POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Segmentation of Cell Nuclei from Histopathological Images

Tesi di Laurea Triennale in
Ingegneria Biomedica

Author: **Ludovica Beretta, Gianfranco Botta, Chiara Cavigliano, Luca Drole.**

Student ID: 956497, 959222, 955956, 960889
Advisor: Prof. Marco Domenico Santambrogio
Co-advisors: Ph.D. Eleonora D'Arnese and Dott. Ing. Isabella Poles
Academic Year: 2022-2023

We dedicate this thesis to Beatrice, Maurizio, Elena, and Giuseppe.

Acknowledgements

We want to sincerely thank our co-advisors, Isabella Poles and Eleonora D'Arnese, for their constant support. Thanks to their valuable feedback and insights, working on this thesis has represented a significant opportunity for our personal and academic growth. Furthermore, we wish to thank our advisor, Professor M. Santambrogio, for always being on the front line in providing unique experiences and projects to his students, going above and beyond his didactic obligations, and for creating a friendly and stimulating environment for us. Finally, we owe our heartfelt thanks to our parents, whose support allowed us to pursue our studies at the Politecnico di Milano, enabling us to delve in topics that truly passionate us at the highest level.

Abstract

The detection of various types of cells characterizing the Tumor MicroEnvironment (TME) is of vital importance for cancer prognostication, as well as to research. Indeed, the organization and structure of the TME can contribute to the assessment of the clinical state of the tumor, while helping to predict its possible development. However, detecting, segmenting, and classifying nuclei from tissue images takes up precious time for pathologists, preventing them from undertaking higher-value activities. Besides, the errors caused by fatigue and subjectivity may sometimes lead to different evaluations of the same cell type. It is, therefore, advisable to exploit Machine Learning (ML) solutions to automate these steps and to prevent the procedure from being overly influenced by inter- and intra-observer variability. Hence, we exploit a supervised Deep Learning (DL) framework to segment and identify epithelial cells, Tumor-Infiltrating Lymphocytes (TILs), Tumor-Associated Macrophages (TAMs), and neutrophils from microscopy images, obtained by staining pathological tissue samples with Hematoxylin and Eosin (H&E). We employ a large annotated dataset, featuring images obtained from patients affected by cancer in four different anatomical districts. Furthermore, we design a series of images pre-processing steps in order to optimize the model's performance. We employ a U-Net-based model, called HoVer-Net, which features three specialised decoders. The decoders approach separates the tasks of cell segmentation and classification while sharing a common encoder. The network is designed to be able to efficiently differentiate between overlapping or adjacent cells. We experiment with an array of different combinations of loss functions to address the significant class imbalance within the employed dataset. The best result is obtained by implementing the Asymmetric Unified Focal (AUF) loss function in the network, achieving a mean Panoptic Quality (mPQ) of 0.329, a mean Dice Score ($mDSC$) of 0.493 and a binary Dice Score ($bDSC$) of 0.768. Finally, we show how comparable results can be obtained by only analysing the blue color channel of the H&E stains present in the employed dataset.

Sommario

La rilevazione di profili cellulari che caratterizzano il MicroAmbiente Tumorale (TME) è di vitale importanza per la prognosi di tumori, così come per la ricerca scientifica. Infatti, l'organizzazione e la struttura del TME possono contribuire in maniera significativa alla valutazione dello stato clinico del tumore, e, al contempo, contribuire alla previsione di possibili sviluppi. Tuttavia, rilevare, segmentare e classificare i nuclei presenti nelle immagini dei tessuti richiede tempo prezioso ai patologi, impedendo loro di svolgere attività di maggior valore. Inoltre, gli errori causati dalla soggettività dell'osservazione può, talvolta, portare a valutazioni discordanti riguardo uno stesso tipo cellulare. È, pertanto, consigliabile utilizzare soluzioni di Machine Learning (ML) per automatizzare questi passaggi e prevenire l'eccessiva incidenza di variabili, influenzate da osservazioni interne o esterne, sul risultato dell'analisi. Abbiamo quindi utilizzato un framework automatico di Deep Learning (DL) per segmentare e classificare cellule epiteliali, linfociti, macrofagi, e neutrofili, da immagini al microscopio, ottenute mediante colorazione di campioni di tessuto patologico con Ematossilina ed Eosina (H&E). Abbiamo utilizzato un ampio dataset annotato, composto da immagini ottenute da pazienti affetti da tumore in quattro diversi distretti anatomici. Inoltre, abbiamo progettato una serie di passaggi di pre-elaborazione delle immagini al fine di ottimizzare le prestazioni del modello. Abbiamo utilizzato un modello basato su U-Net, chiamato HoVer-Net, che presenta tre decoder specializzati. I decoder affrontano separatamente i compiti di segmentazione e classificazione delle cellule, condividendo lo stesso encoder. La rete è progettata per differenziare efficacemente cellule sovrapposte o adiacenti. Abbiamo sperimentato diverse combinazioni di funzioni di perdita per affrontare il significativo sbilanciamento delle classi all'interno del dataset utilizzato. Il miglior risultato è stato ottenuto implementando la funzione di perdita Asymmetric Unified Focal (AUF) all'interno della rete, raggiungendo così una qualità panottica media (mPQ) di 0.329, un Dice Score medio ($mDSC$) di 0.493, e di un Dice Score binario ($bDSC$) di 0.768. Infine, abbiamo mostrato come possano essere ottenuti risultati comparabili analizzando solo il canale del blu delle colorazioni di Ematossilina ed Eosina presenti nel dataset utilizzato.

List of Figures

List of Tables

List of Abbreviations

Acronym	Description
AI	Artificial Intelligence
AUF	Asymmetric Unified Focal
bDSC	binary Dice Similarity Coefficient
CE	Cross Entropy
CNN	Convolutional Neural Network
DL	Deep Learning
DSC	Dice Similarity Coefficient
DQ	Detection Quality
E	Energy landscape
ECM	ExtraCellular Matrix
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive
GNN	Graph Neural Network
GPU	Graphic Processing Unit
GSL	Graph Structure Learning
GT	Ground Truth
H&E	Hematoxylin and Eosin
HV	HoVer
IoU	Intersection over Union

M	Marker
mDSC	mean Dice Similarity Coefficient
MSE	Mean Squared Error
ML	Machine Learning
MoNuSAC	Multi-Organ Nuclei Segmentation and Classification challenge
mPQ	mean Panoptic Quality
MRF	Markov Random Field
NC	Nuclear Classification
NN	Neural Network
NP	Nuclear Pixel
PQ	Panoptic Quality
ReLU	Rectified Linear Unit
RF	Random Forest
RGB	Red, Green, Blue
RoI	Region of Interest
SONNET	Self-guided Ordinal regression Neural NETwork
SENC	Structure Embedded Nucleus Classification
SQ	Segmentation Quality
SVM	Support Vector Machine
TAM	Tumor-Associated Macrophages
TIL	Tumor-Infiltrating Lymphocytes
TME	Tumor Micro-Environment
TP	True Positive
WCCE	Weighted Categorical Cross-Entropy
WSI	Whole Slide Images

Contents

1 | Introduction and Motivation

It is widely recognized that the proliferation of tumor cells within the human body poses a significant health concern. What makes it particularly dangerous is its challenging diagnosis, as the manifestation of the initial symptoms is often linked to an already advanced stage of proliferation [24]. The Tumor MicroEnvironment (TME) constitutes the complex ecosystem that surrounds a tumor. In healthy conditions, the cellular microenvironment is responsible for maintaining normal tissue functions, whereas, in the case of cancer, the tumor cells can alter their surroundings to promote their own growth, resulting in the proliferation of tumor cells within the organism, and consequently in critical health conditions [15]. Diverse immune cells have the ability to infiltrate tumors, and the composition and organization of these cells within the TME are closely related to the development and the clinical outcome of cancer patients [23]. Moreover, studying the interplay between the TME and the tumor's proliferation can be useful for the development of novel treatment strategies [8]. Our thesis focuses on the observation of specific cell types that play relevant roles within the TME, with particular emphasis on lymphocytes, neutrophils, tumor cells, macrophages, and epithelial cells as shown in Figure 1.1. These, along with other cells that constitute the ExtraCellular Matrix (ECM), are responsible for starting and directing the immune response [62]. To analyse such cell types, it is useful to rely on histopathological imaging, which serves as a valuable tool for presenting an accurate depiction of the damaged tissue and its immediate vicinity, enabling the study of morphological changes resulting from abnormal cell proliferation. Thus, histopathological slides of the TME can provide important information for diagnosis, prognosis, and research. Furthermore, characterizing the TME can help provide valuable insights into the initiation, development, invasion, and clinical outcome of tumor masses. For instance, characterizing the density and distribution of Tumor-Infiltrating Lymphocytes (TILs) has been shown to be predictive of lung cancer recurrence [16]. Nonetheless, the manual identification of cells from histopathological slides can be time-consuming for pathologists. Moreover, it can result in increased errors and potentially incorrect diagnoses, as it is prone to inter- and intra-observer variability, thereby compromising consistency within the obtained results. In this challenging context, the application of techniques based on

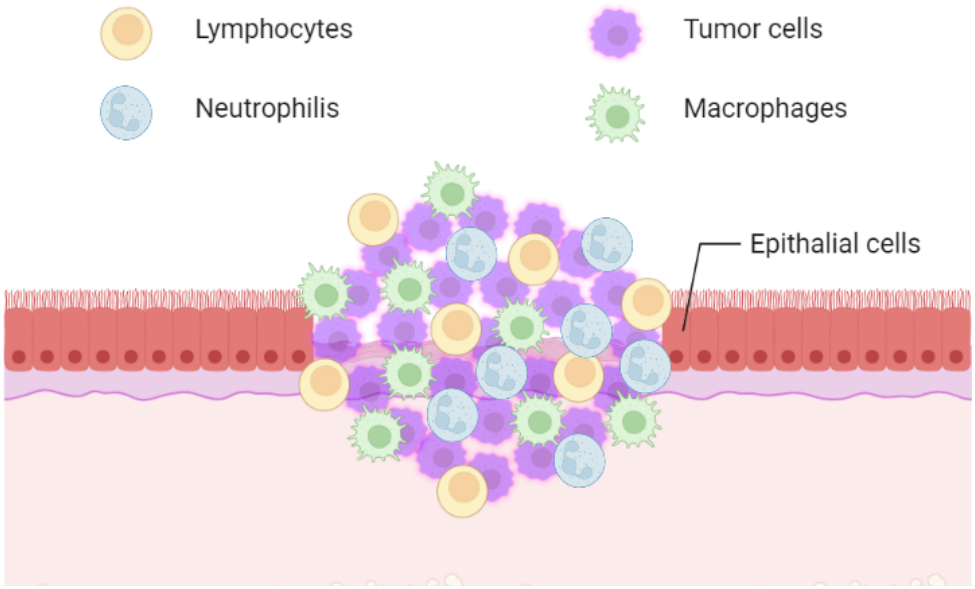

Figure 1.1: An example of TME with particular attention to: lymphocytes, neutrophils, tumor cells, macrophages, and epithelial cells.

Artificial Intelligence (AI) for the automatic detection and classification of cell nuclei can efficiently enhance TME analysis not just allowing experts to save time, but also allowing clinics and researchers to achieve significant advancements in terms of efficiency, accuracy, and reproducibility in the evaluation and characterization of the cellular dynamics within the TME [17]. During the last decade, AI has been growing in popularity to address biomedical segmentation and classification tasks. The rapid development of computer vision algorithms has enabled more accurate analysis of histopathological images using non-Machine Learning (ML) models, ML models, and, Deep Learning (DL) models, which have outperformed all of the previous methods if provided with large enough amounts of data. Between these strategies, DL techniques can enhance the accuracy, speed, and scalability of histopathological image analysis, potentially leading to improved patient outcomes and more efficient healthcare workflows [64]. Thus, in our thesis we propose a DL algorithm with the objective of segmentation and classification of cellular nuclei from histopathological images, exploiting the Multi-Organ Nuclei Segmentation and Classification challenge (MoNuSAC) dataset.

1.1. Outline

Chapter 2 accurately defines the TME, and the process relative to the acquisition of histopathological images. Furthermore, it delves into the main solutions to the task of

image segmentation and classification presented during the last decade, with a focus on the most exploited DL architectures in the biomedical field: Convolutional Neural Networks (CNNs), the U-Net model, and the ResNet model. Chapter 3 recaps the most effective techniques to segment and classify nuclei from histopathological images. Chapter 4 describes the pipeline concerning our computer vision problem. In particular, we dive into the dataset description and analysis, the pre-processing, the data augmentation, the chosen Neural Network (NN), the loss functions, the training optimizer, and the metrics to evaluate the results. Chapter 5 details how we conducted our experiments, the choices we made with the relative motivations, and shows our results under different scenarios and evaluation metrics. In Chapter 6 we evaluate our results and conduct a comparative analysis. Finally, Chapter 7 focuses on conclusions and future steps to improve our workflow and achieve better results in terms of accuracy and computational efficiency.

2 | Background

This Chapter provides a description of the basic concepts underlying our thesis. Section 2.1 presents an analysis of the clinical background, while Section 2.2 gives an overview of the technical tools employed for our purpose delving into the evolution of algorithms for automated segmentation and classification of cell nuclei from threshold approaches, to the more advanced DL-based models.

2.1. Clinical Background

This Section presents the clinical foundations of our thesis. In Section 2.1.1 we provide the clinical definition of tumors and describe the leading diagnostic processes. In Section 2.1.2 we outline the organization of the TME and its influence on the progression of cancer. Eventually, Section 2.1.3 illustrates the process of histopathological examination.

2.1.1. Tumors and Diagnostic Processes

Tumors are defined as the anomalous proliferation of cells within a specific tissue, resulting in the formation of a mass or lump within the body [2]. More in detail, tumors that exhibit expansive growth without infiltrating adjacent tissues are classified as benign, whereas malignant tumors, commonly referred to as cancer, have the ability to metastasize distant organs through the bloodstream and lymphatic system [2]. Repeated exposure to carcinogens, such as tobacco smoke, high-energy radiation, dioxins, and some viral infections can be responsible for genetic and epigenetic alterations that may lead to the development and progression of cancer [3]. Such alterations often involve significant changes to the cellular cycle, a loss in genomic stability, and the resistance to cell death mechanisms [43]. Both benign and malignant tumors pose substantial challenges in the diagnostic process. The diagnostic process usually involves a multidimensional approach, which incorporates clinical evaluation, advanced imaging techniques, pathological examination, and molecular diagnostics. These approaches contribute to determining the tumor's location, size, extent of infiltration, and histopathological characteristics [1]. Then, once the proper diagnosis has been made, appropriate treatment measures can be initiated. Thus, the se-

lection of appropriate treatment modalities is based on a comprehensive assessment of the clinical variables, including tumor type and stage, and the patient's medical background, aiming to achieve optimal outcomes while minimizing adverse effects.

2.1.2. Tumor MicroEnvironment

The Tumor MicroEnvironment (TME) is a complex system where neoplastic cells interact with multiple components and shape the tumor's evolution, invasivity, and metastatic ability. It comprises various cell types and extracellular components, supported by a vascular network. Such components can be:

- Immune cells: they can either recognize and eliminate cancer cells or, in some cases, support their growth and metastasis (e.g. macrophages) [30].
- Stromal cells: they provide structural support to tissues.
- ECM: a network of proteins and sugars that surrounds cells, which can be manipulated by the tumor cells.
- Signalling molecules: they can promote tumor growth.

As the tumor proliferates, the TME is remodeled, growing in heterogeneity and complexity, reacting to the hypoxia and acidosis caused by the abnormal growth of neoplastic tissue. Numerous studies have demonstrated the value of investigating TME development to identify prognostic biomarkers, which can aid in determining the most appropriate therapeutic approach [53]. Moreover, the TME plays a crucial role in modulating the expression of surface receptors, activating signaling pathways, and influencing the overall therapeutic response. Recent advancements in the understanding of TME have led to the emergence of novel therapies. For instance, monoclonal antibody therapies targeting various components of the TME have yielded significant results in treating several types of malignancies [31].

2.1.3. Histopathology

The observation of digital images that represent the change in morphology of the tissue attacked by abnormal cell growths is of great importance, and histopathological imaging is nowadays the best solution to this issue. Histopathology, in the field of scientific inquiry, is the investigation of tissue alterations induced by disease or abnormal cell growth [59]. This multidisciplinary approach entails a comprehensive analysis of tissue structure and composition. The histopathological analysis process consists of a series of consequential

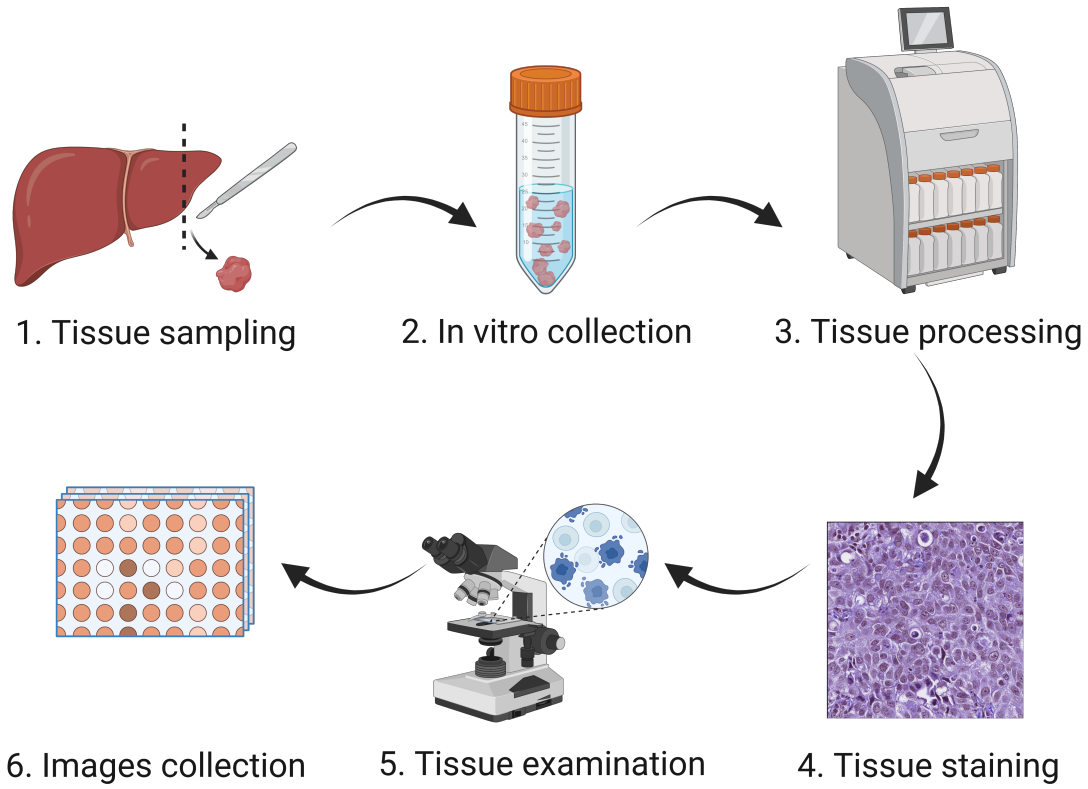

Figure 2.1: The process of histopathological imaging.

steps, shown in Figure 2.1. The first step is tissue sampling, which consists of collecting a sample of the interested tissue from the patient, usually through biopsy or surgical excision [58]. Then, during the in vitro collection, the sample is immersed in a preservation solution to prevent the tissue from decomposing. Over the course of the tissue processing, the sample undergoes fixation in a formalin solution, which helps preserve its structural integrity [58]. Subsequently, the fixed sample is embedded in paraffin wax to provide support and facilitate the slicing process, and finally, thin slices of the embedded sample are cut and mounted onto glass slides, ready to be stained. The tissue staining procedures enhance the visualization of distinct structures and cells within the tissue [58]. One commonly employed staining method involves the use of Hematoxylin and Eosin (H&E), which marks the nuclei in shades of blue and the cytoplasm in shades of purple, facilitating the interpretation of cellular morphology and tissue architecture [20]. Eventually, the sections are examined by experts through histopathological analysis, and the outcome of this process yields a collection of digital images that accurately represents the pathological condition under investigation [49], [58].

2.2. Automated Image Segmentation and Classification

During the observation phase, it is of great relevance the acquisition of information regarding the presence and the level of differentiation of specific cells, such as immune cells, that influences the tumor initiation and proliferation [62]. The extraction of pertinent information necessary for accurate tumor diagnosis and effective treatment can rely on image segmentation and classification techniques. These techniques consist of partitioning an image into sub-components that present homogeneous features, followed by their subsequent classification [51]. Manual segmentation and classification of medical images by experts in the field is not only a tedious and time-consuming process but also subjective, since it mainly depends on the visual interpretation of the analyst. Consequently, the use of computer-aided systems allows to overcome these limitations [47]. Different algorithms have been proposed. These can be clustered into three different classes: non-ML methods, ML methods, and DL methods [51]. DL algorithms have emerged as top performers in recent years, with CNNs, U-Net, and ResNet being the most common architectures utilized.

2.2.1. Convolutional Neural Networks

CNNs have produced excellent results not only in image segmentation and classification but also in other fields, such as video processing, natural language processing, speech recognition, and many others [7], [22], [38]. Such results have been possible because CNNs architecture is inspired by the human visual cortex and the connections between neurons in the human brain. In fact, the primary goal of convolution is to extract meaningful features from the images in a given dataset. To perform this essential step, lots of convolution filters are sequentially applied to the input image to find the so-called feature maps. The first filters usually capture low-level characteristics, such as color, and edges, whereas the high-level characteristics, such as the presence of a nucleus in the image, are learned by the deeper layers. This structure mimics the layer structure that is found in the visual cortex ventral pathway, where, the first layers of neurons recognize simpler features, while as we proceed through the visual pathway, the features become more complex. As the visual stimuli pass through the layers, some neurons focalise on specific locations in the image, in a similar way to how convolution operations are applied in a CNN [39]. Before each convolution step, padding is usually applied to prevent the size of the feature map from shrinking at each layer. This method is known as the same convolution; otherwise, as valid convolution. Another important choice concerning the filters is the stride, which is the step

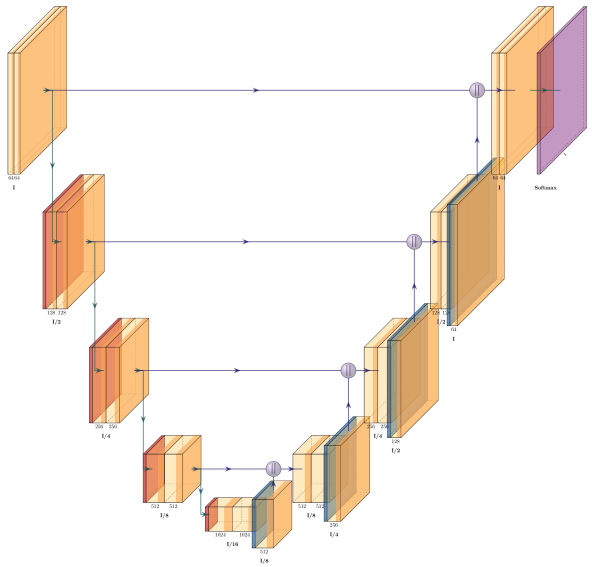

Figure 2.2: U-Net architecture. From the left to the right the encoder and decoder networks, forming the classical "U-shape". The arrows indicate the skip connections.

with which the convolution matrix moves on the input image or the feature maps. After obtaining the feature maps, it is necessary to add a pooling (or sub-sampling) layer, which convolves the features' spatial size. Two forms of pooling are usually exploited in CNN: maximum pooling and average pooling. Moreover, the output of every filter is provided to an activation function. A multitude of activation functions have been exploited over the years, but the most effective employed for CNNs is the Rectified Linear Unit (ReLU). Finally, depending on whether the network is for image segmentation or classification purposes, its architecture could be different. When dealing with segmentation, after the classical encoder, a decoder serves to up-sample the last feature map to obtain a pixel-level segmentation map. Whereas, when dealing with classification, the output of the last pooling layers is flattened, which means that the ultimate feature maps are unrolled into a vector. Finally, the vector is fed as the input of some fully-connected layers, that are able to learn the nonlinear combination of high-level features represented by the input vector [12]. The very last neurons produce real values as outputs, which need to be transformed into a meaningful form. The SoftMax function and the Sigmoid function are often employed at the end of CNNs to map the real values of the output to values between 0 and 1, which now represents a probability. For binary classification and segmentation problems, the Sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

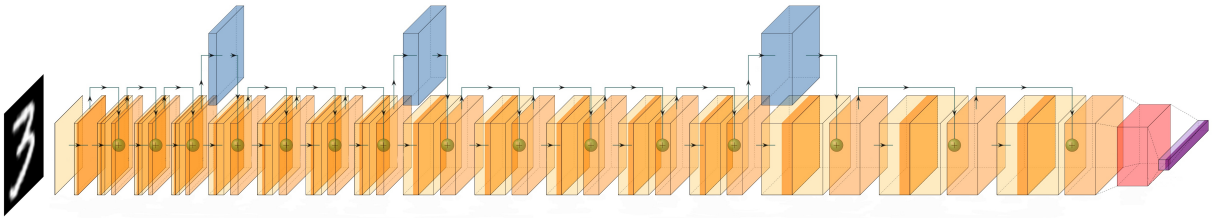

Figure 2.3: ResNet architecture. The residual block (in blue) skips some layers to perform a more effective back-propagation of the gradient.

where x is the output of a neuron in the last layer and the output σ of the Sigmoid function is used to express the confidence in a given prediction. Similarly, for multi-class problems, the SoftMax function is defined as follows.

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{2.2}$$

In Equation (2.2) x_i represents the input value for the i-th class, while $exp(x_i)$ denotes the exponential function applied to x_i, and the sum is taken over all classes j. With this type of formulation, it is ensured that the sum of all probabilities is 1. Therefore, Equation (2.2) is used to model class probabilities.

2.2.2. U-Net

The U-Net is a neural network initially developed for biomedical image segmentation. It is an advancement of a classical CNN which has provided the foundation for multiple improvements in the field during the last years. The name U-Net comes from its U-shaped architecture, comprising an encoder, that captures contextual information and extracts high-level features, and a decoder that generates a pixel-level segmentation map by upsampling the features obtained from the encoder. Particularly, the encoder is designed like a typical CNN. The true innovation of U-Net is to connect the encoder and the decoder using skip connections, which allow the network to better localize and preserve fine-grained details during the up-sampling phase. Moreover, the last layer consists of a 1x1 convolutional layer followed by a SoftMax activation function, which produces a pixel-level probability map, where each pixel represents the likelihood of belonging to a particular class or category [54].

2.2.3. ResNet

Generally, the key concept behind a residual network is residual learning, which addresses the problem of degradation in deep neural networks such as U-Net. In fact, as networks get deeper, they tend to suffer from diminishing gradient performance or even degradation. Gradient degradation, also known as the vanishing gradient problem, refers to the issue where the gradients computed during back-propagation become extremely small, as they propagate through the network [26]. This is especially relevant in networks that present many layers, leading to slow or ineffective learning. Thus, a residual network tackles this problem by introducing residual blocks that enable the network to learn residual functions. These blocks contain shortcut connections, also known as identity mappings, that skip one or more layers. By directly propagating the gradients through these shortcuts, a residual network enables the network to learn the difference between the desired mapping and the identity mapping, thus allowing the network to focus on learning only the residuals [41].

3 | Related Work

This Section details the evolution of algorithms concerning segmentation and/or classification of nuclei, from threshold approaches, to the more advanced DL-based models, that have achieved promising results through the years.

3.1. Semi-Automatic Methods

Within the scientific literature, a wide variety of non-ML methods for image segmentation has been documented. All of these approaches rely on the utilization of predetermined algorithms, meaning designed algorithms consisting of specific sets of instructions or rules that are established in advance to perform a particular task or solve a problem [47]. Notably, the primary method for image segmentation is the thresholding approach, where one threshold value T is chosen to obtain a binary image: pixels with intensity above T are approximated to the foreground region, and all other pixels to the background one [51]. The choice of the threshold value can be automated, optimizing an objective function, based on the variance between classes, such as the Otsu's function, or the Kapur's function, based on the entropy [50]. Additional non-ML algorithms are region-based, where the image is divided into regions that are similar according to a set of predefined criteria [47]. Within this category, there are methods, such as region growing, region splitting, region merging, or the Watershed approach, in which the image is seen as a topographic surface where low-intensity pixels are interpreted as valleys of surfaces, while high-intensity pixels appear as hills or peaks [51]. Furthermore, some approaches aim to address image segmentation by detecting edges, linking pixels between different regions that have rapid intensity transitions, or by aggregating homogeneous data into clusters, relying on an analogy-based criterion [47], [51]. Although their early adoption, non-ML methods are limited in their ability to handle complex scenarios and variations amongst different images [51]. Thus, the advent of ML algorithms, has led to significant advancements in this field, thanks to their ability to learn specific parameters during the training phase, in which the algorithm strives to classify elements and derive knowledge from its errors [51].

3.2. Machine Learning Methods

ML models can be categorized into unsupervised or supervised methods. In unsupervised segmentation and classification, the algorithm identifies hidden patterns within unlabelled input data and has the ability to learn and organize information without providing an error signal to evaluate the potential solution. In contrast, supervised learning is based on training the algorithm by exploiting data with correct segmentation map and classification label Ground Truth (GT) already assigned [55]. Earlier ML approaches focused on the extraction of hand-crafted features to distinguish between nuclear and non-nuclear pixels using algorithms that incorporate the adjustment of specific weights during the training phase, to enhance the performance of segmentation and classification. Examples of classical ML methods are the Support Vector Machines (SVMs), an ensemble of techniques that builds a non-probabilistic binary classifier, the Random Forest (RF), a collection of learning methods that create a forest of random uncorrelated decision trees to arrive at the best possible answer, and, finally, the Markov Random Field (MRF) which is a conditional probability model [56]. However, classical ML methods require relevant attributes or features which should be manually selected or extracted. The advent of DL algorithms has addressed this issue by employing the error back-propagation to adapt the weights of neural connections and consequently improve performance and accuracy [29].

3.3. Deep Learning Methods

As mentioned before, ML utilizes conventional models and algorithms for automatic learning, whereas DL have demonstrated to autonomously extract intricate representations from data without requiring manual feature selection allowing effective generalization to novel data after being trained with a substantial and diverse dataset [29]. This indicates that, under appropriate training conditions, DL algorithms have the potential to proficiently perform image segmentation and classification tasks, demonstrating generalizability across various patient populations, tissue types, staining protocols, and imaging conditions [61]. Despite the necessity of having at disposal a wide range of data, both in terms of quantity and quality, DL algorithms have led to excellent results in image processing. As mentioned in Section 2.2.2, one of first NN employed in the field of biomedical image segmentation is the U-Net. It consists of an encoder that extracts features and a decoder that restores the feature maps to the original space [54]. However, its main innovation is the introduction of skip connections between the two components, which facilitate the integration of low-level information that otherwise may get lost during feature extraction. To exemplify, *Fang et al.* used the U-Net architecture for the segmentation of cell nuclei

in the 2018 Data Science Bowl [14]. Worth mentioning modifications to the basic U-Net architecture, namely U-Net++ and Micro-Net, have been introduced to enhance the performance of nuclei segmentation. *Zhou et al.* created U-Net++ to improve the instance segmentation of individual nuclei by incorporating nested, dense skip pathways [68], while *Raza et al.* exploited Micro-Net to address the challenge of nuclei with varying sizes, by processing the input at multiple resolutions [52]. Furthermore, Mask R-CNN has been introduced to segment and classify nuclei simultaneously, joining three branches: the first for classification, the second for bounding box regression, and the last one for predicting segmentation masks on each Region of Interest (RoI). In particular, the mask branch is a small Fully Convolutional Network (FCN) applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner [28]. In this context, *Lv et al.* created Nuclei R-CNN by modifying the model setting of Mask R-CNN and adapting it to nuclei segmentation task [42]. Another relevant advancement of U-Net is HoVer-Net, which extends upon the U-Net architecture. HoVer-Net is a NN comprising one encoder and three decoder branches, performing complementary tasks. The three branches comprise the Nuclear Pixel (NP) branch, which provides semantic segmentation maps, the HoVer (HV) branch, which produces two maps predicting the horizontal and vertical distances between nuclear pixels and their centers of mass and, finally, the Nuclear Classification (NC) branch, which determines the type of each nucleus by aggregating the predictions within each instance. The first two branches jointly produce a nuclear instance segmentation map and enhance the segmentation of overlapping nuclei [25]. Besides, even HoVer-Net has undergone numerous upgrades, such as the ones proposed by the Self-guided Ordinal regression Neural NETwork (SONNET) and the Structure Embedded Nucleus Classification framework (SENC). SONNET shares with HoVer-Net the decoder architecture, but the key distinction lies in the exploitation of ordinal regression methodology instead of the HoVer branch. This technique consists of stratifying nuclei in a way that inner regions, forming regular shapes of nuclei, are separated from outer regions forming an irregular shape [19]. *Doan et al.* used SONNET with the aim of segmenting and classifying histopathological images from different public datasets. The last enhancement of segmentation and classification has been designed by *Lou et al.*: SENC uses an instance segmentation map from HoVer-Net to feed a Graph Neural Network (GNN) inside a Graph Structure Learning (GSL) architecture, to learn spatial information amongst all nuclei in the original image and not only in each patch [40].

Given the previous details, we propose a HoVer-Net-based model and we optimize the loss functions to obtain an enhancement of the segmentation and classification performances. We chose HoVer-Net not only because *Graham et al.* obtained the best performance in

the 2020 MoNuSAC by relying on this architecture, but also because it appears to provide a valuable compromise between high-level segmentation and classification, and accessible computational costs for training.

4 | Methodology

This Chapter provides a description of the proposed methodology and details the implementation choices in the proposed pipeline. Section 4.1 presents an analysis of the dataset that has been used for the implementation, highlighting the distribution of image classes. Section 4.2 lays out the pre-processing steps that have been undertaken on the data to obtain optimal inputs for the NN model. Section 4.3 describes the employed architecture, while Section 4.4 focuses on the employed loss functions. Section 4.5 outlines the training optimizer utilized to manage the weights within the NN. Then, Section 4.6 presents a description of the post-processing techniques applied to the output data from HoVer-Net. Finally, Section 4.7 presents the metrics that have been employed to evaluate the performance of our model. In Figure 4.1 we illustrate the overall framework of our implementation.

4.1. Dataset Description

To implement the proposed solution we employed the publicly-available MoNuSAC2020 dataset, presented by *Verma et al.* [62] [4]. The MoNuSAC2020 dataset has been developed to facilitate the advancement of computational pathology providing a public platform where to test novel algorithms aimed to detect and characterize the cells within the TME. It is the first large, high-quality, multi-organ, multi-cell dataset that has been published and shared with the scientific community to address this task. The training and testing data were collected from Whole Slide Images (WSIs) available on the Cancer Genome Atlas data portal [6]. The dataset includes a set of 71 patients from 37 different hospitals, mainly located in the USA. The organizers partitioned the patients into two subsets: a training group of 46 individuals and a testing group of 25 individuals. The GTs for both subsets were released to allow the comparison of the performance metrics across various solutions. The dataset comprises images pertaining to samples selected from either breast, kidney, lung, or prostate tissue, as well as the segmentation masks relative to the nuclei of *epithelial cells*, *lymphocytes*, *macrophages*, and *neutrophils*. Additionally, the WSIs were cropped to increase the computational manageability of the data. Subsequently, all the

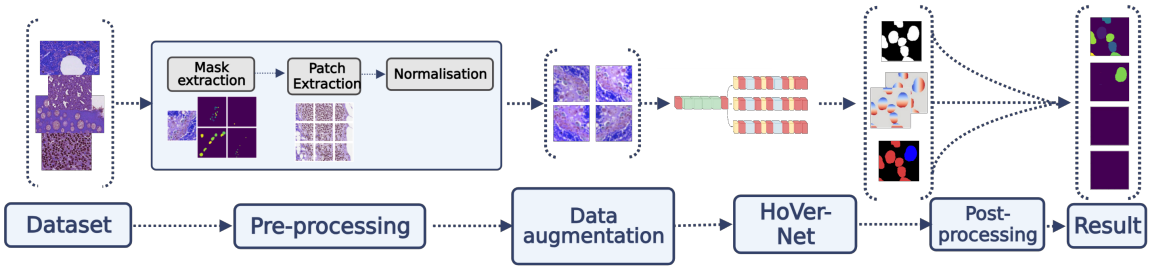

Figure 4.1: The overall framework. The dataset undergoes the pre-processing steps and the transformations of the data augmentation. Then the images are employed as inputs for HoVer-Net, and thanks to the post-processing step, we obtain the results from the three outputs of the network.

annotations were performed manually by engineering students under the supervision of an expert pathologist, in accordance with the protocol outlined in [35]. The annotations were revised iteratively until fewer than 1% of nuclei exhibited any errors. The ultimate dataset includes a total of 31,411 nuclei in the training set and 15,498 nuclei in the testing set, divided as shown in Figure 4.2. Indeed, as highlighted in Figure 4.2, it is clear how the dataset presents a significant class imbalance, since *epithelial cells* and *lymphocytes* prevail over the two other cell types, composing approximately 95% of the dataset. At the same time, Figure 4.2 shows how the distribution of the cells among the tissues does not present significant imbalances. More specifically, considering the training subset, 30.3% of the segmented nuclei come from breast tissue, while kidney samples account for 25.2%, lung samples for 20.1 %, and prostate cells sum up to 24% of the total. Moreover, the images available within the dataset present variable sizes and shapes, with the smallest image having a size of 90×98 pixels and the biggest presenting a size of 1422×2162 pixels. The aspect ratios within the available data are highly variable as well: considering the training set, the aspect ratio can vary from 1:1 up to 1:8.

4.2. Data Pre-Processing

Before feeding the dataset to the NN, it is necessary to perform pre-processing operations that make the data more usable by the model. These operations aim at transforming the data into a suitable format that facilitates learning and improves the networks' ability to extract meaningful features [60]. We start extracting the masks relative to each of the 4 nuclear classes from the dataset using an algorithm provided by the authors [4]. Providing the masks for each different nucleus class enables the NN to learn how to distinguish and classify nuclei accurately. As described in Section 4.1, the images in our dataset have

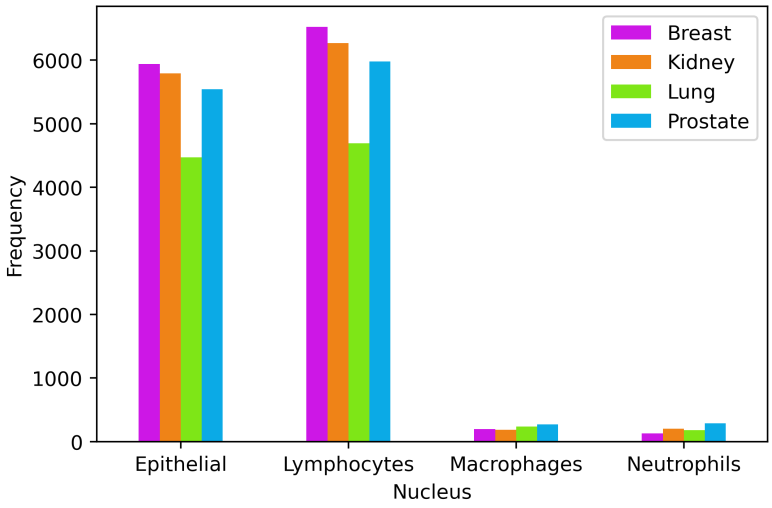

Figure 4.2: The distribution of cell types within the dataset.

vastly different sizes, while neural networks require a constant input shape. Therefore, we generate a set of sub-images, known as patches, to standardize the input image dimensions for the DL model, thus simplifying the training process. Multiple patches are extracted from a single image based on a sliding window, which moves with regular steps across the image. When the borders of the sliding window partially fall outside the image, the missing data is filled by mirroring the contents of the patch. Therefore, using excessively large patches and small step sizes would result in a significant number of heavy images that occupy a considerable amount of memory and slow down the training. On the other hand, reducing the patch size too much would lead to a loss of crucial information for segmenting and classifying the image. Therefore, to find the best compromise between memory usage and patch size, we decided to crop patches with dimensions of 256×256 and opted for a step size of 200, meaning that at each iteration the sliding window moves to the right, and subsequently downwards, of 200 pixels. Finally, to improve the convergence and the stability of the model's training, we normalize the patches one by one, by scaling the initial pixel values (ranging from 0 to 255) to values between 0 and 1. This helps mitigate the impact of varying pixel value ranges across different images, making the optimization process more efficient and ensuring fair treatment of different features during learning [44].

4.2.1. Data Augmentation

As mentioned in Chapter 1, the performance of DL models is strongly linked to the quality and quantity of data available for training. In fact, the greater the quantity and variability of the training data, the better is the model's performance. On the other hand, if these

conditions are not met, the model's training may be affected by overfitting. Overfitting refers to a situation where a ML model performs very well on the training data but fails to generalize well to new unseen data. The reason for this resides in the fact that the model may acquire knowledge of the fluctuations present in the training data rather than comprehending the fundamental features that characterize the subject of interest [37]. In other words, there is the risk that the model memorizes the training data, rather than learning from it. To exemplify, let us consider an imaginary dataset characterized by all cells being located on the right side of the images. It is possible that, if overfitting occurs during training, our model may perceive "being on the right" as a discriminating characteristic, mistakenly considering it significant for nucleus recognition when, in reality, it is merely a tendency within the dataset and does not reflect reality. Indeed, during the testing phase, when the network receives a patch aligned with the dataset's tendency (i.e., a nucleus on the right side), it will perfectly recognize those nuclei, though failing when the nuclei will be placed on the left.

Thus, we attempted to improve the performance of our model by increasing the quantity and quality of data for the network's training phase, using data augmentation. Therefore, we leverage a series of transformations of the images, while preserving the necessary features for accurate classification and segmentation. By applying transformations such as rotations, mirroring, decolorization, and modifications to brightness, contrast, and shades to the patches obtained during the pre-processing phase, we obtained more data to better train our model [9]. Thanks to these techniques, data augmentation proves to be very useful in increasing the quality of our dataset by introducing greater variability into it, allowing the models to extract the truly significant features for nucleus segmentation and classification.

4.2.2. Data Loader

We employ a *Data Loader* structure to iteratively feed the images with the corresponding masks to the NN. A *Data Loader* is a component that facilitates the efficient loading of the patches on the Graphic Processing Unit (GPU). During the training, the *Data Loader* provides groups of images to the neural network, called batches. Moreover, the *Data Loader* shuffles the images at each training iteration, thus contributing to the reduction of overfitting. Therefore, by utilizing a *Data Loader*, we can efficiently manage memory usage and minimize the risk of overwhelming the system's RAM capacity.

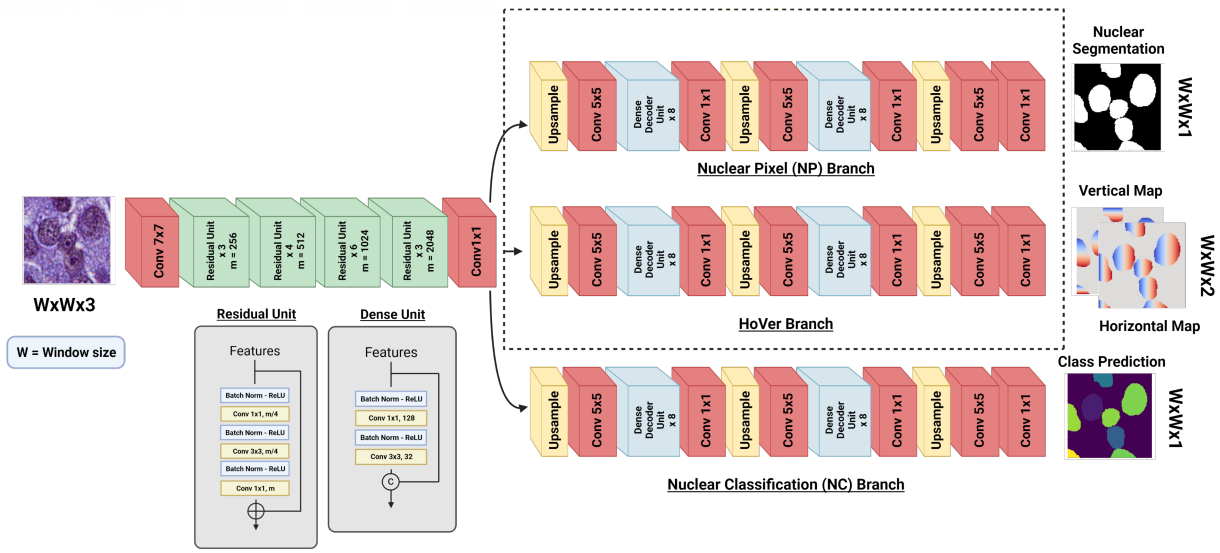

Figure 4.3: The HoVer-Net architecture. One encoder branch followed by three decoder branches. The two upper branches are for the segmentation task, while the other branch is for the classification task.

4.3. HoVer-Net Architecture

The images loaded by the *Data Loader* are subsequentially utilized to train our chosen DL model, namely HoVer-Net. Choosing such an architecture enables to perform instance segmentation and classification within the same network. In fact, segmentation before the advent of HoVer-Net was usually carried out in a separate initial step to take advantage of nuclear features for downstream analysis. However, an early segmentation step allows to overcome different challenges posed by H&E stained tissue slides. Firstly, nuclei display a high level of heterogeneity and they differ in shape, size, and chromatin pattern. Moreover, they present variance across disease types, and even within various regions of a single tissue sample. Finally, nuclei tend to cluster and form lots of overlapping instances as a consequence. With regard to the feature extraction component, it is inspired by Preact-ResNet50 [27], but the total down-sampling factor in the first convolution is reduced from 32 to 8, while the first max-pooling filter is removed to prevent the immediate loss of information, essential to perform an accurate segmentation. More precisely, the encoder comprises a 7×7 convolution layer at the beginning and a 1×1 at the end, as well as 16 residual units, generating 2048 feature maps. Each residual unit contains 3 convolution layers, whose dimensions are 1×1, 3×3, and 1×1 respectively, and 3 ReLU activation functions, one for each convolution layer. As for the up-sampling, HoVer-Net features three decoders based on the nearest neighbor method. Each decoder consists of 3 up-sampling layers, interspersed with 6 convolution layers and 12 dense decoder units, where

each dense decoder unit is composed of 2 ReLu activation functions and 2 convolutional layers of dimensions 1×1 and 3×3. Each decoder performs a complementary role to the other. The first one, named Nuclear Pixel branch (NP), deals with semantic segmentation, since it predicts whether a pixel belongs to a nucleus or to the background. The second one, named the HoVer branch (HV), is the real innovation brought by HoVer-Net, and it is responsible for the prediction of the horizontal and vertical distances of nuclear pixels to their centers of mass. The first two branches jointly focus on instance segmentation, creating a distinct instance for each nucleus. This approach resolves the issue of overlapping nuclei segmentation by effectively characterizing them as separate instances for subsequent individual classification. On the other hand, the third branch, named Nuclear Classification branch (NC), classifies the nucleus type for each pixel. By integrating this information with the outputs from the first two branches, a superior result is achieved, surpassing the previous state-of-the-art work [25]. Given the strong affinity between our goals and the ones described by [25], we decided to rely on HoVer-Net to successfully detect and classify epithelial cells, lymphocytes, macrophages, and neutrophils.

4.4. Loss Functions

To evaluate the fidelity of the model's predictions, individual decoders produce masks that capture both segmentation and classification details. These masks are compared against the ground truth masks provided in the dataset. This evaluation process involves the use of a loss function, that quantifies the discrepancy between the predicted masks, which are the output of the model, and the GTs. Eventually, the loss function provides a feedback signal to guide the learning process by adapting the weights of the layers [45]. DL models are trained using a technique called back-propagation, which iteratively adjusts the model's parameters to minimize the chosen loss function. The loss function acts as a guide for this optimization process, indicating the direction and magnitude of the parameter updates that need to be made. By minimizing the loss function, the model learns to perform more accurate predictions and becomes better at capturing the underlying patterns and relationships in the data. In addition to guiding the training process, the loss function also plays a crucial role in model evaluation. In fact, when the loss function is assessed on a separate validation set, it provides a quantitative measure of the model's performance and it allows for comparison between different models or hyperparameter settings. In our specific approach, we employed a combined loss function, which comprises three distinct components, with each component associated with one of the three decoder branches [25].

The first considered loss function is defined as:

$$L = \underbrace{(L_a\lambda_a + L_b\lambda_b)}_{\text{HoVer branch}} + \underbrace{(L_c\lambda_c + L_d\lambda_d)}_{\text{NP branch}} + \underbrace{(L_e\lambda_e + L_f\lambda_f)}_{\text{NC branch}} , \tag{4.1}$$

where L_a and L_b represent the regression loss with respect to the HV branch, L_c and L_d are the losses with respect to the NP branch, L_e and L_f are the losses with respect to the NC branch. By utilizing two different loss functions for each branch, it is possible to achieve an overall superior performance. λ_a, λ_b, λ_c, λ_d, λ_e, λ_f, are scalars that give weight to each associated loss function. In our implementation, following empyrical results obtained by the network's original authors, we set λ_b to 2 and the other weights to 1. The Hover-Net architecture consists of four distinct sets of weights: w_0 corresponding to a Preact-ResNet50, w_1 corresponding to the HV branch decoder, w_2 corresponding to the NP branch decoder, and finally, w_3 corresponding to the NC branch decoder. In order to optimize the network, these four sets of weights are simultaneously adjusted by minimizing the loss function L. We define L_a and L_b in Equation (4.1) as follows:

$$L_a = \frac{1}{n}\sum_{i=1}^{n}[p_i(I; w_0, w_1) - \Gamma_i(I)]^2 , \tag{4.2}$$

$$\begin{aligned} L_b = &\frac{1}{m}\sum_{i\in M}[\nabla_x(p_{i,x}(I; w_0, w_1)) - \nabla_x(\Gamma_{i,x}(I))]^2 \\ &+ \frac{1}{m}\sum_{y\in M}[\nabla_y(p_{i,y}(I; w_0, w_1)) - \nabla_y(\Gamma_{i,y}(I))]^2 , \end{aligned} \tag{4.3}$$

where ∇_x and ∇_y represent the gradient in the horizontal and vertical directions, while m denotes the total number of nuclear pixels within the image, and M denotes the set containing all the nuclear pixels. In particular, L_a is the Mean Squared Error (MSE) between the predicted horizontal and vertical distances from the nuclear center of mass and the relative GT, while L_b takes into account the MSE between the vertical and horizontal gradients, and the GT. Regarding the other branches, in Equation (4.1) we calculate the cross-entropy loss for L_c and L_e:

$$CE\_Loss = -\frac{1}{n}\sum_{i=1}^{N}\sum_{k=1}^{K} X_{i,k}(I)\log Y_{i,k}(I) , \tag{4.4}$$

and we calculate the dice loss for L_d and L_f as follows:

$$Dice\_Loss = 1 - \frac{2 \times \sum_{i=1}^{N}[Y_i(I) \times X_i(I)]}{\sum_{i=1}^{N}[Y_i(I) + X_i(I)] + \epsilon} \ . \tag{4.5}$$

In Equation (4.2) and Equation (4.3), X is the ground truth, Y is the prediction, K is the number of classes and ϵ is a smoothness constant that prevents any division by 0. We set ϵ to the value of 10^{-3}.

Furthermore, we experiment with a series of alternative loss functions. The first one is defined as follows:

$$L_W = \underbrace{(L_a\lambda_a + L_b\lambda_b)}_{\text{HoVer branch}} + \underbrace{(L_c\lambda_c + L_d\lambda_d)}_{\text{NP branch}} + \underbrace{(L_{e_W}\lambda_e + L_f\lambda_f)}_{\text{NC branch}} \ , \tag{4.6}$$

where L_{e_W} refers to the Weighted Categorical Cross-Entropy (WCCE), which takes into account the frequency of class samples within the dataset to improve the performance in under-represented classes. L_{e_W} is thus defined as follows:

$$\text{Weighted\_CE\_Loss} = -\frac{1}{n}\sum_{i=1}^{N}\sum_{k=1}^{K} w_i X_{i,k}(I) \log Y_{i,k}(I) \ . \tag{4.7}$$

With w_i in Equation (4.7) representing the weight associated with the specific i-th class. The second loss function we experiment with is defined as follows:

$$L_{Asym} = \underbrace{(L_a\lambda_a + L_b\lambda_b)}_{\text{HoVer branch}} + \underbrace{(L_{aUF}\lambda_{aUF})}_{\text{NP branch}} + \underbrace{(L_e\lambda_e + L_f\lambda_f)}_{\text{NC branch}} \ , \tag{4.8}$$

Where λ_{aUF} is a weight factor that we take as 1. With L_{aUF} we try to take advantage of the newly proposed Asymmetric Unified Focal loss (AUF), which enables selective class enhancement or suppression and is defined as:

$$L_{aUF} = \lambda L_{maF} + (1 - \lambda)L_{maFT} \ , \tag{4.9}$$

where λ is a weighting factor that determined the contribution of each loss term to the overall loss, which ranges between *0* and *1* [66]. L_{maF} is the modified asymmetric Focal loss and is defined as:

$$L_{maF} = -\frac{\delta}{N} y_{i,r} \log(p_{t,r}) - \frac{1-\delta}{N} \sum_{c \neq r} (1 - p_{t,c})^{\gamma} \log(p_{t,r}) \ . \tag{4.10}$$

In Equation (4.10), N is the total number of examples, $y_{i,r}$ denotes the GT examples where the rare class r is present, $p_{t,c}$ represents the GT probabilities for each class, while $p_{t,r}$ represents the GT probabilities only fore the rare class r. Furthermore, δ controls the relative weight of positive and negative examples, and γ is the focusing parameter, which controls both suppression of the background class and enhancement of the class r.

In addition, in Equation (4.9), L_{maFT} is the modified Focal Tversky loss [66]. It is defined as:

$$L_{maFT} = \sum_{c \neq r} (1 - mTI) + \sum_{c=r} (1 - mTI)^{1-\gamma} \ , \tag{4.11}$$

where, mTI is defined as:

$$mTI = \frac{\sum_{i=1}^{N} (p_{0i} g_{0i})}{\sum_{i=1}^{N} (p_{0i} g_{0i}) + \sum_{i=1}^{N} (p_{0i} g_{1i}) + (1-\delta) \sum_{i=1}^{N} (p_{1i} g_{0i})} \ . \tag{4.12}$$

In Equation (4.12), p_{0i} and p_{1i} are the probabilities of the pixel i to belong to the foreground class or to the background class respectively. g_{0i} is a parameter that takes values of 1 for foreground and 0 for background, while conversely g_{1i} takes values of 1 for background and 0 for foreground [66]. The AUF deals with class imbalance by generalizing and compounding Dice and Cross-Entropy losses. In fact, it could be shown that the AUF corresponds to different more simplified losses when its parameters are set to specific values [66].

4.5. Training Optimizer

During the training process, a DL model adjusts its parameters to minimize the difference between its prediction and the desired output. This adjustment is done through an iterative optimization process, where the optimizer plays a key role. Essentially, optimization algorithms serve as the intermediary component that integrates the loss function with the model and its parameters. While the loss function provides directional guidance, indicating the appropriate course of action, the optimizer facilitates the process by executing the necessary steps and adjusting the model accordingly [44]. For our task, we decided to use the Adam optimizer [33], a widely implemented algorithm for segmentation and classification applications [67]. Every time the network's weights need to be adjusted, the gradients

of the model's parameters are computed with respect to the loss function. The gradients indicate the direction and magnitude of the updates needed to minimize the error. When the gradient values are high, suggesting the parameter is moving in the right direction, the Adam optimizer decreases the learning rate to facilitate more precise convergence, and vice versa. Additionally, the optimizer keeps track of the gradient variations over time. The adaptability of the optimizer, achieved through learning rate adjustment, facilitates faster convergence by minimizing the loss function at each iteration. This enables the model to effectively learn from the data and improve its overall performance.

4.6. Post-Processing

In DL, post-processing refers to additional steps or techniques applied to the output of a NN model. These techniques are often specific to the task and the characteristics of the data. The goal is to refine and improve the raw outputs of the neural network models to achieve more accurate and reliable results. As explained in Section 3.1, the post-processing phase of *Graham et al.* involves a marker-controlled Watershed segmentation. Up to this point, it is necessary to compute the Marker (M) [65] and the Energy Landscape (E). M is obtained by applying a threshold function, whose values are chosen to yield optimal segmentation results, and the S_m value, that represents areas of significant differences between neighboring pixels in the horizontal and vertical maps and is defined as:

$$S_m = max(H_x(p_x), H_y(p_y)) \tag{4.13}$$

where p_x and p_y refer to the horizontal and vertical predictions at the output of the HV branch. H_x and H_y compute the horizontal and vertical derivative approximations. Besides, E is derived from the complement of the threshold function and the probability map. In such segmentation, M serves as the marker, providing information on where the boundaries between nuclei should be separated, while E is used as a guide for subsequent marker-controlled Watershed segmentation. Therefore, the nuclear probability map is split based on E, allowing for more accurate segmentation. The marker-controlled Watershed segmentation is used to obtain an instance segmentation. Moreover, the per-pixel nuclear type prediction obtained from the NC branch, described in Section 4.3, is converted into a prediction per nuclear instance. The majority class of predictions made by the NC branch is assigned to all pixels within each instance. This step ensures that each nuclear instance is associated with the class that has the highest frequency count within that given instance [25]. Overall, this process allows for refining the segmentation results obtained and integrating the instance segmentation map with the classification

prediction.

4.7. Evaluation Metrics

In a DL algorithm, a metric is a measure used to evaluate the quality of the output. This evaluation is based on the comparison between the model's predictions with the reference values known from the dataset. The choice of the metrics to be used is defined based on the specific objectives of the problem being addressed. The Dice Similarity Coefficient (DSC) is among the most commonly employed metrics for classification and segmentation tasks [10]. Dividing all the available segmentations into matched pairs (on True Positive, TP), unmatched GT segmentations (on False Negative, FN), and unmatched prediction segmentations (on False Positive, FP), the DSC is defined as:

$$DSC = \frac{2TP}{2TP + FP + FN} \ , \tag{4.14}$$

However, some works in the literature have raised skepticism regarding the employment of this metric to the task of segmentation in the case of histopathological images, as it excessively penalizes some instances involving overlapping nuclei [57]. We have, therefore, decided to rely mostly on the Panoptic Quality (PQ), which seemed to be more appropriate for the tasks of segmentation and classification of histopathological images [63]. Originally proposed for general image segmentation by *Kirillov et al.* [34], the PQ is defined as:

$$PQ = \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Detection Quality (DQ)}} \times \underbrace{\frac{\sum_{(x,y)\in TP} IoU(x,y)}{|TP|}}_{\text{Segmentation Quality (SQ)}} \ , \tag{4.15}$$

where x represents a GT segmentation, y represents a predicted segmentation, and IoU represents the Intersection over Union. *Kirillov et al.* also mathematically proved that each (x, y) pair is unique across the entire set of prediction and GT segmentations, if their $IoU(x, y) > 0.5$. From this, the Detection Quality (DQ) can be evaluated using the F1-Score, which is commonly used for instance detection assessment. The Segmentation Quality (SQ), instead, can be interpreted as a measure of how closely each correctly detected instance matches its corresponding GT segmentation. Throughout the training, the metrics are calculated for each image, nucleus type, and the average of all images is reported as final values for each different setup. It should also be noted how a recent paper has questioned the validity of the PQ as the best metric to be used for the seg-

mentation and classification of histopathological images. In their comprehensive study, *Foucart et al.* [21] highlighted an important aspect regarding the applicability of the PQ metric in the context of instance segmentation and classification. They brought attention to the fact that originally the PQ metric was specifically purposed for panoptic segmentation, which is a different task compared to instance segmentation and classification. Furthermore, they made a significant observation regarding the utilization of IoU as both a matching rule and a segmentation quality measure within the PQ framework, emphasizing its inadequacy when dealing with small objects, such as nuclei. These findings shed light on the challenges faced when employing the PQ metric for tasks that deviate from its intended application, particularly when confronted with the segmentation and classification of small-scale entities such as nuclei [21]. As for now, the computational pathology community has not developed an alternative evaluation framework following these observations. Thus, we decided to incorporate both the above-mentioned metrics as indicators of the success of our training. By utilizing both these metrics, we aim to assess more comprehensively the outcomes delivered by our algorithm.

5 | Experimental Evaluation

This Chapter contains the details regarding the experiments performed to obtain a combination of parameters and decisions with which the model would yield optimal performance during the training and testing phases. In particular, we provide a detailed description of the exploited libraries, the hardware components, the choice of the window, the step size, the learning rate, and other training variables, as well as the parameters set during the data augmentation process, together with the evaluation of different loss functions.

5.1. Experimental Setup

To train the network we use a 24GB NVIDIA RTX A5000 GPU. We choose to carry out our project in Python (version 3.8.10.), one of the best-known coding languages for data analysis [18]. In fact, Python features different built-in modules and functions to handle data in a rapid and intuitive way. We exploited the content of the original HoVer-Net repository, as a starting point for the dataset pre-processing step [25]. In this context, we customize the normalization approach by performing it at a patch level, rather than at an image level, to better exploit the entire range of conversion. Furthermore, we implement the data augmentation step exploiting Albumentations, an open-source image augmentation library [13]. After completing the pre-processing steps, we chose to use the PyTorch library [46] (v2.0.1) to train our DL model. Concerning the NN architecture implementation we use a pre-configured model from the PathML library [11], as it provides the basis to develop a straightforward and efficient pipeline for applying HoVer-Net to our data. The provided pre-configured network is also easy to reconfigure to meet the specificities of our dataset.

We split the original dataset from the MoNuSAC challenge (Section 4.1) in a training set corresponding to 80% of the patients, and in a validation and a test set comprising roughly 10% of the cases. Therefore, we end up with 36 patients in the training set, 6 patients in the validation set, and 4 patients in the test set. Afterward, we employ a patch size of 256×256 pixels to generate regularly-shaped images from the original ones. When generating the patches from each image, a step size of 200 pixels is used both horizontally

and vertically, maintaining the use of mirror padding, as explained in Section 4.2. Thus, our training set contains 2530 patches, while the validation set and the testing set 425 and 203 patches, respectively. Furthermore, we assure that the four classes share roughly the same distribution within the three subsets. Epithelilal cells and lymphocytes compose between 45% and 50% of the nuclei each, while macrophages and neutrophils account for a quantity that ranges between 1% and 4% for each cell type.

Subsequently, we augmented the dataset by applying both geometrical and video intensity transformations with 0.5 probability. In particular, for the firsts, we use horizontal and vertical flip, as well as 90 degrees rigid rotations, while for the latter we use color jitter, Gaussian, and median blur. Moreover, when applying color jitter we adjust the image brightness, contrast, saturation, and hue by multiplying the just mentioned image features by a value ranging from 0.1 to 1.9. To deal appropriately with memory limitations, we have set a batch size of 4 patches. By grouping the data into batches, we can process multiple patches simultaneously, taking advantage of parallelization to reduce the overall training time. We train our models for 75 epochs, using the Adam optimizer with a starting learning rate of 10^{-3}, decreased to 10^{-4} after 40 epochs. After each training, we save the weights configuration which achieved the best results over the validation set. We perform the training with 5 different setups to provide comparative results. We initially train the model with the original loss function described by Equation (4.1) without employing data augmentation, to later showcase the effectiveness of data transformations in both enhancing the model's performance and mitigating the issue of overfitting. Furthermore, we perform the training by keeping Equation (4.1) as the loss function and by only using the blue channel of the patches. Finally, we train our NN to minimize alternative loss functions. We consider L_w (Equation (4.6)), which takes into account the WCCE for the NC branch Section 4.3. We set the weights to balance the class distribution, setting them up as [1, 2, 2, 50, 50], where the first element represents the weight associated with the background class, followed by the weights associated with epithelial cells, lymphocytes, macrophages, and neutrophils respectively. We also consider L_{aUF} (Equation (4.8)), where we set the parameters as follows: $\gamma = 0.5$, $\delta = 0.6$, $\lambda = 0.5$, as recommended in [66]. As mentioned in Section 4.7, we use DSC and PQ metrics to provide a comprehensive evaluation of our results. It is relevant to mention that we calculate the DSC in binary and at a class-level for each presented setup. The binary DSC is computed by comparing the segmentation masks without taking into account any class distinction, meaning that we consider how well the network is able to distinguish the nuclei from the background.

Table 5.1: The results of our experiments, evaluated with the PQ per class and its mean.

	PQ_1	PQ_2	PQ_3	PQ_4	mPQ
Original Loss	0.371	0.322	0.160	0.327	0.295
No Data Aug.	**0.401**	0.304	0.069	0.337	0.278
Blue Only	0.371	0.323	0.108	0.398	0.300
Weighted CE	0.377	0.276	0.132	0.163	0.237
AUF	0.384	**0.326**	**0.206**	**0.401**	**0.329**

Table 5.2: The results of our experiments, evaluated with the DSC per class, its mean, and the binary DSC.

	DSC_1	DSC_2	DSC_3	DSC_4	$mDSC$	$bDSC$
Original Loss	0.531	0.442	0.376	0.460	0.452	0.731
No Data Aug.	**0.579**	0.413	0.114	0.467	0.392	0.713
Blue Only	0.532	**0.451**	0.302	0.531	0.454	0.750
Weighted CE	0.560	0.386	0.334	0.273	0.388	**0.768**
AUF	0.568	0.449	**0.403**	**0.550**	**0.493**	**0.768**

5.2. Experimental Results

In this section we present in Table 5.1 and in Table 5.2 the results of our models, across the setups illustrated in Section 5.1. The PQ and DSC values for each class in the MoNuSAC dataset are indicated with the subscripts 1, 2, 3, and 4 (epithelial cells, lymphocytes, macrophages, and neutrophils respectively). Moreover, the PQ's mean (mPQ), the DSC's mean ($mDSC$), and the binary DSC ($bDSC$) values are reported.

In our baseline training, performed with the three RGB channels and the original loss function in Equation (4.1), we achieve a mPQ of 0.295, a $mDSC$ of 0.452, and a $bDSC$ of 0.731. Training the network without the data augmentation step decreases the performance by 0.017 in terms of mPQ and of 0.060 in $mDSC$, while worsening the capability of segmentation with a $bDSC$ of 0.713. In Figure 5.1 and Figure 5.2 we provide insights into the training results, by comparing the original loss and the binary DSC obtained by training with and without data augmentation. Besides, when we perform a training exploiting only the blue channel, we achieve a mPQ of 0.300 and a $mDSC$ of 0.454. We note also that the $bDSC$, in this case, is 0.750. When testing with the WCCE (see Equation (4.6)) in the NC branch, we achieve a $mDSC$ of 0.388 and a mPQ of 0.237 in the testing phase. Finally, by implementing the AUF loss in the NP branch, explained in Section 4.4, we achieve a $mDSC$ of 0.493, a mPQ of 0.329, and a $bDSC$ of 0.768.

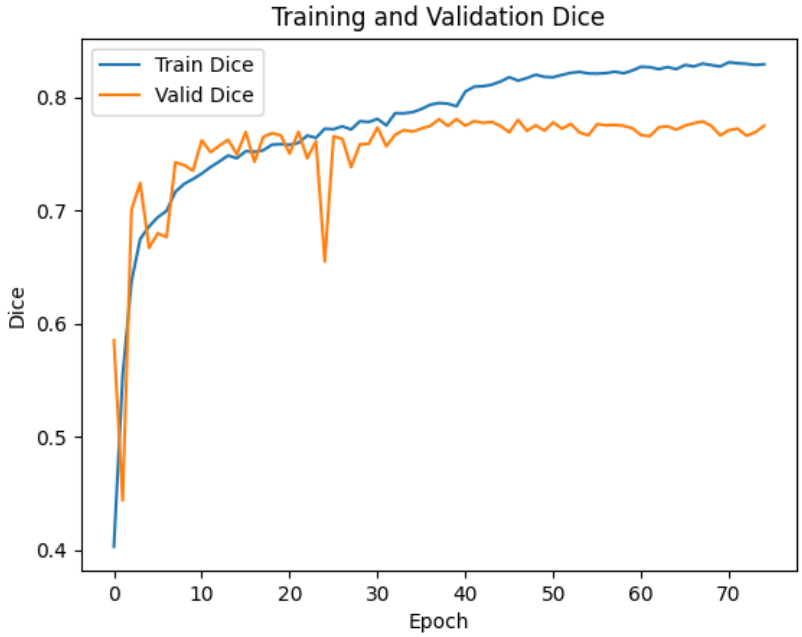

Figure 5.1: Binary *DSC* during the training with data augmentation using RGB images. The validation *DSC* fits well the training *DSC*.

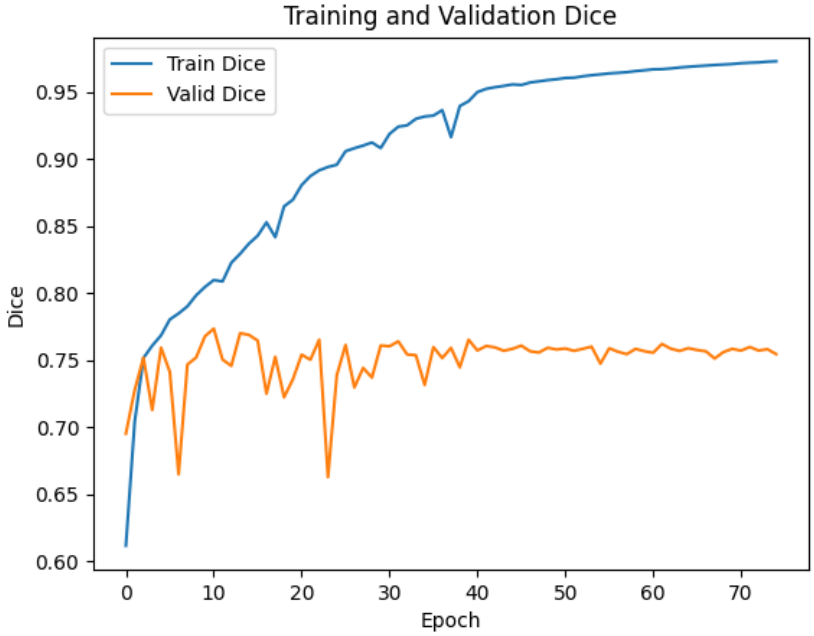

Figure 5.2: Binary *DSC* during the training without data augmentation using RGB images. The validation *DSC* is far from the training *DSC* due to overfitting.

6 | Discussion

In this work, we have developed a framework to effectively utilize HoVer-Net by identifying optimal hyperparameters and conducting a comparative analysis with various loss functions, with the aim of segmenting and classifying four types of nuclei from histopathological images: epithelial cells, lymphocytes, macrophages, and neutrophils. We obtain favorable results not only using the combination of loss functions proposed by *Graham et al.*, the authors of HoVer-Net (Section 4.4), but also by incorporating the Asymmetric Unified Focal loss (AUF) in the NP branch, achieving a performance improvement compared to the original loss functions of 0.034 in terms of mean PQ, 0.041 in terms of mean DSC, and 0.037 in terms of binary DSC, as shown in Table 5.1 and Table 5.2. Thereby, we can assert how DL is an effective solution to achieve segmentation and classification of nuclei from histopathological slides.

First, we ensure that data augmentation, especially applied with our parameters settings, is a valuable tool. In fact, using data augmentation enables the network to generalize across images and prevents overfitting, as explained in Section 4.2.1. As Figure 5.1 and Figure 5.2 show, in the first instance the network barely memorizes the training set, thus being less able to provide insights on unseen data, while in the second one the validation DSC is significantly lower than the train DSC, due to the occurrence of overfitting. Therefore, by employing data augmentation we can increase the network performance by 0.017 in terms of mPQ and by 0.060 in terms of $mDSC$. In addition, from PQ and DSC values pertaining to the four nuclei classes, we can observe how, in the absence of data augmentation, the macrophages are the most penalized class. This could be due to the combination of the relative infrequency of this class within the dataset, and the general complexity of shapes associated with the macrophages, which could incentive the network to memorize the specific instances of the cells seen during the training phase. Thus, the model fails when facing different instances of the macrophage class during the testing phase. However, we notice how the model without data augmentation performs well on the epithelial cells, suggesting how it is possible that the parameters of the data augmentation could be reconsidered with a more optimal configuration. Moreover, we train HoVer-Net both with images featuring all the RGB channels and by only using the blue channel,

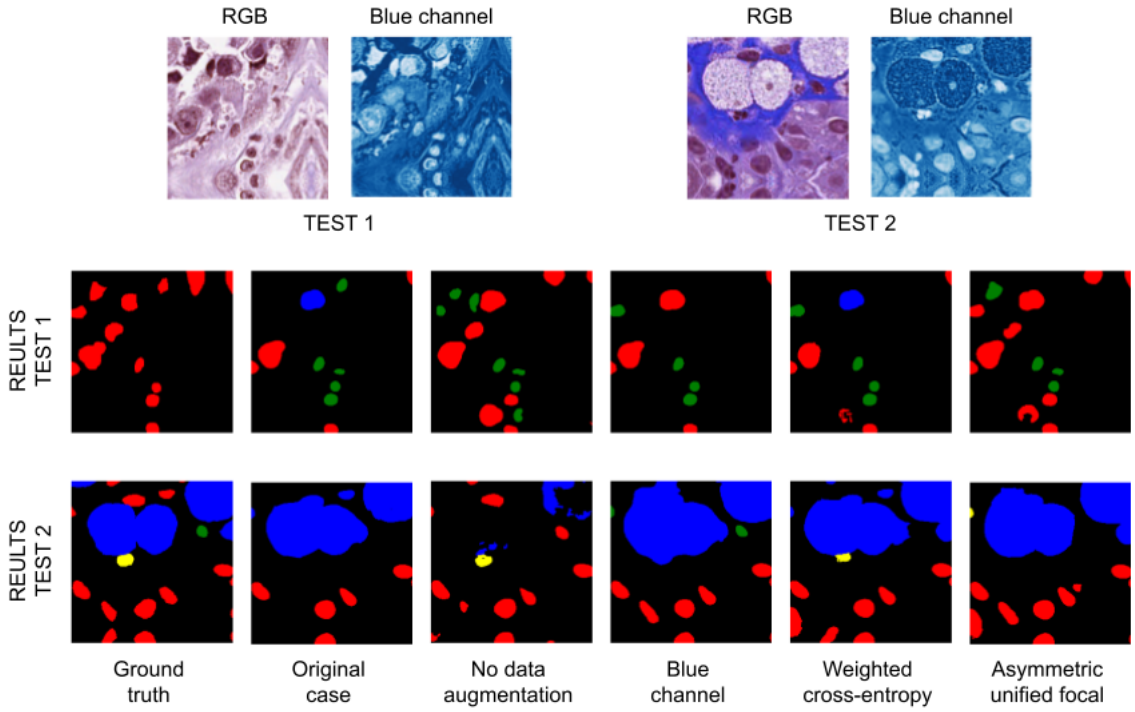

Figure 6.1: Comparison between the proposed implementations using two test patches. Black is the background, red indicates epithelial cells, green indicates lymphocytes, blue indicates macrophages, and yellow indicates neutrophils.

to show how most of the relevant information resides in the blue channel since H&E slides are typically coloured in a range of blues. The resulting metrics, as described in Table 5.1 and Table 5.2, are consistent with this hypothesis. In particular, it can be seen that the results are comparable with the base case. This occurs because, as shown in Figure 6.1, the blue channel preserves clear contours and significant features of the nuclei. This observation suggests the possibility of creating lightweight versions of H&E datasets, which would enable easier storage without significant loss of information required for accurate segmentation and classification of nuclei. By reducing the dataset's size, this lightweight version could be effectively utilized in storage-constrained environments such as Google Colab or computational setups with limited resources. This concept holds particular significance when dealing with large-scale datasets, offering practical solutions for efficient data management and analysis. Furthermore, we can notice from Table 5.1 and Table 5.2 how the network generally achieves poor results in the class corresponding to the macrophages. These results are consistent with the performances of other works in the literature [5], [62]. This occurs because, within the TME, the macrophages, called Tumour-Associated Macrophages (TAMs), are phenotypically and functionally diverse, resulting in various shapes inter- and intra-TME. Thereby, some of them promote tumour

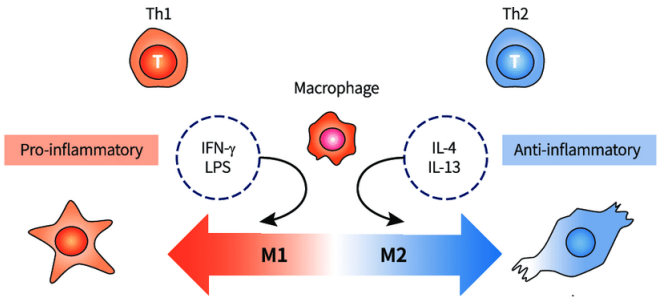

Figure 6.2: Difference of shapes between M1-like and M2-like macrophages [36].

progression whereas others exhibit anti-tumour activity [48]. More specifically, as shown in Figure 6.2, TAMs are classified into activated M1-like macrophages and alternatively activated M2-like macrophages. However, this division is only on the theoretical level, because M1-like and M2-like macrophages are two extremes in a continuum in a wide range of functional states, so the classical M1 versus M2 polarization model can be extended to a "spectrum model" [32]. For such a reason, it is hard for the model to segment correctly a macrophage, since its shape is not predefined and the characterizing features vary from time to time. As for the implementation of the new loss functions, we come up with the following conclusions. Since we deal with class imbalance, as mentioned in Section 4.1, it should be advisable to weigh every class. This should allow us to better handle the class imbalance because the correct or incorrect segmentation and classification of a nucleus which usually appears with less frequency results in more significant feedback during back-propagation. Unfortunately, by exploiting the WCCE loss we achieved poorer results compared with the original setup, but we notice an increase in $bDSC$, whose reason may lie in a greater NC loss function result, which influences all three branches and enables better segmentation. However, the results in the case where the AUF loss is employed are consistent and even better than the ones of the base case. We achieve an increase in mPQ by 0.034 and an increase in the $mDSC$ by 0.041, as well as a $bDSC$ of 0.768. We can notice how the AUF loss yields an increase in the class-based $DSCs$ and PQs for the underrepresented classes. This means that by exploiting this loss function, the network learns to better distinguish the macrophages and the neutrophils, increasing the overall performance of the network. Reproducing our experimental setup, which includes the AUF loss, within the framework proposed by *Graham et al.* [25], presents an intriguing opportunity. However, it should be noted that there are certain discrepancies between our setup and theirs. Specifically, the data pre-processing and post-processing steps employed by *Graham et al.* are more refined compared to our own approach [25].

In Figure 6.1 are shown two examples of patches to compare the results of the different

experimental setups with the respective GT. In the first example, all the models tend to confuse epithelial cells with lymphocytes, but training using AUF seems the most reliable choice to better distinguish these two classes. From the second example, we can notice the challenge in the segmentation and classification of macrophages, the class with the worst performance in all our training. It is worth mentioning how training the network in the base case and implementing the AUF are the two most effective ways to tackle this issue. In fact, according to the Table 5.1 and Table 5.2, these two models achieve the best performance when dealing with macrophages, in contrast with the model trained without data augmentation, which totally fails over this task. In general, we present the agreement between our results in terms of mPQ, $mDSC$, and $bDSC$, presented in Section 5.2 and the real segmentation and classification of the related models, highlighting the good results obtained by implementing our model exploiting the AUF loss. Moreover, we stress the effectiveness of carrying out data augmentation to better generalize across different images and classes.

7 | Conclusions and Future Work

This Chapter provides an overview of the clinical problem, and describes our objective and the best results we obtain in Section 7.1. Moreover, in Section 7.2 we discuss possible advancements of our project that could be explored in the future.

7.1. Conclusions

In conclusion, histopathological slides of the TME hold significant potential for various applications in diagnosis, prognosis, and research. By characterizing the TME, valuable insights can be gained regarding tumor initiation, development, invasion, and clinical outcomes. Hence, in this thesis, we explored the feasibility of utilizing a supervised DL algorithm to improve the accuracy, speed, and scalability of histopathological image analysis. This, in turn, has the potential to enhance patient outcomes and optimize healthcare workflows [64]. To achieve our objective, we proposed and developed a DL algorithm that focuses on the segmentation and classification of cellular nuclei from histopathological images based on the HoVer-Net architecture. For this purpose, we leveraged the MoNuSAC dataset, which provides annotated histopathological images. More specifically, we aimed to address the issue of class imbalance within this dataset. We experimented with alternative loss functions which have contributed to increasing the performance of underrepresented classes. We found that training HoVer-Net including the AUF loss within the composite loss function allowed us to obtain favorable results in terms of mPQ and $mDSC$, as shown in Table 5.1 and Table 5.2. In fact, the employment of this new loss function resulted in a better PQ and a better DSC for the most underrepresented classes, corresponding in a better mPQ and a better $mDSC$. Besides, exploiting AUF loss improves also the segmentation quality, which can be noticed from the $bDSC$. Furthermore, we found how by using only the blue channel of the H&E images we can obtain comparable results to those obtained with RGB images, thus potentially allowing for more lightweight datasets.

7.2. Future work

The results we have obtained present a promising opportunity to increase the performance of HoVer-Net-based models by using the Asymmetric Unified Focal loss. It should be noted how we applied a simplified version of the original HoVer-Net algorithm, so it would be ideal to evaluate this loss function on the complete version of the algorithm to be able to perform a more meaningful comparison with the state of the art. Furthermore, it would be ideal to assess the performance of trainings performed with blue-channel images on the complete version of HoVer-Net as well. Finally, it would be interesting to assess if other algorithms used for the same task would perform in similar ways with only blue-channel images.

Bibliography

[1] How Cancer is Diagnosed. https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis.

[2] Tumor definition. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/tumor.

[3] Known and Probable Human Carcinogens data. https://www.cancer.org/cancer/risk-prevention/understanding-cancer-risk/known-and-probable-human-carcinogens.html/.

[4] Grand Challenge MoNuSAC 2020 data. https://monusac-2020.grand-challenge.org/Data/, .

[5] MoNuSAC Supplementary Material. https://drive.google.com/file/d/1kdOl3s6uQBRv0nToSIf1dPuceZunzL4N/view, .

[6] The Cancer Genome Atlsa. https://www.cancer.gov/ccg/research/genome-sequencing/tcga.

[7] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[8] N. M. Anderson and M. C. Simon. The tumor microenvironment. *Current Biology*, 30(16):R921–R925, 2020.

[9] T. Araújo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polónia, and A. Campilho. Classification of breast cancer histology images using convolutional neural networks. *PloS one*, 12(6):e0177544, 2017.

[10] S. Badrigilan, S. Nabavi, A. A. Abin, N. Rostampour, I. Abedi, A. Shirvani, and M. Ebrahimi Moghaddam. Deep learning approaches for automated classification and segmentation of head and neck cancers and brain tumors in magnetic resonance images: a meta-analysis study. *International journal of computer assisted radiology and surgery*, 16:529–542, 2021.

[11] A. G. Berman, W. R. Orchard, M. Gehrung, and F. Markowetz. Pathml: a unified framework for whole-slide image analysis with deep learning. *medRxiv*, pages 2021–07, 2021.

[12] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat. Cnn variants for computer vision: history, architecture, application, challenges and future scope. *Electronics*, 10(20):2470, 2021.

[13] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11 (2):125, 2020.

[14] J. C. Caicedo, A. Goodman, K. W. Karhohs, B. A. Cimini, J. Ackerman, M. Haghighi, C. Heng, T. Becker, M. Doan, C. McQuin, et al. Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature methods*, 16(12):1247–1253, 2019.

[15] S. Chakraborty and T. Rahman. The difficulties in cancer treatment. *Ecancermedicalscience*, 6, 2012.

[16] G. Corredor, X. Wang, Y. Zhou, C. Lu, P. Fu, K. Syrigos, D. L. Rimm, M. Yang, E. Romero, K. A. Schalper, et al. Spatial architecture and arrangement of tumor-infiltrating lymphocytes for predicting likelihood of recurrence in early-stage non–small cell lung cancer. *Clinical cancer research*, 25(5):1526–1534, 2019.

[17] A. Das, M. S. Nair, and S. D. Peter. Computer-aided histopathological image analysis techniques for automated nuclear atypia scoring of breast cancer: a review. *Journal of digital imaging*, 33:1091–1121, 2020.

[18] A. J. Dhruv, R. Patel, and N. Doshi. Python: the most advanced programming language for computer science applications. *Science and Technology Publications, Lda*, pages 292–299, 2021.

[19] T. N. Doan, B. Song, T. T. Vuong, K. Kim, and J. T. Kwak. Sonnet: A self-guided ordinal regression neural network for segmentation and classification of nuclei in large-scale multi-tissue histology images. *IEEE Journal of Biomedical and Health Informatics*, 26(7):3218–3228, 2022.

[20] A. T. Feldman and D. Wolfe. Tissue processing and hematoxylin and eosin staining. *Histopathology: methods and protocols*, pages 31–43, 2014.

[21] A. Foucart, O. Debeir, and C. Decaestecker. Panoptic quality should be avoided as a metric for assessing cell nuclei segmentation and classification in digital pathology. *Scientific Reports*, 13(1):8614, 2023.

[22] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech. Convolutional neural network for video fire and smoke detection. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 877–882. IEEE, 2016.

[23] N. A. Giraldo, R. Sanchez-Salas, J. D. Peske, Y. Vano, E. Becht, F. Petitprez, P. Validire, A. Ingels, X. Cathelineau, W. H. Fridman, et al. The clinical role of the tme in solid cancer. *British journal of cancer*, 120(1):45–53, 2019.

[24] R. R. Gomis and S. Gawrzak. Tumor cell dormancy. *Molecular oncology*, 11(1): 62–78, 2017.

[25] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[29] C. Janiesch, P. Zschech, and K. Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021.

[30] L. M. Janssen, E. E. Ramsay, C. D. Logsdon, and W. W. Overwijk. The immune system in cancer metastasis: friend or foe? *Journal for immunotherapy of cancer*, 5: 1–14, 2017.

[31] D. N. Khalil, E. L. Smith, R. J. Brentjens, and J. D. Wolchok. The future of cancer treatment: immunomodulation, cars and combination immunotherapy. *Nature reviews Clinical oncology*, 13(5):273–290, 2016.

[32] J. Kim, J.-S. Bae, et al. Tumor-associated macrophages and neutrophils in tumor microenvironment. *Mediators of inflammation*, 2016, 2016.

[33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[35] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging*, 36(7):1550–1560, 2017.

[36] K. Y. Lee. M1 and m2 polarization of macrophages: a mini-review. *Med Biol Sci Eng*, 2(1):1–5, 2019.

[37] L. Li and M. Spratling. Data augmentation alone can improve adversarial training. *arXiv preprint arXiv:2301.09879*, 2023.

[38] P. Li, J. Li, and G. Wang. Application of convolutional neural network in natural language processing. In *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 120–122. IEEE, 2018.

[39] G. W. Lindsay. Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, 33(10):2017–2031, 2021.

[40] W. Lou, X. Wan, G. Li, X. Lou, C. Li, F. Gao, and H. Li. Structure embedded nucleus classification for histopathology images. *arXiv preprint arXiv:2302.11416*, 2023.

[41] Y. Lu, X. Qin, H. Fan, T. Lai, and Z. Li. Wbc-net: A white blood cell segmentation network based on unet++ and resnet. *Applied Soft Computing*, 101:107006, 2021.

[42] G. Lv, K. Wen, Z. Wu, X. Jin, H. An, and J. He. Nuclei r-cnn: Improve mask r-cnn for nuclei segmentation. In *2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)*, pages 357–362. IEEE, 2019.

[43] A. Y. Maslov and J. Vijg. Genome instability, cancer and aging. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1790(10):963–969, 2009.

[44] D. Masters and C. Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[45] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin,

N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[47] D. D. Patil and S. G. Deore. Medical image segmentation: a review. *International Journal of Computer Science and Mobile Computing*, 2(1):22–27, 2013.

[48] M. J. Pittet, O. Michielin, and D. Migliorini. Clinical relevance of tumour-associated macrophages. *Nature reviews Clinical oncology*, 19(6):402–421, 2022.

[49] B. M. Priego-Torres, D. Sanchez-Morillo, M. A. Fernandez-Granero, and M. Garcia-Rojo. Automatic segmentation of whole-slide h&e stained breast histopathology images using a deep convolutional neural network architecture. *Expert Systems With Applications*, 151:113387, 2020.

[50] R. Rai, A. Das, and K. G. Dhal. Nature-inspired optimization algorithms and their significance in multi-thresholding image segmentation: an inclusive review. *Evolving Systems*, 13(6):889–945, 2022.

[51] K. Ramesh, G. K. Kumar, K. Swapna, D. Datta, and S. S. Rajest. A review of medical image segmentation algorithms. *EAI Endorsed Transactions on Pervasive Health and Technology*, 7(27):e6–e6, 2021.

[52] S. E. A. Raza, L. Cheung, M. Shaban, S. Graham, D. Epstein, S. Pelengaris, M. Khan, and N. M. Rajpoot. Micro-net: A unified model for segmentation of various objects in microscopy images. *Medical image analysis*, 52:160–173, 2019.

[53] C. Roma-Rodrigues, R. Mendes, P. V. Baptista, and A. R. Fernandes. Targeting tumor microenvironment for cancer therapy. *International journal of molecular sciences*, 20(4):840, 2019.

[54] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[55] R. Sathya, A. Abraham, et al. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38, 2013.

[56] H. Seo, M. Badiei Khuzani, V. Vasudevan, C. Huang, H. Ren, R. Xiao, X. Jia, and L. Xing. Machine learning techniques for biomedical image segmentation: an overview of technical aspects and introduction to state-of-art applications. *Medical physics*, 47(5):e148–e167, 2020.

[57] K. Sirinukunwattana, J. P. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez, et al. Gland segmentation in colon histology images: The glas challenge contest. *Medical image analysis*, 35:489–502, 2017.

[58] M. Slaoui and L. Fiette. Histopathology procedures: from tissue sampling to histopathological evaluation. *Drug Safety Evaluation: Methods and Protocols*, pages 69–82, 2011.

[59] M. Titford. A short history of histopathology technique. *Journal of Histotechnology*, 29(2):99–110, 2006.

[60] A. T. Tunkiel, D. Sui, and T. Wiktorski. Impact of data pre-processing techniques on recurrent neural network performance in context of real-time drilling logs in an automated prediction framework. *Journal of Petroleum Science and Engineering*, 208:109760, 2022.

[61] M. H. L. Tveter. Exploring high dimensional, sparse reward problems using deep learning and neuroevolution. Master's thesis, 2021.

[62] R. Verma, N. Kumar, A. Patil, N. C. Kurian, S. Rane, S. Graham, Q. D. Vu, M. Zwager, S. E. A. Raza, N. Rajpoot, et al. Monusac2020: A multi-organ nuclei segmentation and classification challenge. *IEEE Transactions on Medical Imaging*, 40(12): 3413–3423, 2021.

[63] M. Weigert and U. Schmidt. Nuclei segmentation and classification in histopathology images with stardist for the conic challenge 2022. *arXiv preprint arXiv:2203.02284*, 2022.

[64] J. Wilkinson, K. F. Arnold, E. J. Murray, M. van Smeden, K. Carr, R. Sippy, M. de Kamps, A. Beam, S. Konigorski, C. Lippert, et al. Time to reality check the promises of machine learning-powered precision medicine. *The Lancet Digital Health*, 2(12):e677–e680, 2020.

[65] L. Xie, J. Qi, L. Pan, and S. Wali. Integrating deep convolutional neural networks with marker-controlled watershed for overlapping nuclei segmentation in histopathology images. *Neurocomputing*, 376:166–179, 2020.

[66] M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.

[67] Z. Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pages 1–2. Ieee, 2018.

[68] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. 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: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.