefore we need to reformulate the output of the segmentation problem into a classification problem [HS15], which entails shifting the focus from assessing pixel-wise classification accuracy within the segmentation mask to quantifying the number of cells captured within the image segmentation mask.

5.4.1 Evaluation metrics

Binary classification tasks are usually evaluated using a confusion matrix, see Table 5.5. In this table, the predicted classes (rows) and the "true" classes (columns) are compared.

True positives and true negatives are the correctly classified instances, while false positives and false negatives are the misclassified instances. These can be used to evaluate the performance of a binary classification method.

The following matrices will be used for the evaluation of the classifiers:

F1 score is the harmonic mean between recall (r) and precision (p)

$$\text{F1-score} = \frac{2 * p * r}{p + r} \quad (5.2)$$

Precision: The ratio of correctly positive and all the positive predicted patterns

$$p = \frac{tp}{tp + fp} \quad (5.3)$$

5. APPLICATION OF THE U-NET FOR CANCER CELL DETECTION IN FLOW CYTOMETRY DATA

training set	test set
vie	vie
bln	bue
bln	vie14
bln	vie20
bue	bln
bue	vie14
bue	vie20
vie14	bln
vie14	bue
vie14	vie20
vie20	bln
vie20	bue
vie20	vie14

Table 5.4: Experiments conducted in this thesis.

	Actual Positive Class	Actual Negative Class
Predicted Positive Class	True positive (tp)	False negative (fn)
Predicted Negative Class	False positive (fp)	True negative (tn)

Table 5.5: Confusion matrix [HS15]

Recall: The ratio of positive patterns and all correctly classified patterns

$$r = \frac{tp}{tp + tn} \quad (5.4)$$

The flow cytometry samples are highly imbalanced, indicating that the occurrence of the target variables, i.e., either cancer cells or non-cancer cells, is not evenly distributed across the data set. Such imbalances can cause problems in analysis and modeling, as learning algorithms have difficulty adequately capturing the minority class (in this case, non-cancer cells), often leading to biased predictions. Addressing this imbalance and choosing appropriate evaluation metrics are critical to ensuring accurate and reliable results when analyzing flow cytometry data. In order to deal with this challenge and reduce the influence of outliers, the median F1-score was used for the comparison.

CHAPTER 6

Results

In this chapter, the results of the automated cancer cell detection using the U-Net architecture will be presented. They are compared to the baseline method, namely the Transformer model proposed by Wödinger et al. [WRW⁺22].

The first part of this chapter summarises the insights gained through the implementation, training, and evaluation of the models with a focus on the flow cytometry datasets introduced in Section 2.2. As explained in Section 5.4, the evaluation of the proposed approach is based on the number of correctly detected cancer cells. However, to gain a better understanding and identify potential weaknesses of the model, it is important to evaluate each gate separately.

In the next step, we will examine whether the model performance depends on the number of cancer cells, by comparing ground truth and predictions for different MRD values.

The last part of this chapter consists of the evaluation of the overall results of the automated gating procedure using the U-Net architecture.

6.1 Descriptive analysis

In this section, the insights gained through the implementation of the proposed approach are presented. In this thesis, the manual gating procedure was automated for the detection of cancer cells. The results of the proposed method differ in various scenarios. In this section, the impact of factors such as the image size, the different gates and the number of cancer cells per sample are examined. In addition, the information obtained by evaluating each gate separately is presented.

6.1.1 Analysis of quantization error

The size of the predicted masks used for the partitioning of the cancer cell populations has a great influence on the results of the proposed method. As described in Section

width x height	median F1-score	median precision	median recall
15x15	0.289855	0.985618	0.170068
25x25	0.829283	0.993312	0.743786
50x50	0.988822	0.997113	0.985047
100x100	0.993346	0.998868	0.989283
200x200	0.995951	0.999210	0.995539
500x500	0.996786	0.999866	0.996290
1000x1000	0.998522	0.999633	0.999261

Table 6.1: The effect of image size on the performance of the evaluation using the ground truth.

5.2.5, the partitioning is accomplished pixel-wise using the binning method.

With a low resolution/image size, more and more cells fall into one pixel, which in turn worsens the performance of the algorithm. In fact, cancer and non-cancer cells cannot be separated if they fall into one pixel. In order to show this effect, we used the original input images with the corresponding segmentation masks (ground truth, labelled by the medical experts) with different image sizes and evaluated the performance based on median F1-score, median precision, and median recall (see Table 6.1).

Although the F1-score is not 1, as would be desirable, the errors made in pixel-wise evaluation via binning are negligible with an appropriate image size. The F1-score is above 0.99 for image sizes above 100x100, as shown in Table 6.1. In terms of execution time and complexity, it is desirable to use a small image size. Therefore, we choose the smallest image size for which precision and recall are above 0.99, i.e. 200x200.

After examining the results, we can see that most of the misclassified cells lie on the convex hull (see Figure 6.1). Hence, we added padding around the segmentation mask (one pixel), which improved the performance marginally. It increased the recall to 1 but reduced the precision slightly to 0.9990. In the automatic detection of cancer cells, both measures, precision and recall, are relevant, with recall being marginally more important. Thus, the algorithm detects more cancer cells than present in the ground truth, which is preferable compared to detecting fewer than are actually present. Therefore, we used the padding of one pixel.

6.1.2 Hierarchical gating as consecutive classification tasks

As introduced in Section 2.3, in the manual gating procedure medical experts track down cancer cells hierarchically by drawing polygons around different cell populations, where in each step all events outside the gate are considered as irrelevant background events and are thus discarded. Hence, the scatter plot used in the next step shows only a subset of events, allowing for more accurate positioning of the next gate. Finally, the target population consists of all remaining events after the last gate which correspond to the events passing through all gates of the hierarchy (Boolean AND-operation). In each step of the gating hierarchy, different scatter plots with different specific markers (e.g.,

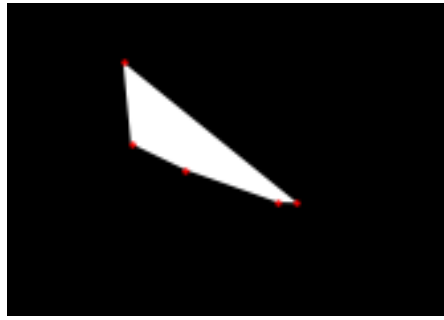


Figure 6.1: Sample with a low amount of MRD, containing only 5 cancer cells (vie14 dataset). The red dots are the cancer cells and the white area is the ground truth segmentation mask. Due to the low number of cancer cells, they all lie on the edge of the convex hull.

CD19 vs. SSC-A for finding mature B-cells) are used. In our automated procedure, this corresponds to finding decision regions in each of these scatter plots, i.e., predicting the segmentation mask of each scatter plot consecutively, where each plot shows only events that were not filtered out in the previous step. Since the gates of a particular step (e.g., gates in the CD19 vs. SSC-A scatter plot) have their own characteristic location, shape, and size variations, we train and evaluate each step separately. As input, the scatter plots with all events removed by the manually drawn gates of previous steps are used. As shown in Figure 2.3, cancer cells and non-cancer cells are more easily separable in some steps of the gating procedure than in others.

To compare the different gates, the vie dataset was used.

The models were trained separately for each gate and we used the ground truth, i.e. the remaining cells of the previous gates selected by medical experts, as input data. This allowed for a valid comparison of the individual gates and avoided the effects of error propagation from one gate to the next. In other words, previous errors are ignored and we tested each gate as if it was a separate instance. This section provides an overview of potentially problematic steps in the gating procedure and how these gates affect the overall performance of the proposed method.

The results of the first four gates are similarly good (see Figure 6.2) based on the scoring metrics used, F1-score, precision and recall (see Section 5.4.1 for more details). For the first four gates, the problem of overlap between cancer and non-cancer cells does not exist, thus, the performance is quite balanced between the three evaluation metrics mentioned. The last gate is more complex; in this case, the samples overlap, i.e. three different images must be taken and the intersection of these selected cells is considered as cancer cells in the evaluation. The main challenge with this gate is the small number of cancer cells in some samples (see figure 6.3). As the number of cells in the selected subsamples decreases from gate to gate, the problem of unbalanced precision and recall worsen in the last gate.

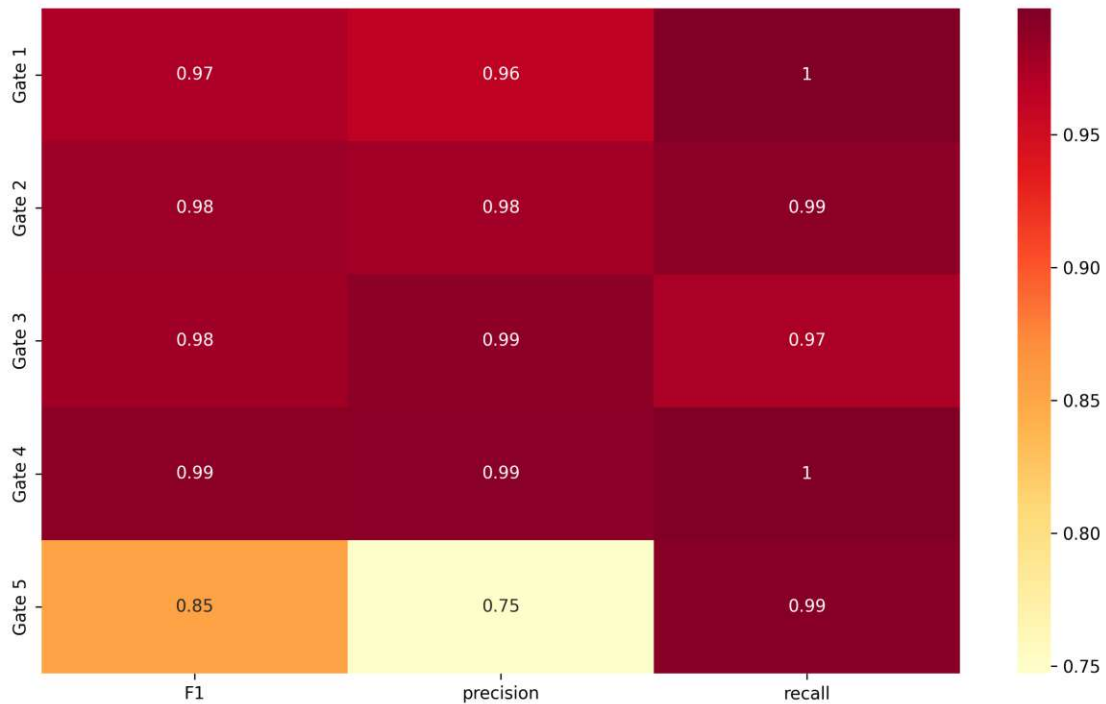


Figure 6.2: Gate-wise evaluation using the [vie](#) dataset. The columns show the median values of the performance metrics over the entire test set. The gradient colours span from the minimum to the maximum values.

In gate 5, recall is almost 1, which means that all cancer cells were found. However, the mean precision is rather low (0.75), suggesting that we had identified more cells as cancer cells than the ground truth. In the case of individualised therapy, this may lead to a problem, as it would mean that patients receive higher doses of radiation than they should.

This gatewise evaluation has only an exploratory advantage, given that the results based on the hierarchical approach are not equal to those presented (see Chapter [6.2](#)). The high median value of recall (see Figure [6.2](#)) could be advantageous for hierarchical sub-selection, as this means that we always keep more cells than we should. If too many cells were discarded in the early stages of gating, we might also drop cancer cells at earlier stages. This effect worsens the error propagation through the five gates.

6.1.3 Evaluation of the Conditioned U-Net model

We propose two alternative ways to model the hierarchical sequence of gating: either using separately trained models (one for each gating step) or training a single model for all steps, using a conditional U-Net model introduced in Section [4.8.2](#).

To address the question of whether we need separate models for each gate or can use a single model, both approaches were implemented and evaluated using the same state-of-

Datasets		Separated U-Net models			Conditioned U-Net model		
train set	test set	median F1-score	median precision	median recall	median F1-score	median precision	median recall
vie	vie	0.92	0.91	0.96	0.93	0.93	0.98
bln	bue	0.69	0.65	0.95	0.93	0.95	0.99
bln	vie14	0.36	0.70	0.37	0.87	0.88	0.96
bln	vie20	0.85	0.91	0.95	0.42	0.35	0.97
bue	bln	0.40	0.26	0.98	0.52	0.49	0.91
bue	vie14	0.35	0.97	0.21	0.78	0.69	0.97
bue	vie20	0.65	0.62	0.92	0.79	0.68	1.00
vie14	bln	0.84	0.86	0.98	0.94	0.96	0.94
vie14	bue	0.95	0.96	0.98	0.96	0.96	0.99
vie14	vie20	0.91	0.91	0.98	0.85	0.90	0.94
vie20	bln	0.87	0.87	0.95	0.79	0.72	0.93
vie20	bue	0.78	0.71	0.99	0.94	0.95	0.98
vie20	vie14	0.87	0.78	0.99	0.95	0.97	0.97

Table 6.2: Comparison of the different approaches: training different models for each gate and a conditioned model for all gates at once

the-art models as Wödinger et al. [WRW⁺22]. The conditional U-Net model has the additional advantages of being sparse, easier to use and we do not need to store multiple models for a given evaluation scenario. The results and comparison of the experiments can be seen in Table 6.2. The numbers in bold indicate which model performed better in which experiment.

Table 6.2 shows that the U-Net models perform worse when trained with the small Berlin and Buenos Aires datasets. This result is consistent with a recent paper by Kowarsch et al. [KWW⁺22]. Based on this study, this could be caused not only by overfitting, but also by the shape of the polygons (target masks) predicted for the smaller datasets, which differ from the polygons in the Vienna data for Gate 5 (while the Vienna dataset contains horizontal and vertical polygons for Gate 5, the datasets collected in Buenos Aires and Berlin contain only vertical ones). Therefore, the models trained with the smaller datasets do not learn to predict horizontal polygons.

Based on the results in Table 6.2 we can see a gap in performance between the approaches. The conditioned U-Net model outperformed the approach with the separated models in 10 out of 13 cases. Therefore the conditioned U-Net model will be compared with the state-of-the-art methods in Section 6.2. As shown in Table 6.2, the performance of the different experiments based on the median F1-score, median precision, and median recall varies greatly. This can be the result of numerous factors such as the size of the training dataset or the number of cancer cells per sample. The latter will be explored further in the next section.

6.1.4 Evaluation based on the MRD

In this section, we relate the performance of our model to the amount of MRD. The MRD varies significantly across different samples. Figure 6.3 shows the statistics of the MRD contained by the samples of the datasets introduced in Section 2.2.

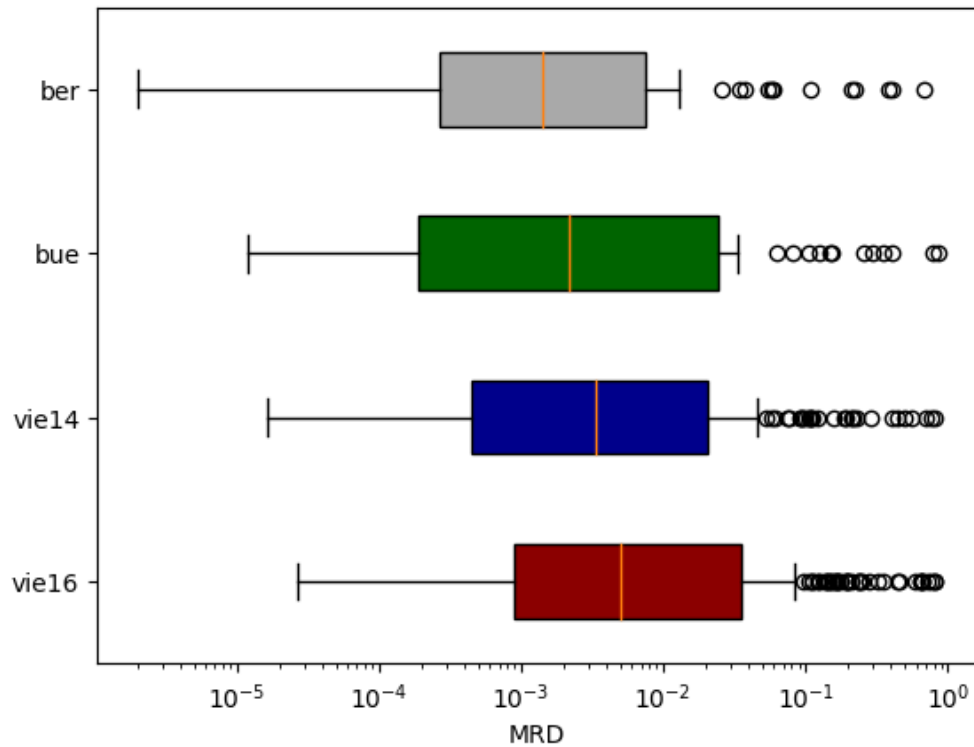


Figure 6.3: MRD contained in the bone marrow samples of the four datasets. On the x-axis, the MRD in absolute number of cancer cells on a logarithmic scale can be seen. The samples that do not contain cancer cells were discarded beforehand.

The variation of MRD in the different datasets is shown in Figure 6.3. The range of the proportion of cancer cells varies between 1.205×10^{-5} and 0.847, meaning there is a wide variation between the different samples in terms of the amount of MRD. The datasets collected in Vienna contain on average a higher median of cancer cells per sample, see 6.3). The minimum amount of cancer cells in a sample is also different in the datasets. The samples gathered in Berlin had a lower minimum MRD compared to the others. In this section, the vie dataset will be used.

Imbalanced designs can make classification problems challenging [EdRCH17] and a lower number of cancer cells can be harder to detect. In order to examine whether the MRD% of a sample has an influence on the results, the F1-score of the samples of the vie dataset will be visualised based on the number of cancer cells (see Figure 6.4).

The performance of the estimations of the model based on the median F1-score strongly depends on the MRD in the samples (see Figure 6.4).

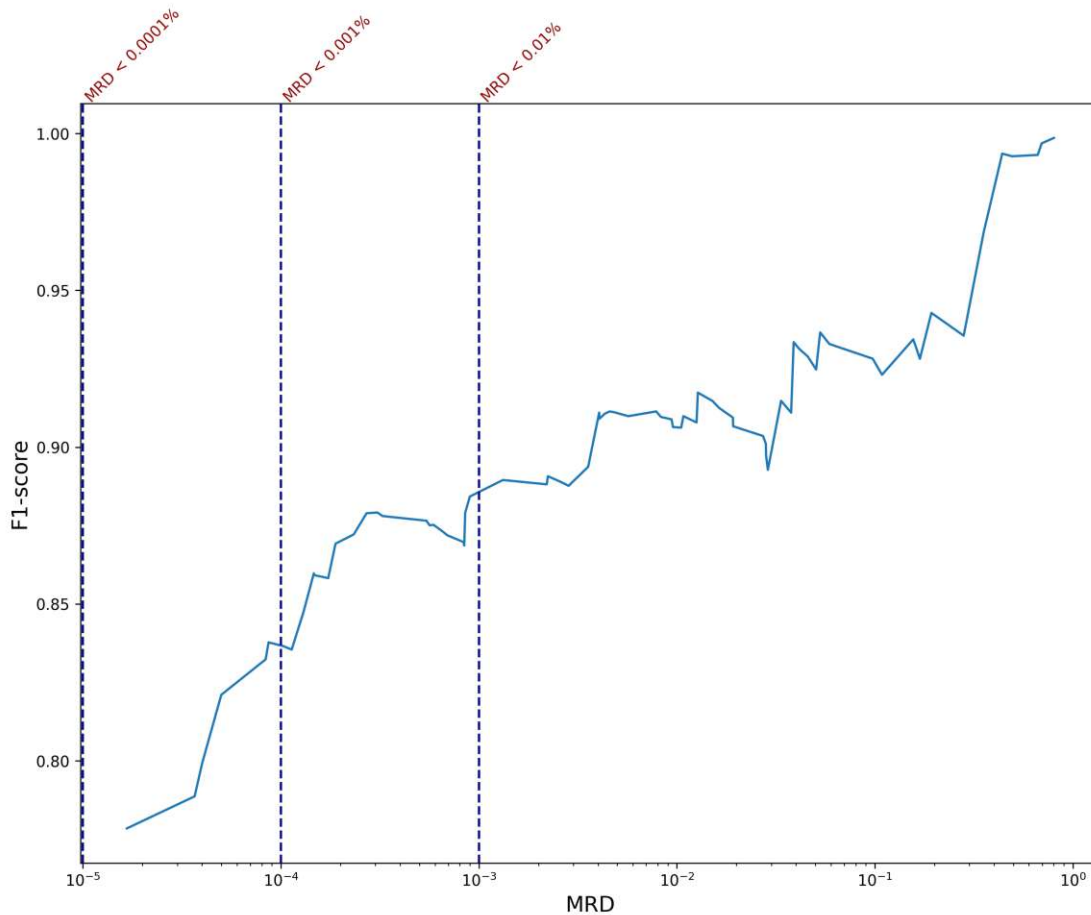


Figure 6.4: Evaluation based on the proportion of cancer cells contained by the samples of the dataset gathered in Vienna. On the x-axis, the number of cancer cells contained in each sample is represented on a logarithmic scale. On the y-axis, the corresponding F1-score is represented. The blue line indicates the average F1-score for all samples having a **MRD** greater or equal to x .

The amount of cancer cells per sample varies strongly. The cancer cells can be detected more accurately in samples with a larger amount of cancer cells. Based on Figure 6.4 the samples, which contain at least 0.001 % **MRD** have an average F1-score above 0.8.

Figure 6.5 shows the concordance between predicted and true **MRD**. The samples, which lie on the straight line, are classified correctly, the samples outside the dashed lines are not acceptable predictions according to [DGR⁺08]. This problem occurs more frequently in cases with a lower **MRD**. The samples in which most cells were misclassified are found on the upper left side of the image. This indicates that we tend to predict more cells

than are actually present in the ground truth (this phenomenon was also indicated by the high recall beforehand).

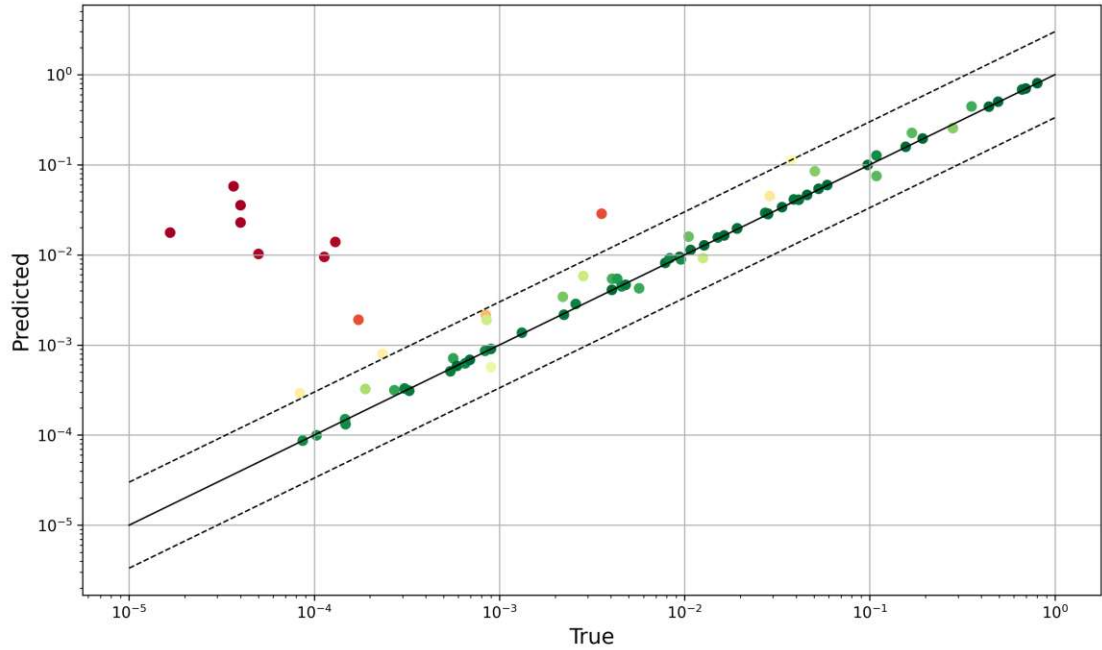


Figure 6.5: Percent of the ground truth (x-axis) and predicted cancer cells (y-axis) on a logarithmic scale for each sample of the `vie` dataset. According to Dworzak et al. [DGR⁺08], a prediction can be considered accurate if the amount of detected cancer cells lies within a range of 1/3 and 3 times of the `MRD`. The colour of the points indicates the F1-score reached after the evaluation of the test set (between green and red, where red indicates an F1-score of 0 and green F1-score of 1).

In order to see the differences between the predictions and the ground truth labelled by medical experts, we ordered the samples by ascending F1-scores. For the visualisation in Figure 6.6, we used an image with two different scales: the scale on the left shows the F1-score of the samples and the scale on the right the `MRD` in the samples on a logarithmic scale. The dependence between the number of cancer cells and the errors made by the proposed method are visible. The higher the `MRD` in a given sample, the better the predictions based on F1-score.

6.2 Overall evaluation

In this section, the results of the models trained on the datasets collected in Vienna, Berlin, and Buenos Aires are compared to the best results of the state-of-the-art literature by Wödinger et al. [WRW⁺22] and Reiter et al. [RDS⁺19].

In the first experiment (`vie`), the two datasets gathered in Vienna (`vie14` and `vie20`) are

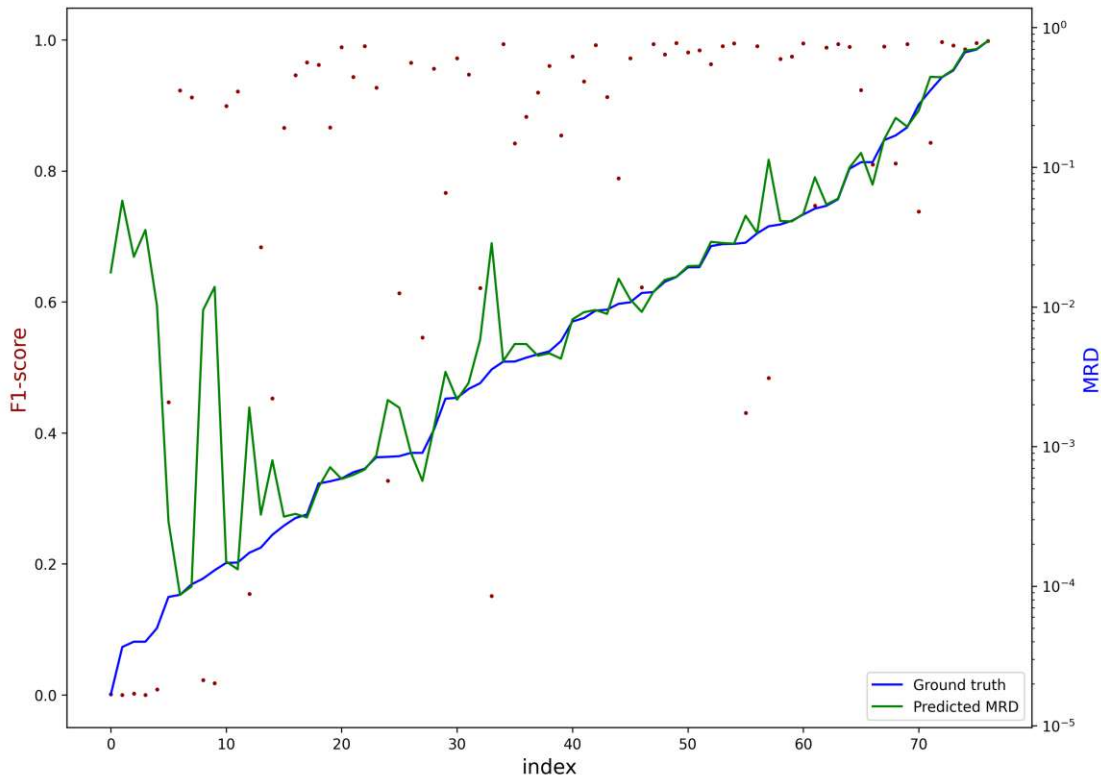


Figure 6.6: The relationship between the number of cancer cells in a sample, represented on a logarithmic scale, and the F1-score achieved by the U-Net model using the samples collected in Vienna. The blue line corresponds to the number of **MRD**, the green line to the predicted amount of **MRD** and the red dots represent the F1-score of each sample.

mixed and randomly split into a train (70%), validation (15%), and test set (15%). In every other experiment, one dataset is used for training and another one for testing, see Table 6.3. The U-Net model used for the overall evaluation is a conditioned U-Net model based on Section 6.1.3. For each train and test set combination, we used the remaining two datasets as validation sets.

In Table 6.3 the results of the evaluation can be seen. The results indicate that the performance of the Transformer model and the U-Net approach are similar. Error propagation may be responsible for most cases, where the state-of-the-art significantly outperforms the proposed method. In numerous examples, Gate 4 detects far more cells than there are in the ground truth (low precision and high recall), so the following images appear quite unusual and/or do not contain all the cancer cells.

Our main goal is not to outperform the state-of-the-art method but to meet the same performance with the additional advantage of explainability gained through visual representation. In order to test, whether there is a significant difference between the performance of the two approaches, we used a two-sided Wilcoxon sign-rank test.

train set	test set	med F1-score U-Net	med precision	med recall	med F1-score Transformer	med F1-score GMM
vie	vie	0.93	0.93	0.98	0.94	-
bln	bue	0.93	0.95	0.99	0.87	0.68
bln	vie14	0.87	0.88	0.96	0.9	0.35
bln	vie20	0.42	0.35	0.97	0.87	0.48
bue	bln	0.52	0.49	0.91	0.77	0.5
bue	vie14	0.78	0.69	0.97	0.9	0.84
bue	vie20	0.79	0.68	1.00	0.88	0.86
vie14	bln	0.94	0.96	0.94	0.9	0.81
vie14	bue	0.96	0.96	0.99	0.95	0.84
vie14	vie20	0.85	0.9	0.94	0.89	0.86
vie20	bln	0.79	0.72	0.93	0.81	0.25
vie20	bue	0.94	0.95	0.98	0.86	0.81
vie20	vie14	0.95	0.97	0.97	0.86	0.89

Table 6.3: Results of the evaluation:

The results are compared with the results of Wödingner et al. [WRW⁺22] and with the results of Reiter et al., [RDS⁺19] based on the median precision, median recall and the median F1-scores of the models evaluated on the test sets. Bold numbers highlight the highest performance achieved using a specific training and test set.

The Wilcoxon Signed-Rank test is a non-parametric test which is used when dealing with dependent samples. The assumption of the normal distribution required to use a paired t-test is not fulfilled and the sample size (in this case, the number of experiments, 13) is too marginal to rely on asymptotic results, hence we use a non-parametric version of the paired t-test [WB02].

Null hypothesis H_0 :

The median F1-scores of the U-Net models - median F1-scores of the Transformer models [WRW⁺22] are symmetric around $\mu = 0$.

Two-sided alternative hypothesis H_1

The median F1-scores of the U-Net models - median F1-scores of the Transformer models are symmetric around $\mu \neq 0$.

The test returns a p-value of 0.861, which indicates that the null hypotheses cannot be rejected. This means that there is no significant difference between the results of the two approaches. In other words, the results obtained by using the proposed method meet the performance of the state-of-the-art methods. However, there are some cases, e.g. the data from Buenos Aires, where the state-of-the-art methods performed significantly better.

Conclusion

In this chapter, this thesis is summarised, the research questions described in chapter 1 are answered, the limitations of the proposed algorithm are indicated and the future work is addressed. In this thesis, a new approach for the automated detection of cancer cells was presented using an image segmentation model, the U-Net architecture. As shown in Section 6.2, the proposed method aligns with the performance of the state-of-the-art methods.

7.1 Conclusion of research questions

The main goal of this thesis is to detect cancer cells automatically based on a sequence of 2D images of flow cytometry data using an image segmentation method, namely, the U-Net architecture. To achieve this, it has been essential to generate the input images with the most appropriate size and resolution, to implement a hierarchical sub-selection of cancer cell populations, and to compare the results with outcomes in the current literature. The main focus of this thesis lies on the following points:

What is the optimal U-Net architecture in terms of layers and kernel parameters for automating the gating procedure in flow cytometry data?

As described in Table 5.2, the best performing model architecture for the automation of the gating procedure contains 5 layers, a 3x3 convolutional kernel and a reduced number of feature channels compared to the original model developed by Ronnenberger et al. [RFB15].

Alongside the selection of the most appropriate algorithm for the automation of the gating procedure, the significance of the data preparation should not be underestimated, since it can significantly enhance the algorithm's performance. This entails tasks such as normalizing and visualizing the samples (see Section 5.1.1). The input data is

visualised using a scatter density plot as described in Section 5.1.2. For each scatterplot, a segmentation mask (target mask) is generated (see Section 5.1.3), which can be considered as the ground truth. As introduced in Section 5.2.5, cancer cells are tracked down hierarchically using the predicted segmentation masks. The cells, which lie outside the predicted segmentation mask are discarded and not used for the next step of the hierarchical gating procedure.

To investigate the question as to whether we need to train different models for the different phases of the gating procedure, we trained and evaluated two different approaches: the U-Net architecture trained separately for each gate and a conditioned U-Net model that can be applied to every gate and which uses a one-hot encoded vector as additional information about the source of the data. In this case, the source indicates the gate number. As shown in Section 6.1.3, the conditioned U-Net model significantly outperformed the approach with the separated models.

The trained models tend to detect more cancer cells than the medical experts (higher recall and lower precision). This scenario might be preferable because failing to detect cancer cells could result in the patient not receiving treatment.

What distinct advantages does the proposed method offer in comparison to state-of-the-art techniques for identifying cancer cell populations?

The main advantage of the proposed method is the ease of understanding the automated gating procedure due to the visual representation based on the input images and the corresponding segmentation masks.

The effort required for data preparation, such as cleaning as well as normalising the samples and creating the input images and masks, can be seen as a disadvantage seeing as it is very time-consuming and prone to errors. In comparison, state-of-the-art methods which work with multidimensional data can be applied directly to the samples.

Based on Section 6.1.4, we can say that the performance of the proposed method depends on the number of cancer cells contained in the sample; it performs better in samples with a higher MRD (see Figure 6.5). As the implementation is based on a hierarchical sub-selection of cancer cell populations, it is vital to determine at which stages of the hierarchy, in which gate, we make errors. If errors occur at early stages, the error propagation could be so high that this approach would then be ineffective. Fortunately, the first gates work very well, the greatest challenge is the last gate which consists of three images, and the overlaps of the detected cells need to be considered (see Figure 6.2).

In order to investigate the question as to whether the results of the proposed method differ significantly from the baseline method, a Wilcoxon signed-rank test was applied. Based on the results shown in Section 6.2, the performance of the proposed method meets the results of the state-of-the-art methods.

7.2 Limitations

Three limitations of this work that we would like to highlight are the amount of training data, the hierarchical sub-selection of cell-populations, and the influence of the number of remaining cancer cells in the samples.

The availability of data, especially labelled data is generally a significant challenge for biomedical studies [GDDM19]. The low amount of training data used for this thesis can be considered as one of the biggest limitations. The total amount of data used for this work is almost 650 samples. We attempted to solve this issue with the help of data augmentation methods, but a training size of fewer than 100 data may not be enough to train a robust model.

Finding the most appropriate method for partitioning the input data using the predicted segmentation masks in order to reproduce the hierarchical gating procedure was among the most challenging aspects of this thesis. The chosen approach, pixel-wise binning, makes small errors even when evaluating the ground truth. Each sample of the datasets contains a very high amount of cells (approximately 300.000), which was visualised on a scatter-density plot of size 500x500 or lower. Therefore, many cells fall within one pixel of the image. When partitioning the original data using the predicted segmentation masks, the algorithm makes pixel-by-pixel decisions. This means that all data points that fall within a pixel are retained or discarded for the next step in the gating hierarchy. In the case that both cancer cells and non-cancer cells are found within one pixel, the cells in one of these groups will be misclassified. The evaluation on the ground truth has the highest F1-score of 0.99, which gives an upper bound for the performance of the proposed method.

The main goal of this thesis was to automate the gating procedure with scatterplots as input using the U-Net architecture. For this purpose, we used a sequence of 2-dimensional images. Through this sequential approach, error propagation becomes a significant factor, which highly influences the performance of the algorithm. False negatives in previous gates can have a significant impact on the output of the following gate. This is a major disadvantage of this method when compared to the methods used in state-of-the-art literature which work with multidimensional data simultaneously.

The amount of MRD in a sample highly affects the performance of the algorithm (see Figure 6.4). In cases of low MRD, the convex hull was constructed around the cancer cells (see Figure 6.1). Such cases are major challenges given that the border pixels are very hard to learn for an image segmentation model. Since we generated the segmentation masks as a convex hull around the cancer cells labelled by medical experts, we are not able to draw a convex hull if there are fewer than 3 cells marked in a sample since it is the smallest convex set of cells [BDH96].

Although our proposed approach aims to achieve a wider acceptance among medical professionals due to the visual representation and the reproduction of the gating procedure, it lacks an explanation of why certain cells are included in one gate and others are not.

There is no model and algorithm which is perfectly suited for solving this problem. There are many edge cases where other algorithms would achieve even better results.

The above limitations could lead to further research directions for automated cancer cell detection using the U-Net architecture. Additionally, experiments could be conducted with other versions of the U-Net architecture which might perform better, such as U-Net++ [ZRSTL18], Residual U-Net [ZLW18] and Attention U-Net [OSF⁺18].

7.3 Contributions and Future Work

The experiments conducted in this thesis show that the proposed method is well suited for the detection of cancer cell populations using bone marrow samples from the patients 15 days after chemotherapy. The proposed method achieves a high F1-score (above 0.85) in most cases, especially for models trained with the Vienna datasets; the performance is in line with the results of state-of-the-art methods. In this scenario, the most significant benefit stems from the number of samples available and from the shapes of the polygons in the last gates, since the Vienna datasets contain both horizontal and vertical polygons, but the other datasets contain only vertical ones.

For future work, increasing the number of annotated samples can open many doors of opportunity.

At this point, the automated generation of the input images as segmentation masks, the U-Net architecture, and the evaluation on the test set were implemented in this thesis. Nevertheless, this framework might not be easily accessible to medical professionals who lack proficiency in the Python programming language. In order to allow doctors easier usability, an application could be developed where new datasets (test sets) could be loaded and, based on the pre-trained models, the detected cell populations of the various gating steps could be visualised.

List of Figures

2.1	Disease burden based on number of MRD ch913	8
2.2	Forward Scatter and Side Scatter Light ch213	9
2.3	Manual gating procedure RRK⁺16	11
2.4	Visualisation of the last steps of the manual gating procedure. Image a shows the 4th gate while images b, c, and d are parts of the 5th gate. The images are two-dimensional projections of a sample JS12 gathered in Vienna. The cells within the red dashed polygons are the cells, labelled by medical experts.	
	12
4.1	A simple feed forward network with one hidden layer containing two units GBC16	19
4.2	Visual representation of a 2D convolution GBC16	21
4.3	Visual representation of a convolutional layer FLL⁺17	22
4.4	Convolution operation using kernel size 3×3, no padding, and stride 1 YNDT18	23
4.5	Fully Connected layers WBAK20	23
4.6	Illustration of the need for early stopping. The negative log-likelihood loss was visualised for the training and a validation set GBC16	26
4.7	U-Net architecture RFB15	30
4.8	Comparison of a convolution and transposed convolution GDDM19	30
4.9	2x2 dimensional example for the upsampling convolution, or with other words, transposed convolution ZLLS21	31
4.10	Max Pooling and Max Unpooling GDDM19	31
5.1	The implementation workflow	34
5.2	Implementation of the hierarchical approach for the automated gating procedure.	35
5.3	Example of an input image of the Vienna 14 dataset. The image belongs to Gate 4; the x-axis represents the SSC-A and the y-axis the FSC-A of the given cells.	36
5.4	Example of a segmentation mask of the Vienna 14 dataset (it belongs to the input image in Figure 5.3). The image belongs to Gate 4; the x-axis represents the SSC-A and the y-axis the FSC-A of the given cells.	37

5.5	Raw predictions for a segmentation mask in the continuous interval from 0 to 1.	37
5.6	Training of the U-Net model.	38
5.7	Modified version of the U-Net architecture proposed by Ronnenberger et al. RFB15	39
5.8	The input images and the predicted segmentation mask (white area) for three different epochs using the validation set of the vie dataset.	43
5.9	Imitation of the partitioning of the input data based on the predicted segmentation mask. The cells which lie on a black pixel on the right visualisation will be rejected for the next gate. The blue marked cells will be selected and used for the next gate. This example uses a 40x40 grid for visualisation purposes, in the experiment 200x200 pixels and grids will be used.	44
5.10	Training, validation and test data split YNDT18	45
6.1	Sample with a low amount of MRD , containing only 5 cancer cells (vie14 dataset). The red dots are the cancer cells and the white area is the ground truth segmentation mask. Due to the low number of cancer cells, they all lie on the edge of the convex hull.	49
6.2	Gate-wise evaluation using the vie dataset. The columns show the median values of the performance metrics over the entire test set. The gradient colours span from the minimum to the maximum values.	50
6.3	MRD contained in the bone marrow samples of the four datasets. On the x-axis, the MRD in absolute number of cancer cells on a logarithmic scale can be seen. The samples that do not contain cancer cells were discarded beforehand.	52
6.4	Evaluation based on the proportion of cancer cells contained by the samples of the dataset gathered in Vienna. On the x-axis, the number of cancer cells contained in each sample is represented on a logarithmic scale. On the y-axis, the corresponding F1-score is represented. The blue line indicates the average F1-score for all samples having a MRD greater or equal to x.	53
6.5	Percent of the ground truth (x-axis) and predicted cancer cells (y-axis) on a logarithmic scale for each sample of the vie dataset. According to Dworzak et al. DGR⁺08 , a prediction can be considered accurate if the amount of detected cancer cells lies within a range of 1/3 and 3 times of the MRD . The colour of the points indicates the F1-score reached after the evaluation of the test set (between green and red, where red indicates an F1-score of 0 and green F1-score of 1).	54
6.6	The relationship between the number of cancer cells in a sample, represented on a logarithmic scale, and the F1-score achieved by the U-Net model using the samples collected in Vienna. The blue line corresponds to the number of MRD , the green line to the predicted amount of MRD and the red dots represent the F1-score of each sample.	55

List of Tables

3.1	Results of the state-of-the-art literature [WRW ⁺ 22]	15
5.1	Features and gate labels used for the creation of input images in the various gating level.	35
5.2	Grid search for the best fitting model structure using the vie dataset.	40
5.3	Values used for the hyperparameter tuning.	42
5.4	Experiments conducted in this thesis.	46
5.5	Confusion matrix [HS15]	46
6.1	The effect of image size on the performance of the evaluation using the ground truth.	48
6.2	Comparison of the different approaches: training different models for each gate and a conditioned model for all gates at once	51
6.3	Results of the evaluation: The results are compared with the results of Wödinger et al. [WRW ⁺ 22] and with the results of Reiter et al., [RDS ⁺ 19] based on the median precision, median recall and the median F1-scores of the models evaluated on the test sets. Bold numbers highlight the highest performance achieved using a specific training and test set.	56

Acronyms

ALL Acute Lymphoblastic Leukemia. [7](#)

bln Berlin dataset. [37](#), [41](#)

bue Buenos Aires dataset. [37](#), [41](#)

CNN Convolutional Neural Network. [3](#), [20](#), [23](#), [25](#), [28](#), [29](#)

GMM Gaussian Mixture Models. [14](#), [56](#)

MRD Minimal Residual Disease: Number of remaining cancer cells. [xiii](#), [1](#), [4](#), [7-9](#), [16](#),
[47](#), [49](#), [52-55](#), [58](#), [61](#), [62](#)

vie Vienna dataset. [40](#), [44](#), [49](#), [50](#), [52](#), [54](#), [62](#), [63](#)

vie14 Vienna dataset from 2014. [10](#), [37](#), [44](#), [49](#), [54](#), [62](#)

vie20 Vienna dataset from 2016-2020. [10](#), [37](#), [44](#), [54](#)

Bibliography

- [AMAZ17] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.
- [ANHB11] Nima Aghaeepour, Radina Nikolic, Holger H Hoos, and Ryan R Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79(1):6–13, 2011.
- [AZH⁺21] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1):1–74, 2021.
- [Bai17] Barbara J Bain. *Leukaemia diagnosis*. John Wiley & Sons, 2017.
- [BDH96] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.
- [BDVDGS⁺18] Paul Blanc-Durand, Axel Van Der Gucht, Niklaus Schaefer, Emmanuel Itti, and John O Prior. Automatic lesion detection and segmentation of 18f-fet pet in gliomas: a full 3d u-net convolutional neural network study. *PLoS One*, 13(4):e0195798, 2018.
- [Ben12] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.
- [BIK⁺20] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020.
- [BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

- [BN06] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [BVV⁺09] Giuseppe Basso, Marinella Veltroni, Maria Grazia Valsecchi, Michael N Dworzak, Richard Ratei, Daniela Silvestri, Alessandra Benetello, Barbara Buldini, Oscar Maglia, Giuseppe Masera, et al. Risk of relapse of childhood acute lymphoblastic leukemia is predicted by flow cytometric measurement of residual disease on day 15 bone marrow. *Journal of Clinical Oncology*, 27(31):5168–5174, 2009.
- [BW00] Michael Brown and Carl Wittwer. Flow cytometry: principles and clinical applications in hematology. *Clinical chemistry*, 46(8):1221–1229, 2000.
- [Cam09] Dario Campana. Minimal residual disease in acute lymphoblastic leukemia. In *Seminars in hematology*, volume 46, pages 100–106. Elsevier, 2009.
- [ch213] *Principles of Flow Cytometry*, chapter 2, pages 3–19. John Wiley Sons, Ltd, 2013.
- [ch913] *Minimal Residual Disease*, chapter 9, pages 184–201. John Wiley Sons, Ltd, 2013.
- [CMV⁺21] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, 2021.
- [CRU] Cancer research uk. <https://www.cancerresearchuk.org/funding-for-researchers/our-research-strategy>. Accessed: 2023-01-01.
- [CvdWTBS⁺98] Hélène Cavé, Jutte van der Werff Ten Bosch, Stefan Suci, Christine Guidal, Christine Waterkeyn, Jacques Otten, Marleen Bakkus, Kris Thielemans, Bernard Grandchamp, Etienne Vilmer, et al. Clinical significance of minimal residual disease in childhood acute lymphoblastic leukemia. *New England Journal of Medicine*, 339(9):591–598, 1998.
- [DGR⁺08] Michael Norbert Dworzak, Giuseppe Gaipa, Richard Ratei, Marinella Veltroni, Angela Schumich, Oscar Maglia, Leonid Karawajew, Alessandra Benetello, Ulrike Pötschger, Zvenyslava Husak, et al. Standardization of flow cytometric minimal residual disease evaluation in acute lymphoblastic leukemia: Multicentric assessment is feasible. *Cytometry Part B: Clinical Cytometry: The Journal of the International Society for Analytical Cytology*, 74(6):331–340, 2008.

- [DWWK] Markus Diem, Lisa Weijler, Matthias Woedlinger, and Florian Kowarsch. Flowmepy - a python api for flowme. <https://pypi.org/project/flowmepy/>. (accessed: 2023-03-28).
- [FBBG09] G Finak, A Bashasharti, R Brinkmann, and R Gottardo. Merging mixture model components for improved cell population identification in high throughput flow cytometry data. *Advances in Bioinformatics*, 100, 2009.
- [FdRCH17] Alberto Fernández, Sara del Río, Nitesh V Chawla, and Francisco Herrera. An insight into imbalanced big data classification: outcomes and challenges. *Complex & Intelligent Systems*, 3(2):105–120, 2017.
- [FLL⁺17] Jian Feng, Fangming Li, Senxiang Lu, Jinhai Liu, and Dazhong Ma. Injurious or noninjurious defect identification from mfl images in pipeline inspection using convolutional neural network. *IEEE Transactions on Instrumentation and Measurement*, 66(7):1883–1892, 2017.
- [FM82] Kunihiro Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [GDDM19] Swarnendu Ghosh, Nibaran Das, Ishita Das, and Ujjwal Maulik. Understanding deep learning techniques for image segmentation. *ACM Computing Surveys (CSUR)*, 52(4):1–35, 2019.
- [Giv01] Alice L Givan. Principles of flow cytometry: an overview. *Methods in cell biology*, 63:19–50, 2001.
- [Giv13] Alice Longobardi Givan. *Flow cytometry: first principles*. John Wiley & Sons, 2013.
- [HD17] Seyed Hossein Hassanpour and Mohammadamin Dehghani. Review of cancer from perspective of molecular. *Journal of Cancer Research and Practice*, 4(4):127–129, 2017.
- [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.
- [HGYR17] Zeshan Hussain, Francisco Gimenez, Darwin Yi, and Daniel Rubin. Differential data augmentation techniques for medical imaging classification tasks. In *AMIA annual symposium proceedings*, volume 2017, page 979. American Medical Informatics Association, 2017.

- [HM22] Nicholas C Herold and Prasenjit Mitra. Immunophenotyping. In *StatPearls [Internet]*. StatPearls Publishing, 2022.
- [HS15] Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.
- [HSLB12] Dieter K Hossfeld, Charles D Sherman, Richard R Love, and FX Bosch. *Manual of clinical oncology*. Springer Science & Business Media, 2012.
- [HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [Hun07] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.
- [HWC⁺22] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2022.
- [IGMA18] Sakshi Indolia, Anil Kumar Goswami, Surya Prakesh Mishra, and Pooja Asopa. Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science*, 132:679–688, 2018.
- [Jad20] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.
- [JDV⁺17] Simon Jégou, Michal Drozdal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19, 2017.
- [JS12] Chad Rosenberg Nikesh Kotecha Ryan R. Brinkman Josef Spidlen, Karin Breuer. Flowrepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. *Cytometry Part A (Journal of Quantitative Cell Science)*, 2012.
- [JWF16] Kerstin Johnsson, Jonas Wallin, and Magnus Fontes. Bayesflow: latent modeling of flow cytometry cell populations. *BMC bioinformatics*, 17(1):1–16, 2016.

- [KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [KKR14] M Jogendra Kumar, Dr GVS Raj Kumar, and R Vijay Kumar Reddy. Review on image segmentation techniques. *International Journal of Scientific Research Engineering & Technology (IJSRET)*, 3(6), 2014.
- [KWG⁺18] Viksit Kumar, Jeremy M Webb, Adriana Gregory, Max Denis, Duane D Meixner, Mahdi Bayat, Dana H Whaley, Mostafa Fatemi, and Azra Alizad. Automated and real-time segmentation of suspicious breast masses using convolutional neural network. *PloS one*, 13(5):e0195816, 2018.
- [KWW⁺22] Florian Kowarsch, Lisa Weijler, Matthias Wödlinger, Michael Reiter, Margarita Maurer-Granofszky, Angela Schumich, Elisa O Sajaroff, Stefanie Groeneveld-Krentz, Jorge G Rossi, Leonid Karawajew, et al. Towards self-explainable transformers for cell classification in flow cytometry data. In *International Workshop on Interpretability of Machine Intelligence in Medical Image Computing*, pages 22–32. Springer, 2022.
- [LBG08] Kenneth Lo, Ryan Remy Brinkman, and Raphael Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 73(4):321–332, 2008.
- [LGV⁺21] Andreanne Lemay, Charley Gros, Olivier Vincent, Yaou Liu, Joseph Paul Cohen, and Julien Cohen-Adad. Benefits of linear conditioning for segmentation using metadata. In *Medical Imaging with Deep Learning*, pages 416–430. PMLR, 2021.
- [LRD⁺18] Roxane Licandro, Michael Reiter, Markus Diem, Michael Dworzak, Angela Schumich, and Martin Kampel. Application of machine learning for automatic mrd assessment in paediatric acute myeloid leukaemia. *Cancer cells*, 1012:1010, 2018.
- [LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [MBP19] Gabriel Meseguer-Brocal and Geoffroy Peeters. Conditioned-u-net: Introducing a control mechanism in the u-net for multiple source separations. *arXiv preprint arXiv:1907.01277*, 2019.
- [MBP⁺21] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using

- deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.
- [MCN⁺21] Jun Ma, Jianan Chen, Matthew Ng, Rui Huang, Yu Li, Chen Li, Xiaoping Yang, and Anne L Martel. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71:102035, 2021.
- [MG18] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.
- [MP69] Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 479:480, 1969.
- [mpl20] Mpl scatter density plots. <https://github.com/astrofrog/mpl-scatter-density>, 2020.
- [Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [OSF⁺18] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.
- [PD13] Dinesh D Patil and Sonal G Deore. Medical image segmentation: a review. *International Journal of Computer Science and Mobile Computing*, 2(1):22–27, 2013.
- [Per21] Aladdin Persson. Machine learning collection. <https://github.com/aladdinpersson/Machine-Learning-Collection>, 2021.
- [PGLVKB12] Julien Picot, Coralie L Guerin, Caroline Le Van Kim, and Chantal M Boulanger. Flow cytometry: retrospective, fundamentals and recent instrumentation. *Cytotechnology*, 64(2):109–130, 2012.
- [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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

- [PRL08] Ching-Hon Pui, Leslie L Robison, and A Thomas Look. Acute lymphoblastic leukaemia. *The Lancet*, 371(9617):1030–1043, 2008.
- [PSDV⁺18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [RDS⁺19] Michael Reiter, Markus Diem, Angela Schumich, Margarita Maurer-Granofszky, Leonid Karawajew, Jorge G Rossi, Richard Ratei, Stefanie Groeneveld-Krentz, Elisa O Sajaroff, Susanne Suhendra, et al. Automated flow cytometric mrd assessment in childhood acute b-lymphoblastic leukemia using supervised machine learning. *Cytometry Part A*, 95(9):966–975, 2019.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
- [Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [RRK⁺16] Michael Reiter, Paolo Rota, Florian Kleber, Markus Diem, Stefanie Groeneveld-Krentz, and Michael Dworzak. Clustering of cell populations in flow cytometry data using a combination of gaussian mixtures. *Pattern Recognition*, 60:1029–1040, 2016.
- [SOvdV⁺01] Tamasz Szczeparski, Alberto Orfão, Vincent HJ van der Valden, Jésus F San Miguel, and Jacques JM van Dongen. Minimal residual disease in leukaemia patients. *The lancet oncology*, 2(7):409–417, 2001.
- [SPED21] Nahian Siddique, Sidike Paheding, Colin P. Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 9:82031–82057, 2021.
- [SS09] Martin Stanulla and Martin Schrappe. Treatment of childhood acute lymphoblastic leukemia. In *Seminars in hematology*, volume 46, pages 52–63. Elsevier, 2009.
- [SSA17] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.

- [SSP⁺03] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. Edinburgh, 2003.
- [TLC⁺18] Guofeng Tong, Yong Li, Huairong Chen, Qingchun Zhang, and Huiying Jiang. Improved u-net network for pulmonary nodules segmentation. *Optik*, 174:460–469, 2018.
- [TPX19] DK Thara, BG PremaSudha, and Fan Xiong. Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques. *Pattern Recognition Letters*, 128:544–550, 2019.
- [VCR⁺16] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 41–48, 2016.
- [VdVBVWVD04] VH Van der Velden, Nancy Boeckx, ER Van Wering, and JJ Van Dongen. Detection of minimal residual disease in acute leukemia. *Journal of biological regulators and homeostatic agents*, 18(2):146–154, 2004.
- [VSP⁺17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.
- [Wan14] Xin Maggie Wang. Advances and issues in flow cytometric detection of immunophenotypic changes and genomic rearrangements in acute pediatric leukemia. *Translational Pediatrics*, 3(2), 2014.
- [WB02] Elise Whitley and Jonathan Ball. Statistics review 6: Nonparametric methods. *Critical care*, 6(6):1–5, 2002.
- [WBAK20] M Arif Wani, Farooq Ahmad Bhat, Saduf Afzal, and Asif Iqbal Khan. *Advances in deep learning*. Springer, 2020.
- [Woo13] Brent L Wood. Flow cytometric monitoring of residual disease in acute leukemia. In *Hematological Malignancies*, pages 123–136. Springer, 2013.
- [WRW⁺22] Matthias Wödlinger, Michael Reiter, Lisa Weijler, Margarita Maurer-Granofszky, Angela Schumich, Elisa O Sajaroff, Stefanie Groeneveld-Krentz, Jorge G Rossi, Leonid Karawajew, Richard Ratei, et al. Automated identification of cell populations in flow cytometry data with transformers. *Computers in Biology and Medicine*, 144:105314, 2022.

- [YNDT18] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.
- [ZLLS21] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.
- [ZLW18] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.
- [ZRSTL18] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.