POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Master's Thesis

Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Supervisors

Prof. Massimo Salvi

Ing. Nicola Michielli

Candidate

Alessia Giustolisi

December 2024

Con un piede nel passato,
e lo sguardo dritto e aperto nel futuro

Abstract

In lymphoma diagnostics, flow cytometry is widely used for cell counting and sorting, providing essential insights into individual cell characteristics; however, it is limited by its single-cell analysis approach, which lacks spatial and morphological context. On the other hand, the manual analysis of whole-slide images (WSIs) from scanned slides is highly time-consuming and prone to both intra- and inter-operator variability. This thesis project proposes a tool for the rapid and automated delineation of cell nuclei in WSIs, with the goal of not only minimising diagnostic time but also reducing variability through the repeatable nature of automated analysis.

Traditionally, Haematoxylin and Eosin (H&E) staining is among the most used methods in this field, with extensive related instance segmentation research in literature. Although cytological stains like Giemsa and immunohistochemical (IHC) markers offer complementary information not visible in H&E, their automatic segmentation has been less explored in the literature. Giemsa staining, selected for its high-quality delineation of chromatin and nuclear membranes, proves advantageous in distinguishing various cell populations within the tissue.

This is the first work to address instance segmentation in Giemsa-stained WSIs quantitively validated. Integrating the segmentation of nuclei stained with various IHC markers enables the identification and classification of nuclei as immunopositive or immunonegative, whose presence and combinations are subsequently evaluated by the clinician to make a diagnosis.

To achieve precise instance segmentation and classification of cell nuclei in Giemsa- and IHC-stained WSIs from lymphoma patients, provided by the A.O.U. Città Della Salute e Della Scienza Hospital in Turin and publicly available images such as the Multi-Organ Nuclei Segmentation (MoNuSeg) dataset, this research employs and modifies HoVerNet, a robust deep learning model for nuclear instance segmentation. HoVerNet's modification to incorporate border-class prediction enhances its ability to separate closely positioned or overlapping nuclei, a common challenge in instance segmentation tasks. A key strength of this work lies in using a single architecture – HoVerNet – modified and optimized to manage different classification requirements for diverse segmentation goals. A comparison was also conducted with another state-of-the-art instance segmentation architecture, the Centripetal Direction Network (CDNet).

Multiple inference and post-processing strategies were explored to optimize segmentation performance, which was assessed quantitatively through pixel-based metrics (e.g., Dice Score) and object-based metrics (e.g., Aggregated Jaccard Index), as well as qualitatively through consultations with a pathologist.

Accurate segmentation of cell nuclei enables cell-by-cell analysis of morphology, texture, and colour intensity to determine marker positivity, directly informing pathologists about the tissue's pathological status. Validation against ground-truth annotations confirms that this framework not only surpasses current state-of-the-art deep learning models in accuracy but also offers a viable alternative to flow cytometry by enabling multiplexed analysis of differently stained cellular populations within their spatial context. These findings support the development of an AI-driven diagnostic tool that enhances the speed and accuracy of lymphoma diagnostics and streamlines the histopathological workflow.

Acknowledgements

I would like to express my gratitude to my supervisors, Professor Massimo Salvi and Ph.D. Nicola Michielli, for guiding me through this thesis work, making it an enjoyable experience and further igniting my passion for the topics addressed with their everyday teachings. I consider myself fortunate to have had the opportunity to work under your wise supervision.

I would also like to thank Professor Filippo Molinari, whose captivating lectures sparked my interest in image processing, leading me to choose this thesis topic, with the hope that one day I too can inspire others about these subjects in the same way.

Table of Contents

List of Tables

List of Figures

Chapter 1

Medical Introduction

1.1 Lymphomas

Lymphoma is a broad term encompassing all tumours originating in lymphoid tissue. Lymphoid tissue is composed of T, B, and NK lymphocytes and their precursors. Lymphomas are classified into Hodgkin and non-Hodgkin types, with the distinction based on the presence of Reed-Sternberg cells, which are characteristic of Hodgkin lymphoma but absent in Non-Hodgkin lymphoma [1].

The International Agency for Research on Cancer (IARC) of the World Health Organization (WHO) released a report in 2024 [2] detailing data on new cases, deaths, and 5-year prevalence for both Hodgkin and non-Hodgkin lymphomas. In Table 1.1 the key statistics are presented, specifically comparing Non-Hodgkin Lymphoma (NHL in Table 1.1) and Hodgkin Lymphoma (HL in Table 1.1). Non-Hodgkin Lymphoma has significantly higher new cases (129,338) and deaths (51,808) compared to Hodgkin lymphoma, which reports 19,723 new cases and 3,796 deaths. Non-Hodgkin Lymphoma ranks higher among cancers, with a cumulative risk of 0.93% for new cases, while Hodgkin lymphoma shows a lower cumulative risk of 0.19%. The 5-year prevalence figures indicate that Non-Hodgkin Lymphoma affects approximately 57.6 people per 100,000, while Hodgkin lymphoma affects 10.7 per 100,000, reflecting differences in survival and disease burden across both types. Overall, Non-Hodgkin Lymphoma has a higher incidence than Hodgkin Lymphoma, and a higher mortality.

Cancer Type	**New Cases**			**Deaths**			**5-Year Prev.**
	Num.	**Rank (%)**	**Cum. Risk**	**Num.**	**Rank (%)**	**Cum. Risk**	**Num. (per 100,000)**
NHL	129,338	10 (2.9)	0.93	51,808	12 (2.6)	0.27	430,454 (57.6)
HL	19,723	24 (0.44)	0.19	3,796	28 (0.19)	0.03	79,763 (10.7)

Table 1.1: Lymphoma Statistics [2]

Non-Hodgkin lymphoma can occur at all ages, making it one among the most frequent cancers affecting children, teens, and young adults. However, the likelihood of developing Non-Hodgkin lymphoma increases with age, over half of people with lymphoma diagnosis are 65 or older. Hodgkin lymphoma, by contrast, can also affect both children and adults but is most common in early adulthood, particularly in people in their 20s. The risk rises again in later adulthood, after age 55, with an overall average diagnosis age of 39. Hodgkin lymphoma seldom affects children under 5 years old, yet it remains the most diagnosed cancer in adolescents aged 15 to 19 [3].

The prevalence, mortality, and relatively young target demographic of lymphoma make prompt diagnosis essential in healthcare.

The initial diagnostic approach involves a lymph node biopsy, which is then examined in the laboratory by an expert pathologist to detect cancerous cells under the microscope. Identifying the exact type of lymphoma – whether NHL or HL and the specific subtype – is crucial for selecting the appropriate treatment [4].

1.2 Flow Cytometry

Flow cytometry is a technology used to analyse cell types from a sample – e.g., cell populations from solid tissues as lymph nodes, solid tumors, fluid tissues as blood and bone marrow [5] – characterising them both quantitatively and qualitatively, or sorting them out for further analysis. The basis of this technology lies in the interaction between individual cells and light produced by one or more external lasers [5, 6].

The operating principle of a flow cytometer can be summarised as follows: a cell suspension in buffered saline solution is transported by a fluidic system maintaining a constant flow to the measurement point, where it encounters a focused light beam (laser). When the light beams intercept the cell stream, signals are generated from the interaction with each individual cell. Signals are typically collected by photomultimeter tubes and photodiodes [5, 6, 7]. The interaction between the cell and laser(s) can either generate visible light scattering or induce fluorescence phenomena (if the cells are transfected to express fluorescent proteins, or labelled with fluorescence-tagged antibodies, or stained with dyes) [5, 6]. These events – light scattering and fluorescence – are independent [5].

Although scattering occurs in all directions, measurement is taken in only two: the forward direction (Forward Scatter) and at 90° (Side Scatter); the former provides information on the relative size of the cell, while the latter on the granularity/internal complexity of the cell [5].
The fluorescent light is steered by dichroic filters into detectors; band-pass filters are employed to separate the wavelengths of interest, in order to isolate signals

from individual fluorochromes [5].

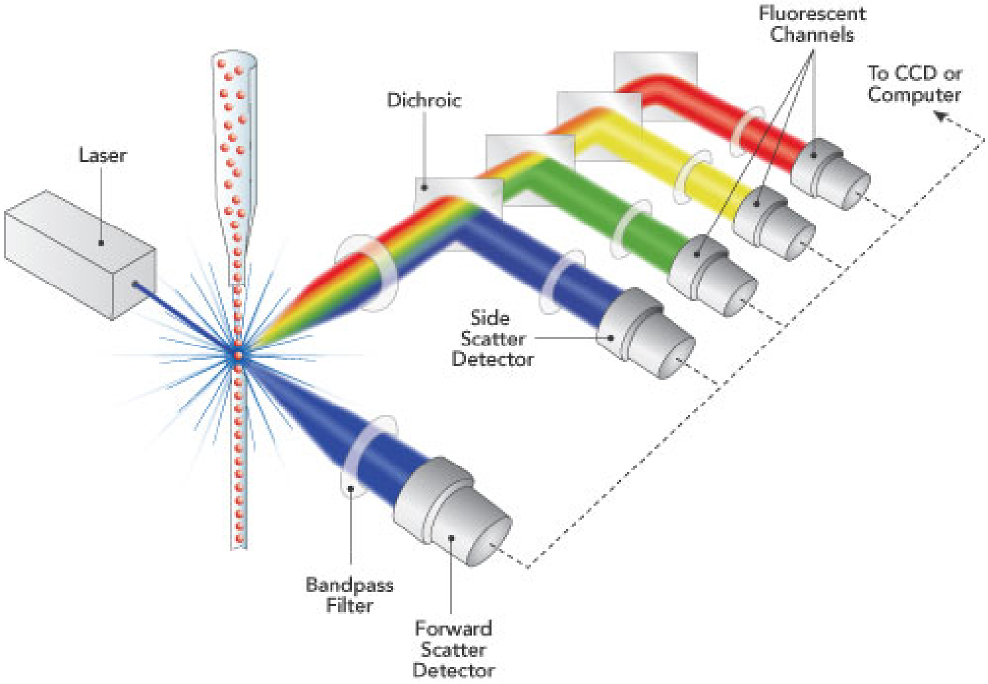


Figure 1.1: Flow Cytometer

Over time, not only has the instrumentation evolved, but so has the collection of dyes, fluorescent proteins and fluorochromes, greatly increasing the number of parameters that can be analysed. The data analysis itself has undergone significant evolution: traditionally, two-parameter gating has been used, which involves drawing dot plots based on the signals from two chosen channels; nowadays the growth in complexity has lead to the use of more advanced data mining methods and clustering algorithms (e.g., principal component analysis) to assess data with a multidimensional approach [5, 6].

Flow cytometry can be employed across a wide range of disciplines, with immunophenotyping of cells being particularly common because of the subsequent diagnostic purposes, such as identifying specific marker patterns for certain types of leukaemia and lymphoma and perform differential diagnosis [6, 8]. Starting from a mixed population of cells, it is possible to isolate the various subpopulations present in the sample by simultaneously analysing a broad range of parameters. Specifically, immune cells are typically labelled with fluorochrome-conjugated antibodies in order to target antigens of interest present on the cell surface. Among these markers, the "clusters of differentiation" (CDs) are noteworthy due to their importance and abundance; these are specific markers respectively for T cells, B cells, NK cells, and monocytes [5, 6]. Immune cells can be subdivided into subgroups based on the presence of these CDs as well as other markers (see Section 1.4) and the division used to make precise diagnosis [8].

1.3 Digital Pathology

1.3.1 Anatomical Pathology

Anatomical pathology is a speciality within the broader field of pathology, which is the study of diseases. What distinguishes anatomical pathology from clinical pathology is the laboratory analysis of samples obtained through biopsies of solid or fluid tissues from the patient. The examination of these specimens is aimed at diagnosing a potential disease or providing a prognosis for an already established condition.

Among the procedures of interest for the aims of this Master Thesis project there is the microscopic analysis of tissue sections in the form of histological slides. To enhance the visualisation of cells and tissue components staining is used. The pathologist can choose whether to stain preferential components, such as the cell wall, nucleus or cytoplasm. The choice of staining is directly related to what and which characteristics the pathologist, using their expertise, determines to be of interest for the case at hand. The main stages of staining are the following – but may vary with reference to the particular procedure [9]:

- Permeabilisation: A mild surfactant is used to dissolve the cell membranes, allowing the dye molecules to enter the cell.
- Fixation: Preserving the tissue morphology under analysis is essential. The use of chemical fixatives helps to stabilise the chemical bonds between proteins.
- Mounting: The sample is adhered to a slide for microscopic analysis or to be scanned.
- Staining: The sample mounted on the slide is exposed to a solution capable of colouring it in a specific manner. Subsequently, the slide is washed to remove dye in excess and is now ready for observation.

The use of flow cytometry for cell immunophenotyping is also among the possible procedures of anatomical pathology; the fundamental difference between histopathological slide analysis and flow cytometry lies in the nature of the two methods: as mentioned in Section 1.2, flow cytometry information is cell-based as single cells are separed from a sample suspended in solution, resulting in the loss of spatial coherence, meanwhile slide analysis retains the sample morphology.

1.3.2 Different kinds of staining

In detail, the procedures in anatomical pathology include: histopathology, cytopathology, and immunohistochemistry (IHC).

Histopathology refers to the microscopic examination of the morphology of tissues fixed on slides using histological techniques; it differs from cytopathology, which generally examines samples of free cells or tissue fragments from a more functional point of view than histopathology.

IHC involves the staining of tissues by exploiting the selective binding between antibodies and antigens present in the cells. These staining techniques allow for the identification of more functional characteristics rather than morphological ones, and as such, are widely employed for the differential diagnosis of tumours or cellular abnormalities.

These three broad ensembles of procedures differ both in the staining techniques used and in the information they provide to the pathologist regarding the tissue under study.

Haematoxylin & Eosin Staining [10]

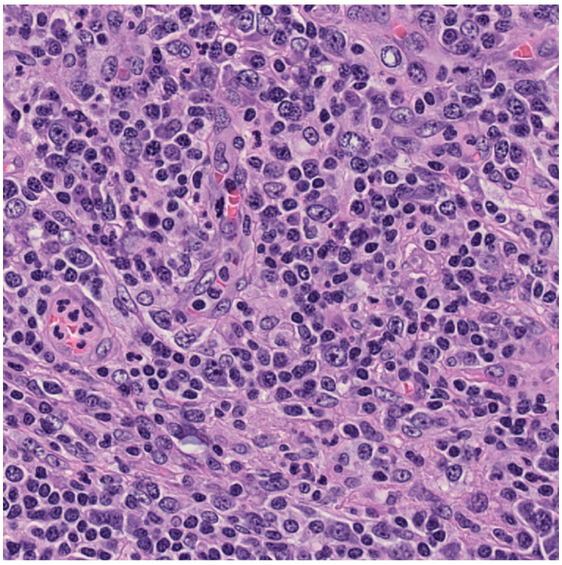


Figure 1.2: H&E stained tissue from Our Private Dataset

The gold standard histopathological stain is haematoxylin and eosin (H&E); haematoxylin is a dye that preferentially binds to acidic components (i.e. nucleic acids), thus staining the cell nucleus, which is visualised in blue-purple. Eosin, on the other hand, preferentially binds to higher pH (basic) substances, staining proteins, cytoplasm and extracellular membrane in pink. Morphological information predominates over functional information in this technique, allowing an expert pathologist to differentiate cellular components, study cellular patterns, and examine the spatial arrangement of cells in relation to one another.

Giemsa Staining [11]

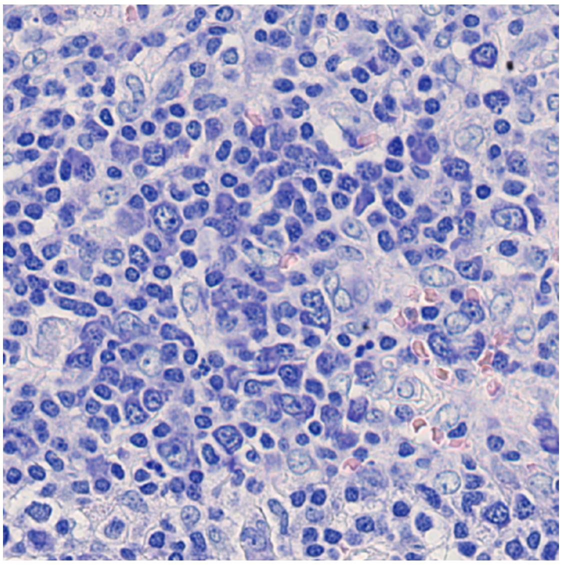


Figure 1.3: Giemsa stained tissue from Our Private Dataset

Giemsa staining is typically used in cytopathology for the study of free cells, but it is also applied to the examination of whole tissue slides. It is produced from a mixture of methylene blue, eosin, and Azure B. The range of conditions that can be studied using this staining is broad, spanning from infectious diseases and inflammatory conditions to tumour diagnosis - and was primarly designed to study parasites in malaria.

Its use is particularly notable in onco-haematology, as Giemsa allows to differentiate various types of granulocytes and reveals cytoplasmic properties of lymphoid and myeloid cells. The DNA of cells is stained blue-violet, and the granularity of eosinophils, basophils, and mast cells is highlighted in various shades of blue. Additionally, it provides high-quality staining of chromatin and the nuclear membrane.

Giemsa staining of lymph node tissue is considered superior to H&E; although nuclear staining with Giemsa may be less clear than with H&E, Giemsa's ability to differentiate between granules proves advantageous. Giemsa analyses are morphological and can be effectively paired with IHC's functional analyses.

DAB Staining [12]

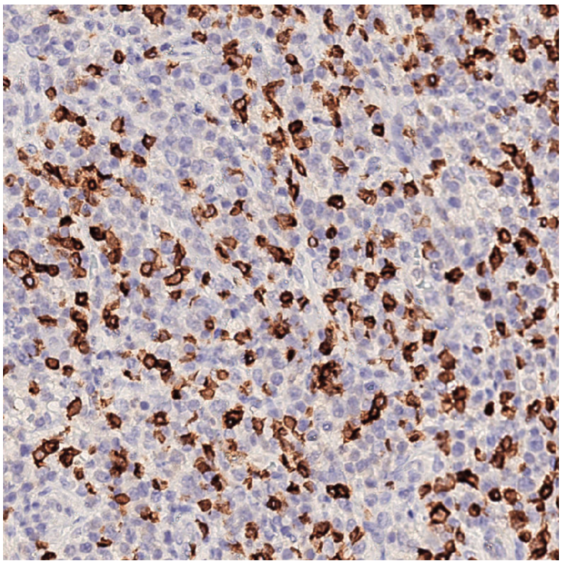


Figure 1.4: DAB stained tissue from Our Private Dataset

The quintessential IHC staining is 3,3'-diaminobenzidine (DAB). The reaction between the antigen and antibody causes a brown staining to appear; a second staining (counterstaining), such as haematoxylin, is used to colour the background – areas where the antigen/antibody reaction has not occurred – in blue. This allows for the naked eye identification of areas where the reaction has taken place (e.g., a tissue is said to be positive for a marker, such as a CD) based on whether the staining appears brown or blue.

If the concentration of DAB is low, indicating a low concentration of antigen/antibody bonds, it becomes challenging to clearly identify the areas, leading to intra- and inter-observer variations in slide interpretation. For this reason, colour deconvolution techniques [13], algorithms and statistical methods can help minimise variability in interpretation.

1.3.3 Why digitalise anatomical pathology?

"Digital pathology includes the acquisition, management, sharing and interpretation of pathology information, including slides and data, in a digital environment" [14].

The shift from traditional anatomical pathology to digital pathology is driven by the convergence of technological advancements and the need to provide increasingly accurate diagnoses, particularly in light of current projections that cancer cases are expected to increase by 60% by 2040 [15]. The process of producing diagnoses is complex and analytical, and the use of physical slides is constrained by their very nature: they are perishable, must be physically transported to be shared or transmitted, and comparisons between slides require significant effort, to name a few examples.

To simplify and enhance the already complex work of pathologists, digital pathology aims to digitalize slides using high-resolution scanners. This creates whole slide images (WSI), i.e., high-magnification images (x20, x40, x60 or more) of the slide content, to be visualized on mobile device and computer screens.

This significantly simplifies the transmission, storage, and retrieval of images for research or clinical consultation purposes, such as viewing the patient's clinical history, comparing areas within a slide or differently stained slices of a specimen from the same patient, asking second opinions and so on.

It is also paramount, particularly in the context of this Master Thesis, to highlight how digital pathology has paved the way not only for telepathology but also for the employment of AI and deep learning in anatomical pathology. These technologies are employed both to develop tools and systems that support the pathologist's diagnostic work – by accelerating and corroborating decisions, reducing inter – operator variability, and generally making the process more efficient – and to achieve broader research objectives aimed at understanding the mechanisms underlying cellular behaviour in pathological conditions.

1.4 Lymphoma Differential Diagnosis

1.4.1 Digital Pathology for Lymphoma Diagnosis [8]

In the context of differential diagnosis of lymphomas, IHC enables the identification of the cell maturation stage, lineage, genetic alterations, and proliferation levels. Various markers can be used for IHC analysis, such as different CDs, Ki-67, and others. Positivity or negativity for a specific marker indicates the existence or absence of certain proteins on the wall cytoplasm or nucleus of the cell.

The advantage of this type of analysis, compared to flow cytometry, lies in its lower cost, shorter processing times, and the ability to perform it on whole tissue samples rather than at the single-cell level. Indeed, slides retain spatial information regarding the arrangement of cells within the tissue.

There are recommendations for the use of markers based on the most recent classification of haematopoietic and lymphoid tumours by the WHO [16].

Recommended Panels for DLBCL and its variants

Diffuse large B-cell lymphoma (DLBCL) can be not otherwise specified (NOS) or present in various subtypes.

1. Diffuse Large B-Cell Lymphoma (DLBCL)
    - CD3: Negative, to exclude T-cell lymphomas.
    - CD20: Positive, indicates B lymphocytes.
    - Ki-67: >80%, indicates high proliferation.
    - Bcl-2: Positive (cut-off: 50%), used to verify double-expressor phenotype.
    - c-myc: Positive (cut-off: 40%), used to verify double-expressor phenotype.
    - CD10, Bcl-6, MUM-1: Used for cell of origin (COO) classification following the Hans algorithm [17]:
        - GCB (Germinal Center B-cell): CD10+ (≥30%) or Bcl-6+ (≥30%) and MUM-1- (<30%).
        - ABC (Activated B-cell): MUM-1+ (≥30%), often CD10- and Bcl-6+.
2. Burkitt's Lymphoma
    - CD10: Positive.
    - Bcl-2: Negative.
    - Ki-67: >95%, very high proliferation.
    - MYC: Detection of rearrangements via FISH.
3. Double Hit Lymphoma (High-grade B-cell Lymphoma with MYC and BCL2 and/or BCL6 rearrangements)
    - FISH for MYC, BCL2, BCL6: To confirm genetic rearrangements.
    - c-myc IHC: Can be used as a screening tool but does not replace FISH.
4. Epstein-Barr Virus (EBV)-Positive DLBCL
    - EBV ISH: Positive in >80% of tumor cells, indicates EBV-positive DLBCL.
    - CD20: Positive, confirms B-cell lymphoma.
5. CD20-Negative DLBCL
    - CD79a: Used to confirm B-cell lineage in patients treated with rituximab.
    - PAX5: B-cell marker, weak or absent in CD20-negative forms.
6. Plasmablastic Lymphoma (PBL)
    - CD138: Positive, confirms plasmacytic differentiation.
    - EBV ISH: Can be used to confirm EBV association.

- ALK: Negative, to exclude ALK+ lymphoma.

7. Anaplastic Lymphoma Kinase (ALK)-Positive Large B-Cell Lymphoma
    - ALK: Positive, confirms diagnosis.
    - CD20, CD79a, PAX5: Negative or weakly positive.
8. Primary Mediastinal (Thymic) Large B-Cell Lymphoma (PMLBL)
    - CD30: Positive ($\simeq$80% of cases).
    - CD23: Positive ($\simeq$70% of cases).
    - Tumor location and histological features: Help in the differential diagnosis from DLBCL NOS.

Essential Markers	**Optional Markers**
CD3	CD79a
CD20	CD30
Ki-67	CD23
CD10	ALK
BCL-6	CD138
MUM1	
BCL-2	
c-myc	
EBV ISH	

Table 1.2: Essential and Optional Markers for DLBCL Analysis

Recommended Panels for Small/Low Grade B-Cell Lymphomas

Classification of low-grade small B-cell lymphomas and accurate identification of subtypes:

1. Follicular Lymphoma (FL)
    - CD10: Positive, associated with germinal centers (centroblasts/centrocytes). However, it may be negative in higher-grade lymphomas.
    - Bcl-2: Positive, caused by the BCL2 gene rearrangement (pathognomonic). Normally, germinal centers are negative.
    - Ki-67: <20% in grades 1 and 2, >30% in grades 3A and 3B.
    - LMO2, GCET1, HGAL: Alternative markers, but not recommended for routine use.

2. Chronic Lymphocytic Leukemia/Small Lymphocytic Lymphoma (CLL/SLL)
    - CD3: Negative, excludes T-cell lymphomas.
    - CD20: Positive, confirms B-cell lineage.
    - CD5: Positive, if diffusely expressed (originally a T-cell marker).
    - CD23: Positive, co-expressed with CD5 (sensitive for CLL/SLL, but not specific).
3. Mantle Cell Lymphoma (MCL)
    - CD5: Positive, although shared with CLL/SLL.
    - Cyclin D-1: Positive because of CCND1 gene rearrangement (specific for MCL).
    - SOX11: Positive, used as a primary marker in cyclin D-1 negative MCL.
    - Ki-67: May vary, but high in blastoid and pleomorphic variants.
4. Marginal Zone Lymphoma (MZL)
    - CD10, CD5, CD23, Cyclin D-1, SOX11: All negative. MZL is often a diagnosis by exclusion when other small B-cell lymphomas have been ruled out.
    - Kappa and Lambda: Used to confirm plasma cell monoclonality in cases with plasmacytic differentiation.
    - Immunoglobulins (IgG, IgA, IgM): Used to differentiate MZL from lymphoplasmacytic lymphomas (LPL).
5. Specific Differential Diagnosis
    - Plasmablastic Lymphoma (PBL): In cases with plasmacytic differentiation, CD138 is used to confirm this.
    - Epstein-Barr Virus (EBV): Assessed using EBV ISH to confirm association with EBV in lymphomas linked to viral infection.

Essential Markers	Optional Markers
CD3	CD5
CD20	Cyclin D-1
Ki-67	SOX11
Bcl-2	CD23
CD10	CD21
	CD138
	Kappa
	Lambda
	CD79a
	IgD
	IgG
	IgM

Table 1.3: Essential and Optional Markers for Low-Grade B-cell Lymphomas

Recommended Panels for T/NK-Cell Lymphomas

There are many different types of T/NK-cell lymphoma according to the current WHO classification [16]:

1. Anaplastic Large Cell Lymphoma (ALCL)
    - CD3: Usually negative, despite ALCL being a T-cell lymphoma.
    - CD30: Strongly and diffusely positive, characteristic of ALCL.
    - ALK: Divides ALCL into ALK-positive or ALK-negative based on the presence of ALK.
    - CD2, CD5, CD7, TIA-1, Granzyme B: Used to confirm T-cell origin, especially in ALK-negative cases.
2. Angioimmunoblastic T-Cell Lymphoma (AITL)
    - CD3: Positive, confirms T-cell lymphoma.
    - CD4: Positive, confirms follicular T-helper (TFH) subtype.
    - CD8: Negative.
    - PD-1, CXCL13, CD10, Bcl-6, ICOS: TFH markers used to support the diagnosis, although they have low sensitivity and specificity.
    - CD21: Marker of follicular dendritic cells, used to identify the typical expansion pattern of the FDC network.
3. Extranodal NK/T-Cell Lymphoma, Nasal Type (ENKTL)

- EBV ISH: Positive, as tumor cells are EBV-positive in nearly all cases.
- CD3: Positive (often weakly).
- CD56: May or may not be expressed, depending on the cell lineage.
- EBV ISH–CD3 and EBV ISH–CD20/CD79a: Double markers used to determine tEBV-positive cells' lineage (T or B).

4. Peripheral T-Cell Lymphoma, Not Otherwise Specified (PTCL NOS)
    - CD3: Positive, highlights cytological atypia of lymphocytes.
    - CD4/CD8: Alternately expressed, but both may occasionally be negative.
    - CD56: Variable expression (depends on PTCL type).
    - β-F1, TCR delta: Used to distinguish $\alpha\beta$ T lymphocytes from $\gamma\delta$.
5. Specific Differential Diagnosis
    - CD30: Often positive in several T-cell lymphomas, and therefore also used as a therapeutic target.

Essential Markers	**Optional Markers**
CD3	CD4
CD20	CD8
Ki-67	CD56
EBV ISH	CD30
	ALK
	TIA-1
	Granzyme B
	PD-1
	CXCL13
	CD10
	Bcl-6
	ICOS
	CD2
	CD5
	CD7
	CD21
	β-F1
	TCR-delta

Table 1.4: Essential and Optional Markers for T/NK-Cell Lymphomas

Recommended Panels For Precursor Lymphomas

The main immunohistochemical (IHC) markers used to distinguish between T-lymphoblastic leukemia/lymphoma (T-LBL) and B-lymphoblastic leukemia/lymphoma (B-LBL):

1. General Diagnosis for Precursor Lymphoid Neoplasia
    - TdT (Terminal Deoxynucleotidyl Transferase): Positive, a marker of lymphoid precursor cells.
    - CD99: Positive, providing further confirmation of precursor lymphoid neoplasia.
2. Distinction Between T-LBL and B-LBL
    - T-LBL:
        - CD3: Positive, confirms T-cell lineage.
    - B-LBL:
        - CD20: Can be negative, thus not reliable for confirming B-cell lineage.
        - CD79a, PAX5, CD19: Used to confirm B-cell lineage in the absence of CD20 expression.
3. Other Supporting Markers
    - Ki-67: Used to assess the proliferation index.
4. Typical Diagnostic Protocol
    - In practice, initial staining for CD3, CD20, and Ki-67 is performed. If the results suggest the possibility of a precursor lymphoid neoplasm, further staining for TdT and CD99 is carried out, followed by additional lineage-specific markers for T or B cells if both CD3 and CD20 are negative.

These markers help distinguish between T-LBL and B-LBL, as they are morphologically difficult to differentiate using only H&E staining.

Essential Markers	**Optional Markers**
CD3	CD10
CD20	CD19
CD79a	PAX5
Ki-67	CD5
TdT	
CD99	

Table 1.5: Essential and Optional Markers for T-LBL and B-LBL Diagnosis

Recommended Panels for Hodgkin Lymphomas

The main markers used for the diagnosis of Hodgkin lymphoma, distinguishing between classic Hodgkin lymphoma (cHL) and nodular lymphocyte-predominant Hodgkin lymphoma (NLPHL):

1. Classic Hodgkin Lymphoma (cHL)
   - CD30: Positive, characteristic in Hodgkin-Reed-Sternberg (HRS) cells with membranous and Golgi patterns; highly sensitive.
   - CD15: Positive, but less sensitive than CD30. It is necessary to distinguish positivity in small granulocytes present in tumors.
   - CD20: Negative, used to exclude lymphocyte-predominant lymphoma.
   - PAX5: Weakly positive, indicating the B-cell nature of HRS cells.
   - EBV ISH: Positivity is variable and depends on cHL subtypes, but does not influence the diagnosis.
2. Nodular Lymphocyte-Predominant Hodgkin Lymphoma (NLPHL)
   - CD20: Positive, confirming the B-cell origin of tumor cells.
   - CD30: Negative, allowing differentiation of NLPHL from cHL.
   - Oct-2 and BOB1: Positive, with Oct-2 strongly positive and BOB1 moderately positive.
   - EMA (Epithelial Membrane Antigen): Sometimes positive, useful in diagnostic support.
3. Differentiation Between NLPHL and THRLBL
   - CD20: Positive for atypical large tumor cells in both NLPHL and THRLBL.
   - Nodular pattern of background B cells: Indicative of NLPHL, contrasting with THRLBL.

Essential Markers	**Optional Markers**
CD3	Oct-2
CD20	BOB1
Ki-67	EMA
CD30	
CD15	
PAX5	
EBV ISH	

Table 1.6: Essential and Optional Markers for Hodgkin Lymphoma Diagnosis

Chapter 2

Technical Introduction

2.1 Deep Learning and CNNs

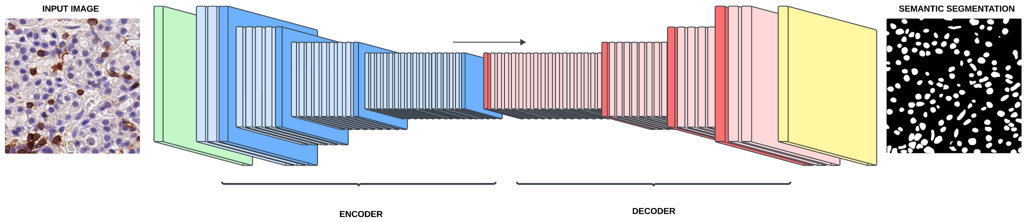


Figure 2.1: Convolutional Neural Network for Semantic Segmentation

When discussing deep learning as opposed to traditional machine learning, we refer to AI techniques that leverage deep neural networks with multiple layers to indipendetly learn complex patterns from wide datasets. In image analysis, a key tool in deep learning is the convolutional neural network (CNN), which uses convolutional layers to detect features, max pooling to reduce dimensionality while preserving essential information, and ReLU (Rectified Linear Unit) activation functions to introduce non-linearity into the model. For segmentation tasks, which involve pixel-by-pixel classification, the simplest architectures consist of an encoder and a decoder. The encoder's aim is the extraction high-level features from the images fed as input, whereas the decoder reconstructs the segmented output using operations such as transposed convolutions. When talking about segmentation a distinction between semantic segmentation and instance segmentation can be made.

Semantic segmentation's goal is pixel-wise classification, without discriminating between individual instances of the same class. Instance segmentation, on the other hand, identifies each distinct object instance, making it ideal for tasks requiring separation of closely grouped elements, such as cell nuclei. In instance segmentation, it is also possible to incorporate instance classification tasks, adding a layer of information to identify each segmented instance by type. Over the years, these architectures have evolved significantly to meet the demands of highly specialized tasks, such as nuclear instance segmentation, adapting to capture the unique complexities of such applications.

2.2 Instance Segmentation with Deep Learning

Although flow cytometry is widely used for cell counting and sorting, it is limited by its single-cell analysis approach, which lacks spatial and morphological context. To recover spatial information, pathology has eventually relied on the manual analysis of WSIs. However, analysing and segmenting cell nuclei in scanned slides is highly time-consuming and subject to intra- and inter-operator variability.

The appearance of deep learning for instance segmentation and classification has enabled the development of specialized architectures for pathology applications, providing a significant boost to digital pathology. This approach to automated segmentation helps reduce intra- and inter-operator variability and shortens analysis times, facilitating quicker diagnoses, grading, and prognostic assessments.

However, this approach has limitations, primarily the need for sufficiently large, expertly annotated datasets to train the network. Additionally, there is a need to develop architectures specifically aimed at addressing the main challenges of instance segmentation. These challenges include the existence of fused, overlapping, or adjacent nuclei, which non-specialized models or algorithms may fail to distinguish, leading to merged segmentation of multiple nuclei as a single instance. Further issues include the risks of over- or under-segmentation of objects, missing the segmentation of existing objects, or the 'ghost' segmentation of non-existent objects not included in the ground truth.

2.3 State of Art

Traditionally, the most commonly segmented stain in the literature is H&E, followed by various IHC stains. As of the development of this master's thesis, there are no advanced deep learning models applied to Giemsa staining for instance segmentation.

One technique used to enhance instance recognition is the addition of a border class, proposed in 2017 by N. Kumar et al. [18]. In their work, the dataset used was H&E-stained. The distinction between the inner region and the cell border, in addition to the background, improves cell separation. A Convolutional Neural Network (CNN) is typically employed for this purpose, taking advantage of large datasets. In 2019, P. Naylor et al. [19] proposed enhancing CNN performance by employing deep regression on the intra-nuclear distance map to separate joint nuclei and refine segmentation accuracy. In the same year, Q. Kang et al. [20] proposed a two-step framework for cell nuclei segmentation, transforming the binary segmentation task into a three-class task by introducing border classification as an intermediate step. This approach uses two cascaded U-Nets to improve segmentation accuracy. The use of marker-initialized watershed for instance segmentation can be found in the work proposed in 2020 by L. Xie et al. [21]. This approach enhanced the ability to separate touching nuclei during the post-processing of outputs from a DIMAN Network. In 2021, X. Liu et al. introduced MDC-net [22], a CNN featuring multi-scale residual connections and distance maps to enhance segmentation, especially for touching nuclei. In 2022, A. Mahbod et al. [23] proposed a U-Net-based framework with dual decoders for H&E segmentation. This approach segments distance maps and binary masks separately, with the outputs used in a watershed process for instance segmentation. An additional branch can be integrated to perform classification on the identified instances.

The models used in this thesis, HoVerNet by S. Graham et al. [24] and CDNet by H. He et al. [25], were developed in 2019 and 2021, respectively.

HoVerNet is an innovative network featuring an encoder and three distinct decoder branches: one for cell nucleus segmentation, one for predicting horizontal and vertical maps, and an optional third branch for instance classification. The outputs of the first two branches are employed in an original post-processing step to initialise a watershed.

CDNet introduces the novel concept of centripetal direction features for distinguishing pixels between instances, implementing a direction-guided refinement module to enhance segmentation accuracy. Tests revealed that CDNet excels over current methods on MoNuSeg dataset, setting a new standard in nuclear instance segmentation.

The articles listed so far consistently focus on segmenting nuclei in H&E-stained images.

In 2024, H. Naji et al. [26] developed Ho-Ly Net as an updated model of HoVerNet, designed for the segmentation of histological images stained with both H&E and IHC from DLBCL samples. This model performs both instance segmentation and instance classification, utilizing a new dataset that provides both instance maps and type maps for each staining.

2.4 Aim of this Master Thesis

The aim of this thesis is to pave the way for a virtual multiplex tool for the visualisation, segmentation, and classification of cells in WSIs stained with various markers, to assist pathologists in making faster and more accurate differential lymphoma diagnoses. To achieve this, the same network, HoVerNet [24], was trained on different datasets of tiles extracted from WSIs with diverse stains, in an effort to identify a single architecture capable of both segmentation and, where applicable, classification of cells. The HoVerNet architecture was also modified to address specific challenges of nuclear instance segmentation, such as overlapping nuclei and closely packed cellular structures.

Additionally, the Centripetal Direction Network (CDNet) [25] was employed to provide a comparative analysis of performance. Transfer learning was used to overcome the lack of annotations in certain datasets obtained from hospital sources, thus enabling the model to be effectively applied to new, unlabelled data for instance segmentation and classification.

This framework sets the groundwork for an AI-driven tool that not only reduces diagnostic variability and time but also expands the applicability of automated analysis to Giemsa- and IHC-stained slides, supporting pathologists in the differential diagnosis and grading of lymphomas.

Chapter 3

Instance Segmentation of Nuclei

3.1 Datasets

Hereby a short description for each dataset used to train the model.

3.1.1 MoNuSeg

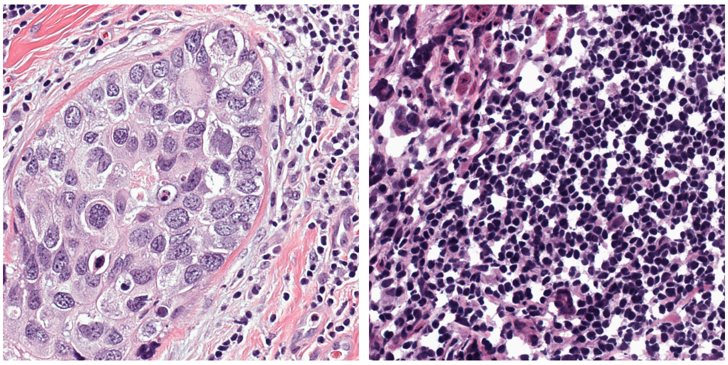


Figure 3.1: Samples from TEST 1 and TEST 2 of MoNuSeg Tiles

The dataset known as MoNuSeg 2018 (Multi-Organ Nucleus Segmentation) was introduced in [18, 27]. It features 30 tiles, sized 1000×1000, extracted from The Cancer Genome Atlas Program (TCGA) WSIs at 40x magnification, stained with H&E. The organs represented are varied (prostate, breast, liver, brain, colon, stomach, bladder) and contain both pathological (tumours at different stages) and non-pathological tissues. Epithelial and stromal nuclei were annotated manually.

Overlapping nuclei were handled by assigning pixels that belonged to multiple nuclei to the most visually superficial structure. These annotations were performed by engineering students; the masks are provided as XML files with the nuclei boundary coordinates rather than binary masks, which is useful in cases where nuclear borders are used for three-class segmentation objectives (background, border, inner).

For this study, the dataset was split so that 13 tiles were in the training set, 3 in the validation set, and the remaining 14 in the test set, which was further divided into two subgroups: TEST 1 and TEST 2. TEST 2 contains structures that the network did not encounter during training (brain and lung).

3.1.2 Giemsa Dataset

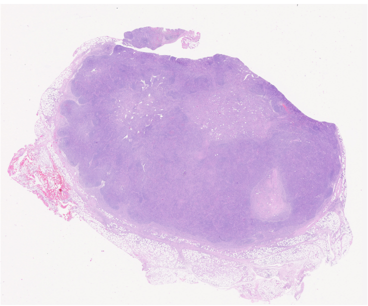


Figure 3.2: H&E

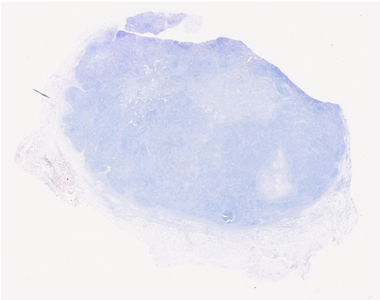


Figure 3.3: Giemsa

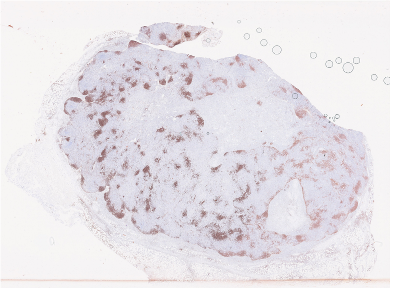


Figure 3.4: CD20

Giemsa Dataset was provided by A.O.U. Città Della Salute e Della Scienza Hospital in Turin. It includes various WSIs extracted from 13 patients' biopsies with a lymphoma diagnosis. For each patient, multiple consecutive slides were extracted and stained differently. For each patient, there is an H&E-stained section and a section stained with Giemsa, followed by multiple sections marked with IHC stains. In Figure 3.2, 3.3, 3.4 are showed three consecutive slices from the same patient. A complete list of the IHC markers present in this dataset is provided in Table 3.1, dividing markers by stained site (nucleus, cytoplasm, or cell membrane). Some of these markers can stain more than one site and are referred to as "mixed", while others have a membrane accentuation (*). This classification was made according to [28].

Markers		
Nucleus (N)	**Cytoplasm (C)**	**Membrane (M)**
BCL1	BCL2	CD2
BCL2	CD4	CD3
BCL6	CD5	CD4*
CD138	CD21-3	CD5
Ki67	CD23	CD7
P53	CD30	CD8
SOX11	CD68	CD10 (CALLA)
CMV	CD79	CD15
CICLIN	CD138	CD20
HHV8	AE1-AE	CD21-3
MNDA	ATTR	CD23
PAX5	CXCL13	CD30
C-MYC	EBER	CD57
MUM1	EMA	CD79
OCT2	GRANZ	IgG
	TIA1	IgG4
Mixed: BCL2(N,C), CD4(C,M*), CD5(C,M), CD21-3(C,M), CD23(C,M), CD30(C,M), CD79(C,M), CD138(C,M), AE1-AE(C,M,N), EBER(C,M), EMA(C,M), KAPPA(C,M), LAMBDA(C,M), PERFOR(C,M)		
(*) has membrane accentuation		

Table 3.1: Giemsa Dataset's IHC Markers

Most of the markers used are among those listed in Section 1.4.1 for the differential diagnosis of lymphomas through digital pathology; only two (TdT, CD99) are "essential" missing markers (for the detection of lymphoma precursors), and another six are missing optional markers.

The slides were scanned with a Hamamatsu scanner to create WSIs with resolution of 0.2208 μm per pixel, with 40x magnification and saved in the pyramid format NanoZoomer Digital Pathology Image (.ndpi), each occupying approximately 3 GB. For network training, a variable number of regions of interest (ROIs) were extracted from the Giemsa-stained section of each patient, creating a collection of 73 1500×1500 ROIs that appropriately represented the variability of structures and staining tones present in the dataset, in order to expose the network to a wide range of examples. The nuclei annotations were performed by this Thesis' author using QuPath software and supervised by an expert pathologist; the format of extraction is '.geojson', which contains the coordinates of the pixels belonging to each instance. This permits to extract both instance maps and boundary coordinates.

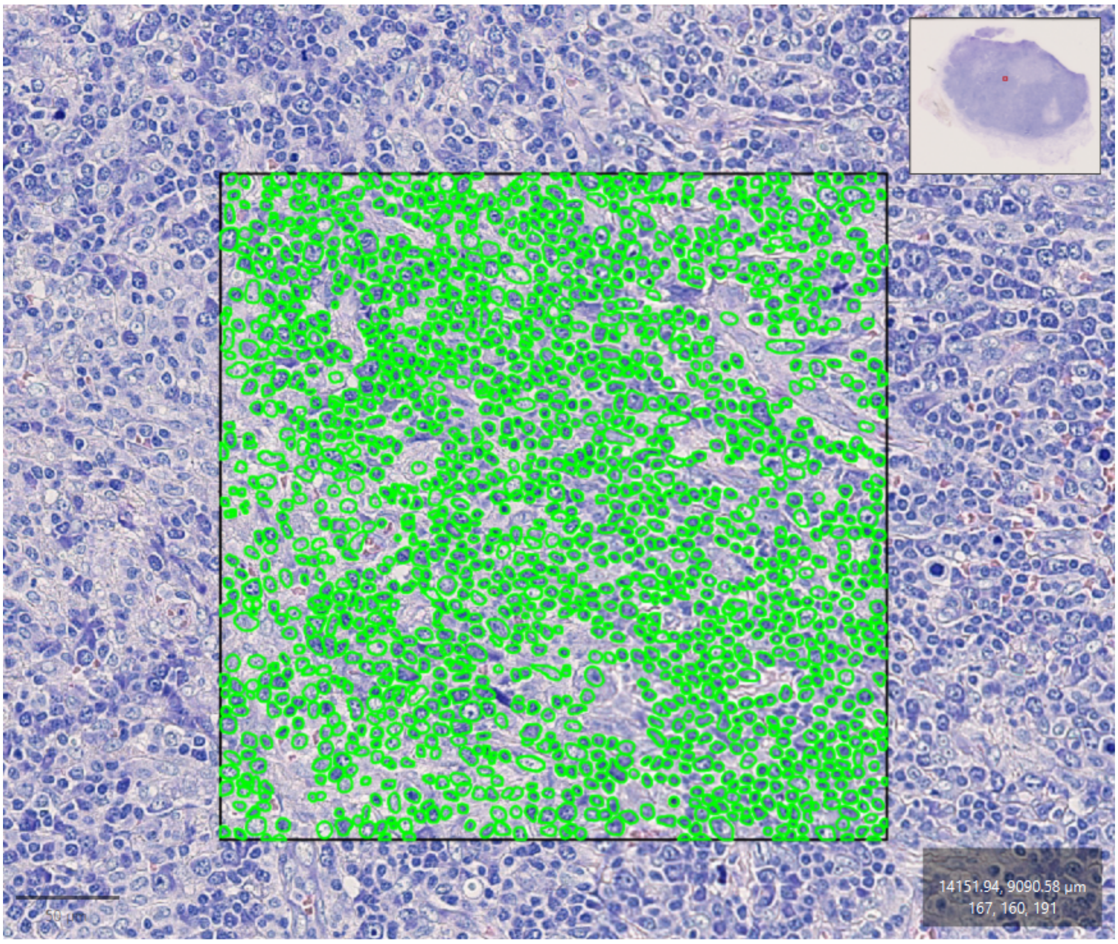


Figure 3.5: QuPath annotation view

The division into training, validation, and test sets was done by patient. 50 ROIs were placed in the training set (9 patients), 11 in the validation set (2 patients), and the remaining 12 ROIs (2 patients) in the test set.

3.1.3 Breast IHC

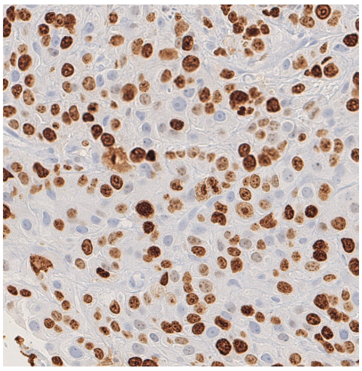


Figure 3.6: Tile from BreastIHC

The Breast IHC dataset was also provided by A.O.U. Città Della Salute e Della Scienza Hospital in Turin. It contains 159 tiles extracted from WSIs of breast tumour tissues obtained by biopsy and stained with IHC, including the markers Ki67, estrogen receptor (ER), and progesterone receptor (PGR). The resolution is 0.442 μm/pixel with a magnification of 20x.

The nuclei were manually annotated by the student author of the Master Thesis in [29] and reviewed by an expert pathologist; the annotations include both segmentation of the nuclei and classification into immunopositive (brown nuclei) and immunonegative (blue nuclei). For the division into subsets, 126 tiles were included in the training set, 25 in the validation set, and 28 in the test set.

3.1.4 LyNSeC 1

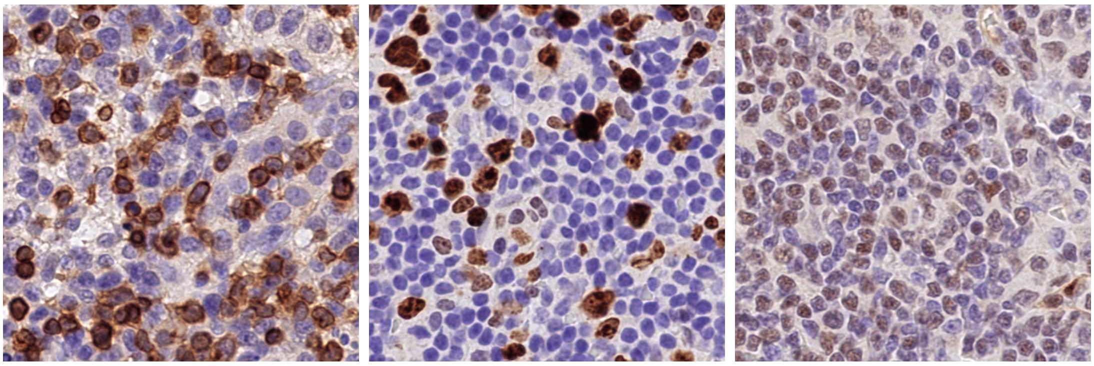


Figure 3.7: CD3, Ki-67 and ERG-stained tiles from LyNSeC 1

The LyNSeC (Lymphoma Nuclei Segmentation and Classification) dataset is a publicly annotated dataset introduced in [26] by H. Naji et al.; it is composed of three subsets: LyNSeC 1, 2, and 3. In this study, we utilised LyNSeC 1, which consists of IHC images with dimensions 512×512 and a magnification of x40. Each image is annotated for both instance segmentation and classification of instances in the immuno-positive and immuno-negative class.

The subset contains 379 images: 280 are stained with the membrane marker CD3, 79 with the nuclear proliferation marker Ki-67, and 20 with the ERG marker, namely the Erythroblast Transformation Specific (ETS)-related gene. Only two markers, CD3 and Ki-67, were actually used for training the networks. The division into train, validation, and test sets was carried out with percentages of 70/15/15 for both marker subsets.

This dataset is provided as a two-layer .mat file in int32 format, consisting of an instance map numbered from 1 to N and a type map in which each instance is assigned a number depending on the type (1 for immunonegative and 2 for immunopositive instances). Additionally, it was refined prior to training to ensure that all instances identified in the instance maps were correctly typed in the type maps.

3.2 HoVer-Net: Simultaneous Segmentation and Classification of Nuclei in Multi-Tissue Histology Images [24]

This section describes the HoVerNet deep learning architecture model for the segmentation and classification of cellular nuclei in histological images.

HoVerNet was developed using H&E-stained slides for training, with the goal of addressing several challenges related to nuclear segmentation: the high level of inter- and intra-instance variability in size, shape, and chromatin patterns across different cell types, diseases, or even within the same tissue; also, the tendency of tumour cell nuclei to form clusters, which complicates instance segmentation and the separation of nearby instances.

Additionally, the model also includes the classification of cellular nuclei, particularly lymphocytic and tumoural nuclei, which are associated with the existence of tumour-infiltrating lymphocytes (TILs), whose presence stands as a crucial predictive indicator for tumour recurrence.

3.2.1 Model Architecture

HoVerNet is composed of three prediction branches: a Nuclear Prediction branch (NP branch), a branch for predicting the Horizontal and Vertical distance maps between pixels and the centroids of the corresponding nucleus (HoVer Branch), and a Nuclear Classification branch (NC branch), which can be activated or deactivated depending on whether nuclear class annotations are available in the dataset.

The HoVer branch is the novel feature introduced by this architecture and it is designed to enhance the network's capability to correctly separate neighbouring nuclei.

All three branches share the same encoder to extract a set of representative features; subsequently, the separation into three distinct branches allows for the simultaneous prediction of both nuclei and HoVer maps, and, when applicable, nuclei type classification.

Based on the outputs from the two or three branches, a custom post-processing step is used to segment the nuclei into separate instances appropriately. This is achieved through a marker-controlled watershed algorithm that leverages the outputs from both the NP branch and the HoVer branch, and, if applicable, classifies the nuclei into their respective types (NC branch).

Network architecture

The encoder used is a PreActivated ResNet50, which is a 50-layer pre-activated residual network (ResNet50) with some variations applied: decrease of the downsampling reduction factor, modifications to kernel strides and max-pooling operations. Additionally, various residual units are applied in series to form residual blocks, which are implemented at different levels of depth.

The decoding, as previously mentioned, occurs across three branches (NP, HoVer, NC) to leverage both segmentation and classification. The upsampling strategy employed is nearest neighbour, and these operations are interspersed with dense units.

Segmentation is carried out through the NP and HoVer branches. The NP branch performs pixel-level classification to distinguish between background and nuclei, while the HoVer branch predicts the vertical and horizontal distances of each pixel from the centroid of the nucleus to which the pixel belongs. The two branches allow for distinguishing the foreground from the background (NP) and subsequently, through HoVer maps, correctly separating the foreground into different nuclear instances, also enabling the separation of overlapping or touching nuclei.

If activated, the NC branch handles the classification of the instances into nuclear types, using a classification algorithm based on majority voting approach on the pixel-by-pixel class predictions of each individual instance.

The network input is 270×270; the output prediction is made on a central area of 80×80. This allows for the avoidance of edge effects, as information from pixels outside this central area is still used for the prediction. This influences the patching and unpatching mechanism during inference (Section 3.2.4).

3.2.2 Additional Border Prediction

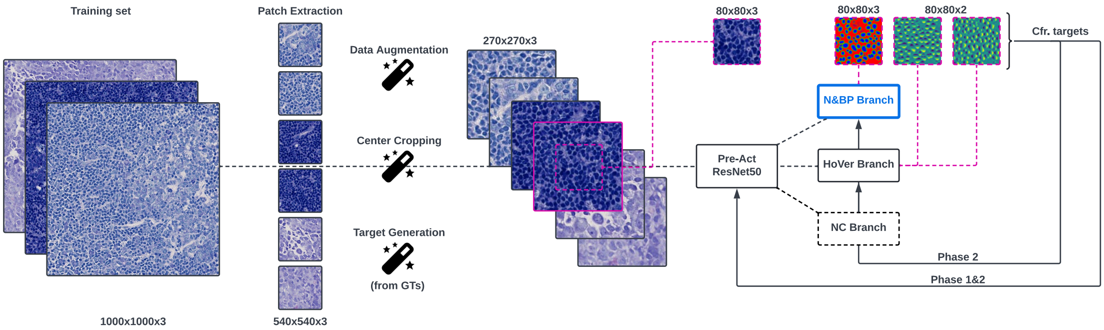


Figure 3.8: Training Process for HoverNet with Border prediction

To further enhance the network's ability to separate neighbouring instances, the NP branch of the network was modified so that it would not merely perform segmentation, i.e., pixel-by-pixel classification as background or nucleus, but as background/nuclear border/nucleus, resulting in the Nuclei & Border Prediction (N&BP) branch. As demonstrated in [30] by Salvi et al., this addition to the prediction task often improves instance segmentation performance. In addition to architectural modifications to the original NP branch, appropriate ground truths must be provided to the network to generate appropriate targets for the N&BP branch; each instance must be accompanied by its border, and border thickness should be consistent across all instances. In particular, to provide the network with the correct masks, two strategies for generating nuclei borders were adopted, referred to as Version 1 and Version 2; see "Patch Extraction and Data Augmentation" in 3.2.3 for details.

3.2.3 Training Process

Patch Extraction and Data Augmentation

Since the network input is 270×270, and we are working with tiles sized 1000×1000 for MoNuseg, 1500×1500 for Giemsa Dataset, 512×512 for LyNSeC 1 or with various dimensions for the BreastIHC Dataset, it is necessary to apply a patch extraction algorithm to feed input into the network during training. This is also required to perform inference, though the mechanisms for patching and unpatching differ from those used in training and are described in Section 3.2.4.

For patch extraction from training tiles, a hand-crafted script is used to define the dimensions of the patches to be extracted and the stride with which the sliding window moves over the image. Additionally, it is determined whether to extract only valid regions or to mirror the image borders. The size of the sliding window used is 540×540, with a step size of 164 in both vertical and horizontal directions. The use of a window larger than the input is aimed at avoiding border effects during data augmentation operations such as rotation, while the smaller step size ensures no loss of information and enables data augmentation by passing over the same pixels multiple times. We apply mirrored borders.

This operation is performed on each RGB channel of the tiles and their corresponding ground truth(s).

After patching tiles and annotations from both training and validation sets, each RGB patch and respective ground truths are concatenated and saved as '.npy' files. These files are then read by the dataloader, which simultaneously applies transformations for data augmentation: for training images, operations such as rotation, translation, scaling, shear, flipping, blurring, noise addition, and adjustments to brightness, contrast, hue, and saturation are performed. Subsequently, center cropping to 270×270 is carried out for both the images and masks to match the input size of the network. For validation, only cropping is performed.

Data Format: Original Configuration

In the original configuration, the dataloader requires as input a multidimensional matrix in '.npy' format composed of the stack of the image in RGB format and its annotations; the annotations for instance segmentation are instance maps where all pixels of the n-th instance are assigned the value n; therefore, for N instances, the foreground pixels have values in the range $[1, N]$. Optionally, the annotations for the classification branch are provided as an additional layer, where, similarly to the instance map, the pixels of a given n-th instance are numbered based on their class, so that in the presence of K possible classes, the foreground pixels vary in the range $[1, K]$. The annotations are saved in '.mat' files, with the data stored in **int32** format.

Finally, the targets for each branch are generated:

- For the NP branch, the target is simply the binarised instance map, meaning the division of pixels into individual instances is lost, and only the dichotomy of background/nuclei is retained.

- For the HoVer branch, the targets are generated with a custom function that retrieves the instance division from the instance map and calculates the horizontal and vertical distances between each pixel and the centroid of the corresponding nucleus.

- For the NC branch, the target is the one loaded in the dataloader as precedently mentioned.

Data Format: Border Version 1

If the segmentation of the third class (the border) is planned, the generation of N&BP target maps according to Version 1 is addressed as follows: the input for the net stays the same, so instance maps with pixels ranging from 1 to N, where N is the number of instances; to obtain the target for the N&BP branch, the function generating targets is modified so that instance maps are processed instance by instance to generate a 3-pixel-wide border each, achieved through the use of a morphological erosion operation with a disk of radius three as structural element. The border pixels of the nucleus will be labelled as '1's and the inner of the nucleus with '2's. The HoVer maps are generated in the same way as before, and the handling of the targets for the type maps in the classification branch remains unchanged.

This approach has drawbacks. The primary issue is that the borders are not the original boundaries of the annotated instances, but are instead generated *a posteriori* through a morphological operation. This can lead to inaccuracies, as the artificially generated borders may not precisely reflect the true anatomical boundaries of the nuclei. For these reasons this Version of the border generation was not brought any further.

Data Format: Border Version 2

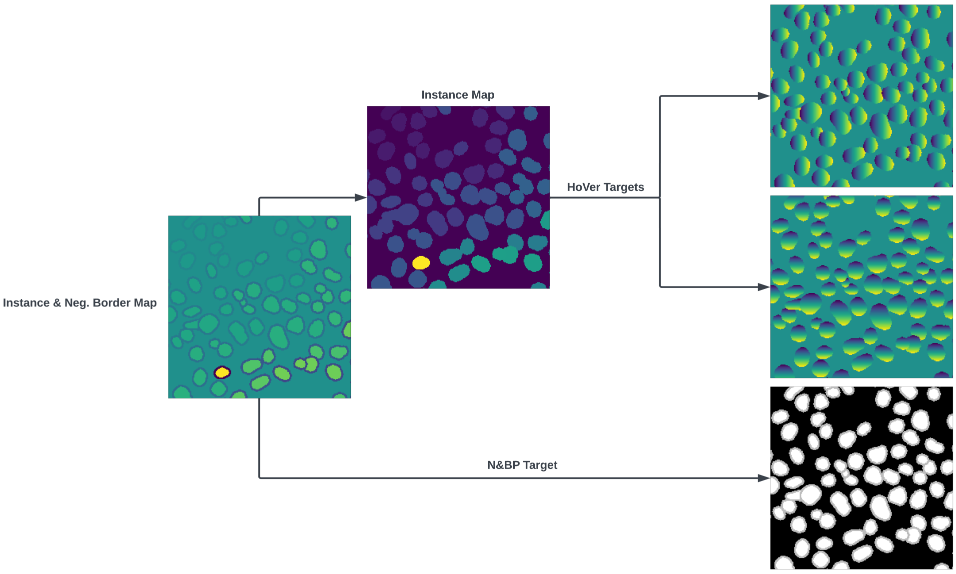


Figure 3.9: Target generation from input with Border Version 2

With Version 2 of the border generation, the aim is to overcome the issues present in Version 1 generation by using an *a priori* strategy, though at the obvious cost of having to modify the network's input. The goal is to use the true borders of the annotated instances, whose coordinates are known because, during the manual segmentation process, the resulting annotations are usually saved as files containing the coordinates of the points belonging to each instance and not as instance maps – the latter is generated from instance coordinates.

To transfer this border information to the network and simultaneously maintain the ability to retreive the original instance map to generate HoVer maps in the same manner as before, the network input and the target generation strategy were altered accordingly.

To uniquely derive the border of each instance without ambiguity between instances and to do so even at patch-level rather than tile-level (as mentioned, target generation occurs after patching), the previously described instance map is modified as follows: for a given n-th instance, whose pixels have a value of n, the pixel values along the borders are changed from n to $-n$, creating a 3-pixel-wide border. The value n for the internal pixels remains unchanged.

With this input, we can generate the 3-class masks with the borders by assigning all negative pixels a value of 1 (borders) and all positive pixels a value of 2 (nuclei inner). To retrieve the instance map (for generating the HoVer maps), it is sufficient to take the absolute value of all pixels. This simple strategy avoids the need for more complex algorithms, for target generation and instead relies on basic operations. Since the masks are encoded as '.mat' files in *int32* format, handling negative values poses no issue.

```python
def gen_targets(ann, crop_shape, **kwargs):

    """Generate the targets for the network."""
    np_map = ann.copy()
    input_hv = ann.copy()
    np_map[np_map > 0] = 2
    np_map[np_map < 0] = 1

    input_hv = abs(input_hv)
    hv_map = gen_instance_hv_map(input_hv, crop_shape)
    hv_map = cropping_center(hv_map, crop_shape)
    np_map = cropping_center(np_map, crop_shape)
    target_dict = {
        "hv_map": hv_map,
        "np_map": np_map,
    }
    return target_dict
```

Loss function

The loss function used to optimize the four sets of weights (one set for the encoding branch and one set for each of the three decoding branches) is defined as follows:

$$\mathcal{L} = \underbrace{\lambda_a \mathcal{L}_a + \lambda_b \mathcal{L}_b}_{\text{HoVer branch}} + \underbrace{\lambda_c \mathcal{L}_c + \lambda_d \mathcal{L}_d}_{\text{NP branch}} + \underbrace{\lambda_e \mathcal{L}_e + \lambda_f \mathcal{L}_f}_{\text{NC branch}}$$

As indicated in the equation, the first two terms represent the loss components that backpropagate the error to the HoVer branch, the second two terms correspond to the NP (or N&BP) branch, and the last two to the NC branch. The λ coefficients are weighting factors for the loss and are chosen empirically. The losses for the classification branch are only computed if it is active.

- $\mathcal{L}_a$ is the mean squared error between the HoVer maps predicted by the network and the provided ground truth.

- $\mathcal{L}_b$ is the mean squared error between the gradients computed on HoVer maps and the corresponding gradients of the ground truth.

- $\mathcal{L}_c$ and $\mathcal{L}_e$ are Cross Entropy losses.

- $\mathcal{L}_d$ and $\mathcal{L}_f$ are Dice losses.

Both Cross Entropy and Dice losses are easy adapted to the multi-class prediction of nuclei and borders mentioned in Section 3.2.2.

Training Configuration

The training of HoVerNet occurs in two phases. During the first phase, only the encoder, which is responsible for feature extraction, is trained; the encoder may also be pretrained. The second phase involves training both the encoder and the three decoder branches, which is consequently more computationally demanding.

Various configurations were tested to fine-tune the parameters. Some options remained unchanged. These are listed in Table 3.2.

Component	**Details**
Optimizer	Adam
Learning Rate (lr)	1.0×10^{-4}
Betas	(0.9, 0.999)
LR Scheduler	StepLR, Step Size: 25
Loss Functions	
NP	BCE, Dice
HoVer	MSE, MSGE
NC	BCE, Dice

Table 3.2: Optimizer and Loss Configuration

In Tables 3.3, 3.4, 3.5, 3.6, and 3.7, the various configurations used for training HoVerNet, both with and without border prediction, are presented. Border prediction was always performed using the 2nd Version of the Border generation.

Config. 1	**Value Ph. 1**	**Value Ph. 2**
Number of processors	12	8
n° Epochs	50	50
Batch size	16	4
Pretraining	Pre-act ResNet50	—

Table 3.3: Configuration 1

Config. 2	**Value Ph. 1**	**Value Ph. 2**
Number of processors	12	8
n° Epochs	50	50
Batch size	16	4
Pretraining	NO PRETRAIN	NO PRETRAIN

Table 3.4: Configuration 2

Config. 3	**Value Ph. 1**	**Value Ph. 2**
Number of processors	12	8
n° Epochs	25	50
Batch size	16	4
Pretraining	Pre-act ResNet50	—

Table 3.5: Configuration 3

Config. 4	**Value Ph. 1**	**Value Ph. 2**
Number of processors	12	8
n° Epochs	50	80
Batch size	16	4
Pretraining	Pre-act ResNet50	—

Table 3.6: Configuration 4

Config. 6	**Value Ph. 1**	**Value Ph. 2**
Number of processors	12	8
n° Epochs	25	50
Batch size	8	4
Pretraining	Pre-act ResNet50	—

Table 3.7: Configuration 6

3.2.4 Inference for Segmentation

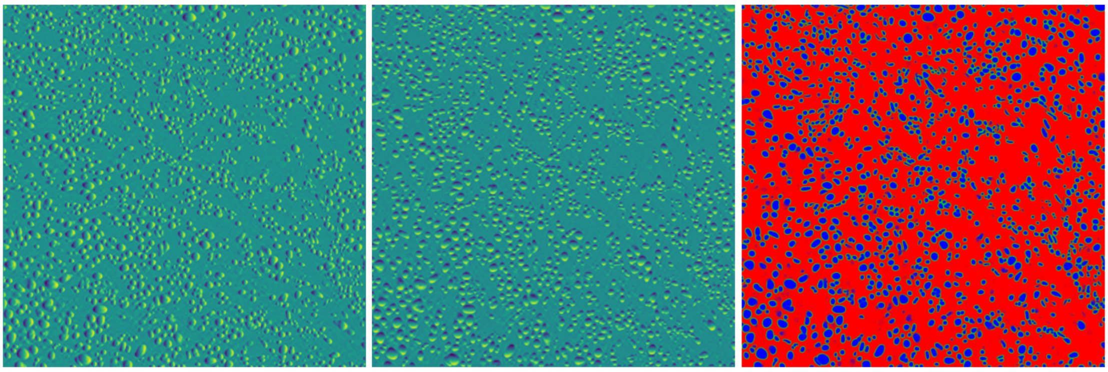


Figure 3.10: HoVer Maps and three-layer Softmax outputs

Patch Extraction and Unpatching

To perform inference on an entire tile input, the first step is to divide the tile into patches. The patch division must be conducted in such a way that it covers the entire surface of the tile. Consequently, using hand-crafted code, the tile is divided into patches by sliding a 270×270 window with an 80×80 stride. Mirror padding is applied so that the various central 80×80 sub-patches cover the entire tile. The choice of mirroring rather than zero padding is made to reduce edge effects when external patches are passed through the network.

After stacking all the extracted patches, they are fed into the network one by one. The net's output, depending on the model used, either with or without NC branch and either with or without border prediction, can have different dimensions on the z-axis.

In any case, from the output's layers a stack is created for each one of them, that is, the horizontal maps are stacked together, the vertical maps are stacked together, the first layers of the softmax are stacked, and so on, so stacking is performed separately for each layer of the output. An unpatching function is then used so that, given the initial dimensions of the tile and the applied padding, it reconstructs each output for the entire image.

This code is written in such a way as to generalize the algorithm for images of any size, not necessarily squared, as the images in the Breast IHC dataset are not squared and have variable dimensions. Based on the post-processing strategy adopted, the necessary outputs are processed.

The following are the two (alternative) post-processing steps aimed at generating an instance map labeled from 1 to N based on the network output, regardless of the presence of the classification step. Afterwards, it is specified how to include the classification of the instances as well.

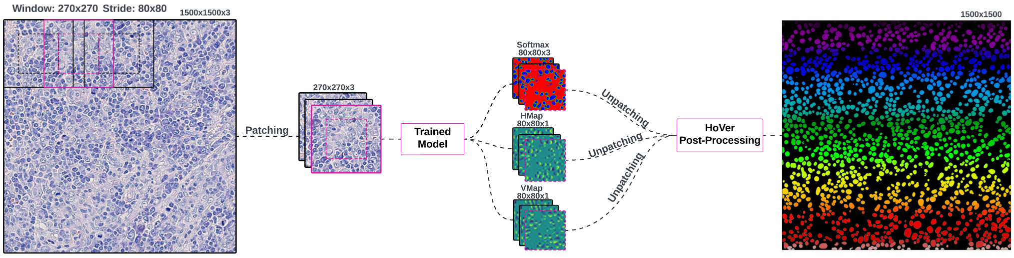


Figure 3.11: Inference Process for HoVer Post Processing

HoVer Post-Processing

The creators of the HoVerNet architecture proposed a post-processing algorithm based on both the nuclei probability map and the horizontal and vertical maps, thus utilizing the outputs from two distinct branches. This approach can be used whether or not border class prediction is performed; in cases where border prediction is present, the sum of the nuclei probability map and the border probability map replaces the former. Moreover, this process is also applied when the network is

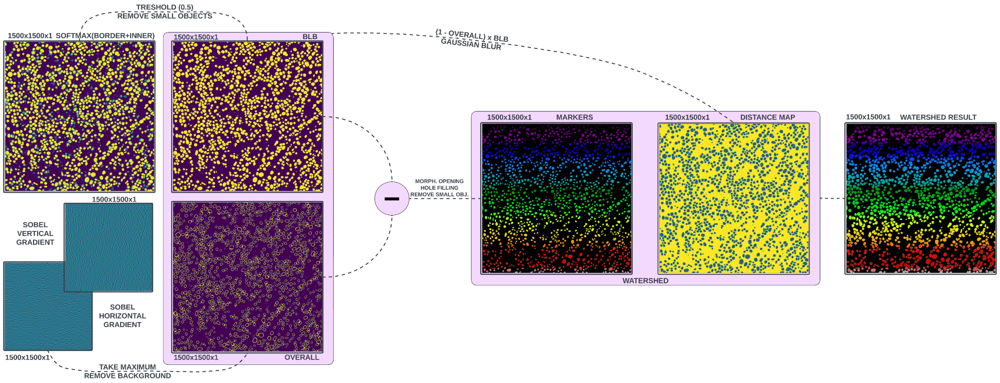


Figure 3.12: Inference Process in detail: HoVer Post-Processing

trained not only for instance segmentation but also for their classification, activating the NC branch; this is further described in the section 3.2.5.

After the steps of patching, inference, and unpatching, the HoVer maps and the softmax layer related to the nuclei probability (or, if border prediction is present, the sum of the nuclei and border probability layers) are retained.

The following operations are then performed:

$$S_m = \max(H_x(p_x), H_y(p_y))$$

where p_x and p_y are the horizontal and vertical maps, and H_x and H_y are the horizontal and vertical Sobel gradient operators, respectively. This highlights significant directional changes in the HoVer maps.

Markers are calculated as:

$$M = \sigma(\tau(q, h) - \tau(S_m, k))$$

where $\tau(q, h)$ is a simple global thresholding of map q with threshold h, and q is originally the output probability map from the NP branch, modified in N&BP situation as the sum of the probabilities of nuclei and borders when borders are segmented.

Similarly, $\tau(S_m, k)$ is the thresholding of the Sm map with threshold k, and the function σ sets all negative values to 0.

The energy landscape is then calculated as:

$$E = [1 - \tau(S_m, k)] \times \tau(q, h)$$

Markers M are used as seeds during the watershed given the energy landscape E.

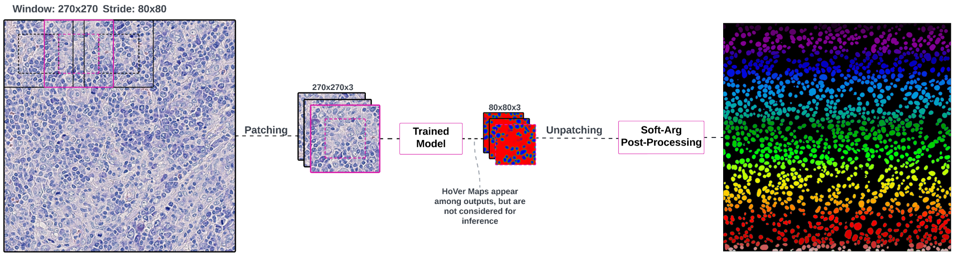


Figure 3.13: Inference Process for Softmax - Argmax Post Processing

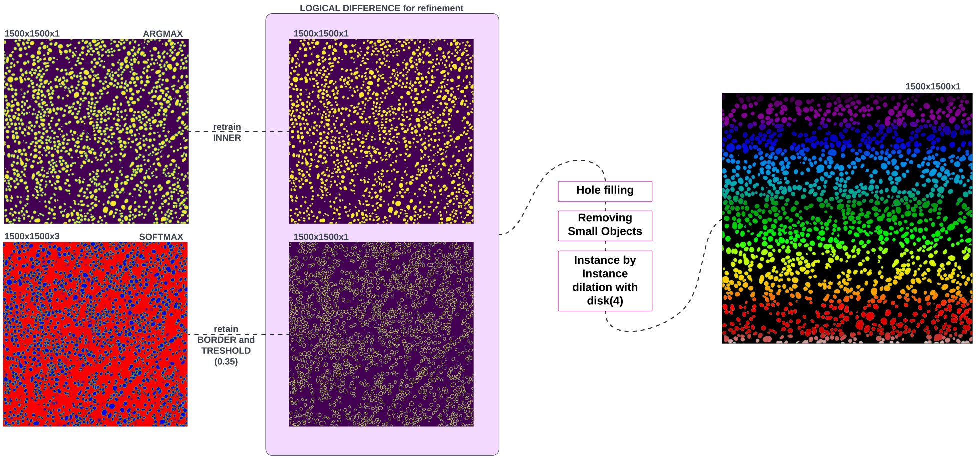


Figure 3.14: Inference Process in detail: Softmax - Argmax Post-Processing

Softmax - Argmax Post-Processing

This inference method is employed specifically when the borders of instances are predicted. After patching, inference and unpatching, a 3-layer softmax probability output (background, border, inner) is obtained among the outputs. Applying argmax along the z-direction produces a 3-tone grayscale mask, where each pixel adopts the grayscale value corresponding to the most probable layer. From this, only the inner parts of the cells, i.e., pixels with a value of 2, are retained. Furthermore, the layer corresponding to the border probability map is kept from the softmax output and binarized using a threshold fine-tuned to 0.35.

The binarized softmax of the borders is then used to refine the argmax of the nuclei. Specifically, any pixel with a value of 1 in the binarized softmax is set to

0 in the argmax of the inners. This step is aimed at enhancing the separation between adjacent or overlapping instances in the final segmentation, recognizing them as distinct entities.

Consequently, operations such as the removal of small objects, hole filling, and, finally, instance by instance, dilation with a radius-4 disc (also fine-tuned to maximize performance) are performed to recover the original dimension of each nucleus and ensuring the integrity of each instance's shape. An instance segmentation map is obtained.

3.2.5 Optional Nuclear Classification Step

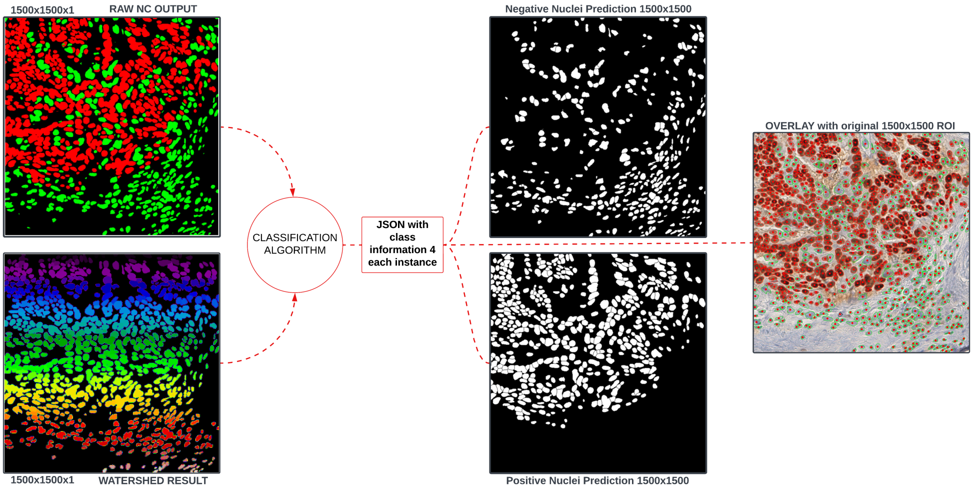


Figure 3.15: Classification Process after HoVerNet Post Processing

The classification process of instances is described as an additional step; it has become apparent from metrics evaluation that the datasets Breast IHC and LyNSeC, on which we perform classification, suffer less from the issues of overlapping or touching nuclei that necessitated the development of the border prediction model, so classification was employed mainly after the HoVerNet post-processing, as the Softmax-Argmax post-processing is only utilized in cases involving border prediction.

As reported by the authors of the HoVerNet model architecture, classification leverages the output from the watershed, which provides an instance map, and the raw output from the classification branch that has been re-patched. However, this kind of classification algorithm can be performed starting from any instance map as long as a raw classification map is provided.

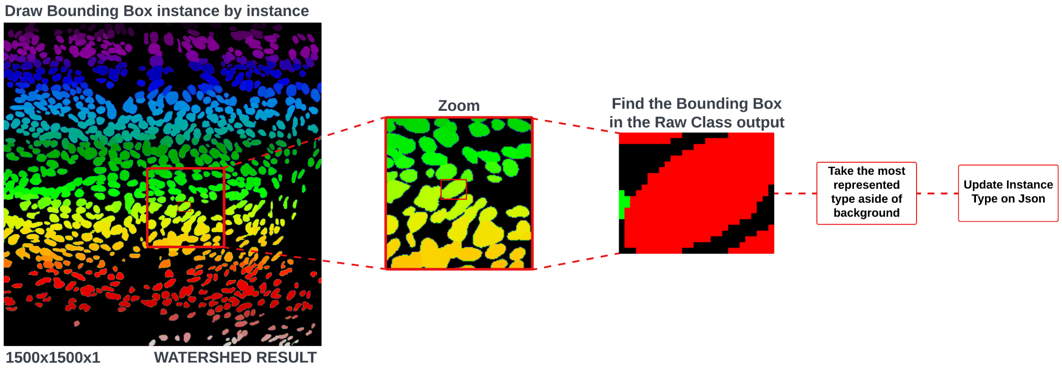


Figure 3.16: Classification Process after HoVerNet Post Processing

From the instance map, a bounding box for each instance is calculated. The area corresponding to the bounding box is then analyzed in the raw type map. Within this area, the predominant type, excluding the background, is determined. So through majority voting between the positive and negative classes, the type is assigned to the i-th instance. This classification information is saved in a JSON file along with the centroids and details for each instance, such as border pixels determined during watershed. The contents of the JSON file can then be used to overlay on the original image or to trace masks for instances classified only as positive or negative types.

JSON Extract with Instance and Type Data

```json
{
    "1": {
        "bbox": [[0, 87], [12, 110]],
        "centroid": [99.05, 4.66],
        "contour": [[87, 0], [87, 1], [88, 2], [88, 3]],
        "type_prob": 0.9906,
        "type": 2
    },
    "2": {
        "bbox": [[2, 170], [16, 216]],
        "centroid": [193.19, 8.18],
        "contour": [[174, 2], [173, 3], [171, 3], [170, 4]],
        "type_prob": 0.9803,
        "type": 1
    }
}
```

3.3 CDNet: Centripetal Direction Network for Nuclear Instance Segmentation [25]

The deep learning architecture, Centripetal Direction Network (CDNet), has been adopted to perform instance segmentation on MoNuSeg dataset to provide a performance comparison. This architecture does not incorporate a classification step. It was designed to overcome challenges in instance segmentation tasks, such as the separation of overlapping and touching nuclei.

A novel contribution of CDNet is the use of Direction Feature Maps, Direction Difference Maps (DDM), and the Direction Guided Refinement Module (DGM). These components are specifically engineered to enhance the precision of segmentation by effectively delineating individual instances within clusters of closely situated nuclei.

3.3.1 Key Components

This network aims to learn to predict the directional features of a pixel belonging to an instance in relation to the center of the instance itself. To achieve this, DDMs are constructed to distinguish between overlapping instances. This method helps the network identify and delineate the boundaries of individual nuclei even when they are closely packed or partially overlapping, thereby improving the accuracy of instance segmentation.

Centripetal Direction Feature

The Centripetal Direction Feature describes spatial relationships among pixels belonging to the same instance. To do this, centripetal adjacent directions are defined. These are directions that pass through the pixel and the center of the instance and point towards the center of the instance. This method enables the precise modeling of the internal structure of each instance.

Direction Feature Map

Direction Feature Maps are directional labels that make the network able to learn the Centripetal Direction Feature for every instance pixel. These maps are generated from manual annotations or instance maps by calculating a distance map between each pixel and the boundary of the instance, which assists in pinpointing the center of each nucleus. Subsequent operations highlight information regarding the gradient and angle, thereby creating a direction map that displays the centripetal direction for every pixel. The steps for generating these maps from ground truth are as follows (Figure 3.17):

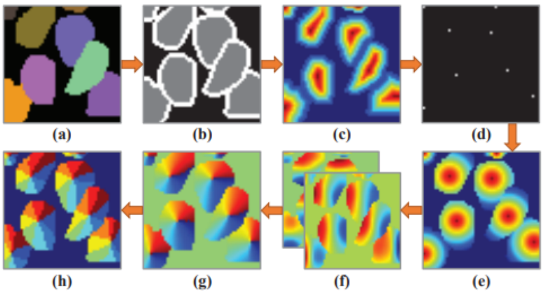


Figure 3.17: The process of generating centripetal direction ground truth [25]

(a) Original Ground Truth Mask: The baseline mask representing the initial annotations.

(b) Three-Class Ground Truth Mask: This mask differentiates between the background, the interior of the instance, and the boundary, represented by black, grey, and white respectively.

(c) Pixel-to-Boundary Distance Map (M): A map that calculates distances between each pixel and its nearest boundary, aiding in the detection of instance edges.

(d) Centre Point Map: This map identifies the central point of each nucleus.

(e) Pixel-to-Centre Distance Map (U): It measures the distance from each pixel to the centre of the instance, which is crucial for defining the nucleus's core.

(f) Gradient Map (G): This map shows the gradient at each pixel, highlighting changes in intensity that define shapes and boundaries.

(g) Angle Map: It provides the angle of the gradient at each pixel, offering directional data.

(h) Centripetal Direction Ground Truth (D^C): This final map shows the direction towards the centre for each pixel, aligning with the centripetal direction feature.

Discretizing the centripetal direction feature into blocks rather than assigning it on a per-pixel basis makes the description of directions less susceptible to noise. As evident from the images, the abrupt variation in this feature allows for the clear distinction and separation of adjacent instances.

Direction Difference Map (DDM)

To better identify instance boundaries, the DDM is employed. It is map calculates both the likeness within instances and the differences between them by analyzing the directional changes among neighbouring pixels. It captures essential contextual details by emphasizing these directional variations, which helps to clarify the boundaries of instances. The DDM is especially effective in outlining the contours of overlapping nuclei, as it highlights the shifts in direction at the borders. These features collectively enhance the spatial connections among pixels and elevate the overall quality of the segmentation effort.

3.3.2 Model Architecture and Training Process

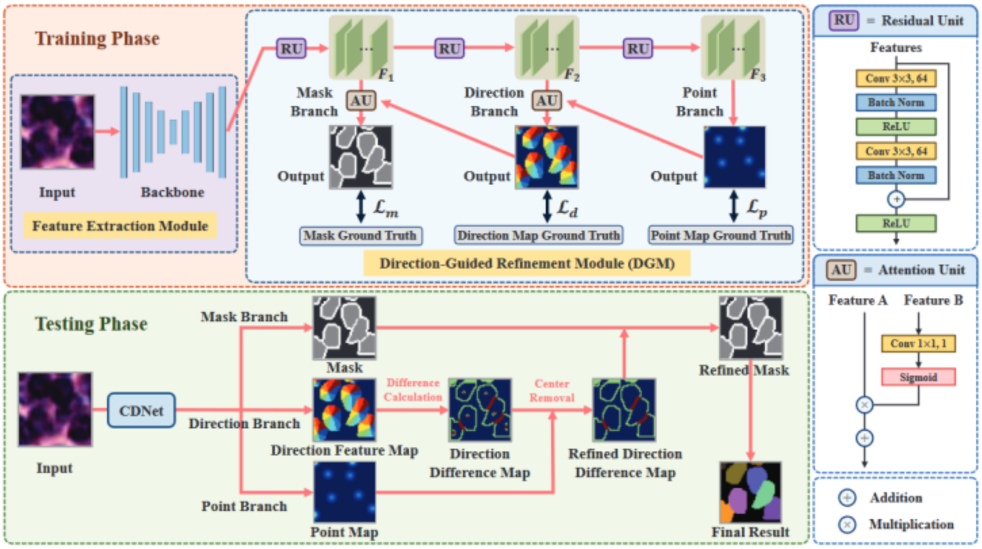


Figure 3.18: Flowchart of the Centripetal Direction Network [25]

The architecture of the CDNet is comprises two main modules: a module for feature extraction and one for direction-guided refinement (DGM).

The module for feature extraction is a CNN trained to pull out the most descriptive features of the input image through successive convolutions, batch normalization, and ReLU activation functions to introduce non-linearity.

The DGM is a versatile, plug-and-play module with three branches: a mask segmentation branch, a center point detection branch, and a direction segmentation branch. The targets for these three tasks are directly derived from the process of creating centripetal direction features.

The branch for direction segmentation is included to enhance the suboptimal performance of the point branch compared to the mask branch. The former tends to overlook features distant from the center of the instance, focusing instead on those near the central area. In contrast, the mask branch benefits from and requires knowledge of features belonging to the entire instance and its border. Consequently, integrating the direction prediction branch allows for the extraction of comprehensive information about the instance and the correlations between its internal pixels, thus bridging the gap. This task is strategically placed between the two previous ones for this very reason.

The feature map extracted from the CNN undergoes processing through Residual Units (RUs), resulting in an output feature map. This output is then directed to a second RU as well as to another pathway, specifically an Attention Unit (AU), referred to as the "reverse transmission path".

There are a total of three successive RUs and two AU branches, with the AU present only in the direction and mask branches, and not in the point branch, as indicated during the Training Phase shown in Figure 3.18. The first RU processes the initial set of features obtained in the feature extraction module, resulting in F_1, which are the features for the mask branch; the second RU processes F_1 to obtain the the direction branch's set of features F_2; and similarly, the third processes F_2 to get F_3 for the point branch. Each RU involves operations including 3×3 convolutions, batch normalization, ReLU activation, and a shortcut connection.

The AUs are utilized to refine feature extraction and integration across different branches. As the term implies, they focus on "attending" to relevant features over less significant ones, such as emphasizing boundaries in densely packed nuclear areas. The maps F_1 and F_2 each feed into an AU, to which the outputs from the direction and point branches are also concatenated respectively. From the first AU, we obtain the output of the mask segmentation branch, and from the second, the output of the direction map. The output for the point branch is derived directly from F_3 without passing through an AU.

Loss Function

The total loss function is expressed as:

$$L_{total} = w_1 L_m + w_2 L_d + w_3 L_p$$

where each loss term is multiplied by a coefficient that weights the respective loss; L_m is the the one for the mask prediction branch, calculated using cross-entropy loss combined with Dice loss, L_d for the direction prediction branch is calculated similarly to L_m, and L_p for the point map prediction branch is calculated using mean square error (MSE) loss.

Training Parameters: MoNuSeg

Parameter	**Value**
Number of Epochs	300
Batch Size	8
Early Stopping Criteria	Validation Loss
Optimizer	RAdam
Initial Learning Rate	0.001
Data Augmentation	Horizontal Flipping, Vertical Flipping, Random Rotation, Cropping
Loss Function Components	Cross-Entropy, Dice Loss, MSE

Table 3.8: CDNet Model Training Configuration

Training was halted after 101 epochs due to the early stopping criterion. Inference was performed using the best checkpoint, which was at epoch 55.

What about Giemsa Dataset?

Training with the Giemsa-stained dataset was not feasible due to its computational demands, which exceeded the architecture's capacity.

3.3.3 Inference

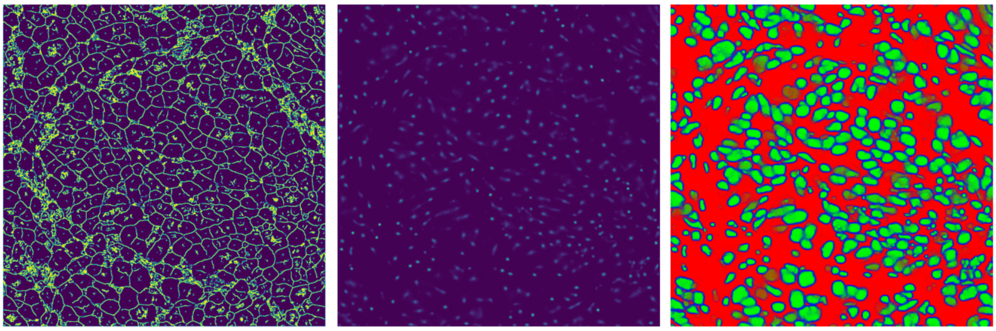


Figure 3.19: DDM, Point Map and three-layer Softmax outputs

Patching e Unpatching

The 1000×1000 tiles from the MoNuSeg dataset are divided into 256×256 patches to align with the network's input requirements, and data augmentation techniques have been employed to expand the number of examples provided to the network. These techniques include horizontal and vertical flipping, cropping, and rotation. Similarly to HoVerNet, the various outputs for each patch are unpatched before post-processing is carried out.

Output Processing

Two methods of performing inference were evaluated. The first involves the use of the softmax output at the mask branch, similarly to what describet in Section 3.2.4. The second employs the DDM to enhance the network's ability to distinguish between adjacent instances. The inference strategy proposed in the original CDNet paper will be detailed further.

The network's output comprises a three-class classification map, specifically for background, border, and inner regions; a continuous point map; and the direction feature. The DDM is calculated from the direction feature. Given D as the DDM and D_{p_i} as the value of the DDM at pixel p_i:

$$D_{p_i} = \begin{cases} 1 - \min_{p_j \in \mathcal{V}_{p_i}} \{\text{dis}(p_i, p_j)\}, & \text{if } p_i \in \mathcal{P} \\ 0, & \text{if } p_i \in \mathcal{B} \end{cases}$$

$\mathcal{V}_{p_i}$ refers to the pixels neighbouring p_i; $\{\text{dis}(p_i, p_j)\}$ is a dissimilarity metric between two pixels, calculated from the gradient map G (see Figure 3.17 and Section 3.3.1).

Pixels within the same instance will have low D values unless they are at the center of the instance; pixels at the borders, overlapping, or common to different instances will exhibit high D values. Thus, we have a single metric to discriminate between internal and overlapping pixels between instances. As said, in DDM not only pixels at the borders but also those at the center of an instance will have high D values. Therefore, the point map comes into play to refine the DDM and eliminate the central directional difference within an instance. The center of the instance is defined as the area where the median DDM value is smaller than the corresponding value in the point map.

Finally, the DDM is calculated: it is a map weighting the border class probabilities in the original three-class prediction, with a reinforcement function shown below:

$$p_i^{b_{re}} = (p_i^b + \frac{D_{p_i}}{2}) \times (1 + D_{p_i})$$

where $p_i^{b_{re}}$ is the probability, for pixel p_i, of belonging to the border in the original softmax, and D_{p_i} is the refined DDM value at pixel p_i.

Test Time Augmentation (TTA)

Test Time Augmentation (TTA) has been implemented as a technique to reduce the uncertainty associated with the network's predictions, applied to both the probability maps and the direction feature maps. This technique involves feeding the same image into the network after applying reversible transformations such as flips and rotations; after inference, the transformations are reversed, and the results are averaged. This process acts as a form of denoising for the prediction in areas where the network is uncertain and strengthens the result. The averaged direction feature maps are then used to generate the DDM, which in turn refines the predicted edges in the probability maps.

Instance Segmentation with Watershed

The softmax output with boundary enhancement is fed into the argmax function, from which the 'inner' areas are extracted. Subsequently, the binary mask undergoes dilation to expand the regions identified as objects, aiding in closing small gaps and connecting nearby components that might have been erroneously segmented as separate. This step generally serves to reclaim border pixels. The dilated mask, converted to a suitable format (multiplied by 255 and cast to uint8), is then passed to a function that performs watershed segmentation.

Within this function, connected regions are labeled, and distance transforms, which are distance maps for each labeled instance, are generated; pixels with values over 125 become markers. Finally, the watershed algorithm is applied using these markers and the negative distance map to separate overlapping instances.

The remove small objects function is used to eliminate instances smaller than a given size. After all cleaning and enhancement steps of the segmentation, all instances are re-labeled sequentially to maintain label consistency.

3.4 Metrics Evaluation

Following the training of a network and its testing, it becomes necessary to quantify its performance. This is important not only to calibrate the training process and assess whether phenomena like overfitting are occurring, but also, more generally, to evaluate the network's effectiveness in performing the designated task and its ability to generalise what it has learned on the training set to the test set.

In the context of instance segmentation, it is essential to use both pixel-level and object-level metrics.

3.4.1 Pixel Level Metrics

Pixel-level metrics assess the quality of segmentation by evaluating, pixel by pixel, the similarity between the ground truth and the network's output prediction. This allows us to understand how well the algorithm can separate the background from the foreground (or distinguish between multiple classes), and therefore classify a single pixel correctly into its respective class, as well as predict the shapes of objects. However, this approach loses information regarding the network's ability to identify individual instances, i.e., to separate the nucleus object into distinct instances, even when they overlap or are adjacent. Consequently, it provides limited insight into the effectiveness of the instance segmentation process itself.

The following metrics all vary within a range from 0 to 1.

Precision

The pixel-based Precision is the ratio of true positive pixels (pixels correctly detected as part of a nucleus) to all positive pixels (all pixels detected as part of a nucleus):

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

This provides an indication of how many of the pixels detected as nuclei are actually nuclei.

Recall

The pixel-based Recall calculates the ratio of all pixels correctly detected as positive to all pixels that should have been detected as such, including those correctly detected as positive and those mistakenly detected as negative. The formula is:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

This gives us an indication of how often a pixel is correctly identified out of all the pixels that should have been recognised.

Jaccard Index

The Jaccard Index is a measure of similarity between two sets (in our case, the pixels segmented in the ground truth and the network's prediction) that can be interpreted as an intersection over union. The intersection represents the pixels correctly detected, while the union includes both the correctly detected pixels and those where errors were made (false positives and false negatives). True negatives are not considered, as they lose significance in a set-based interpretation of pixels. The set definition is:

$$\text{Jaccard} = \frac{|A \cap B|}{|A \cup B|}$$

where A is the set of pixels in the ground truth, and B is the set of pixels in the prediction.

This provides an indication of the overlap between the two sets: the predicted area and the actual area.

Dice Score

The Dice Score is a metric very similar to the Jaccard Index, with similar set notation, but the key difference is that the intersection is weighted twice. This gives more emphasis to correctly predicted pixels.

The formula using set notation is:

$$\text{Dice} = \frac{2 \cdot |A \cap B|}{|A| + |B|}$$

where A is the set of pixels in the ground truth, and B is the set of pixels in the prediction.

This double-weighting of the intersection provides an alternative way to measure the overlap between the predicted and actual areas, often resulting in a slightly higher score compared to the Jaccard Index. A Dice Score of 1 indicates a perfect congruence between the ground truth and the network's prediction.

3.4.2 Object Level Metrics

To more thoroughly evaluate the performance of an instance segmentation task, metrics that account for the presence of multiple instances are used. These metrics shed light on the network's ability to accurately locate, detect, and separate individual instances. They provide insight into how well the network distinguishes between distinct objects, even when they are close together or overlapping, going beyond pixel classification to assess the effectiveness of instance segmentation itself.

Precision

Object-level Precision has the same formulation as pixel-level precision, but the notion of "positive" no longer refers to a pixel but to an instance. Thus, "positivity" refers to a "found instance", and true or false refers to whether the detection is correct or not.

The formula is:

$$\text{Precision}_{\text{objects}} = \frac{\text{True Positives}_{\text{objects}}}{\text{True Positives}_{\text{objects}} + \text{False Positives}_{\text{objects}}}$$

This metric indicates how many of the detected instances are correctly identified as objects.

Recall

Analogous to Precision, the definition of object-based Recall shifts from detecting pixels to detecting instances. False negatives refer to instances that exist but were not detected.

The formula is:

$$\text{Recall}_{\text{objects}} = \frac{\text{True Positives}_{\text{objects}}}{\text{True Positives}_{\text{objects}} + \text{False Negatives}_{\text{objects}}}$$

This metric indicates how many of the actual instances were correctly detected by the network.

Dice Score

The object-level Dice Score has the same formulation as the pixel-level Dice Score, but applied to instances. It measures the overlap between detected and ground truth instances.

The formula is:

$$\text{Dice}_{\text{objects}} = \frac{2 \cdot \text{True Positives}_{\text{objects}}}{2 \cdot \text{True Positives}_{\text{objects}} + \text{False Positives}_{\text{objects}} + \text{False Negatives}_{\text{objects}}}$$

This provides an indication of the overlap between the detected instances and the actual instances, with emphasis on correctly detected instances.

Hausdorff Distance (HD)

The Hausdorff Distance (HD) falls within the category of distance-based metrics, providing a measure of the distance (and hence dissimilarity) between two curves, which are discrete profiles made up of pixels.

It is a "worst-case" measure: given two discrete curves A and B (represented by points), the HD is the maximum of the minimum distances between the points of curve B from the points of curve A and vice versa. In other words, for each point on curve A, the minimum distance to any point on curve B is calculated, and the maximum of these minimum distances is taken, yielding $d_{\mathrm{minA}\to\mathrm{B}}$. The same process is then done in reverse to obtain $d_{\mathrm{minB}\to\mathrm{A}}$. Finally, the maximum of these two measures is taken:

$$\mathrm{HD} = \max(d_{\mathrm{minA}\to\mathrm{B}}, d_{\mathrm{minB}\to\mathrm{A}})$$

A lower HD yields a higher similarity between the two curves; in this case, the two curves considered are those of the predicted instance and its corresponding ground truth.

The full formula is:

$$\mathrm{HD}(A, B) = \max\left(\sup_{a\in A}\inf_{b\in B}\|a-b\|, \sup_{b\in B}\inf_{a\in A}\|b-a\|\right)$$

Where A and B are the sets of points on the two curves, $\|a-b\|$ is the Euclidean distance between points a and b, inf is the minimum, and sup is the maximum function.

Pseudo-code for calculation: Algorithm 1.

Aggregated Jaccard Index (AJI) [18]

The Aggregated Jaccard Index (AJI) is a segmentation metric proposed by Kumar et al. in [18]. To cite the article, *it computes an aggregated intersection cardinality numerator, and an aggregated union cardinality denominator for all ground truth and segmented nuclei under consideration.* In other words, for each i-th nucleus in the ground truth, a corresponding j-th segmented nucleus is associated, and the number of pixels corresponding to the intersection between these nuclei is progressively added to the numerator of the index, while the union of the pixels is added to the denominator.

Pixels present in the ground truth but not segmented are automatically counted in the denominator as false negatives. Additionally, pixels from segmented instances that do not exist in the ground truth (false positives) are also added to the denominator afterward.

This approach penalises the following errors:

- Failure to detect objects present in the ground truth.
- False detection of objects not present in the ground truth.
- Under-segmentation and over-segmentation of correctly annotated objects.

  AJI ranges from 0 to 1, where 1 indicates perfect instance segmentation.

  Formula:

$$\mathrm{AJI} = \frac{\sum_{i=1}^{N}(|G_i \cap S_j|)}{\sum_{i=1}^{N}(|G_i \cup S_j|) + \sum_{\mathrm{unused}\ S_j}|S_j|}$$

Algorithm 1 Calculate Hausdorff Distance (HD)

```
Input: Two sets of points $A$ and $B$ representing two curves
Output: Hausdorff Distance (HD) between $A$ and $B$
$d_{\text{min\_A}\to\text{B}} \leftarrow 0$
for each point $a \in A$ do
    $d_{\text{min}} \leftarrow \infty$
    for each point $b \in B$ do
        $d \leftarrow \text{distance}(a, b)$
        if $d < d_{\text{min}}$ then
            $d_{\text{min}} \leftarrow d$                ▷ Find the minimum distance for point $a$
        end if
    end for
    if $d_{\text{min}} > d_{\text{min\_A}\to\text{B}}$ then
        $d_{\text{min\_A}\to\text{B}} \leftarrow d_{\text{min}}$    ▷ Update the max of minimum distances
    end if
end for
$d_{\text{min\_B}\to\text{A}} \leftarrow 0$
for each point $b \in B$ do
    $d_{\text{min}} \leftarrow \infty$
    for each point $a \in A$ do
        $d \leftarrow \text{distance}(b, a)$
        if $d < d_{\text{min}}$ then
            $d_{\text{min}} \leftarrow d$                ▷ Find the minimum distance for point $b$
        end if
    end for
    if $d_{\text{min}} > d_{\text{min\_B}\to\text{A}}$ then
        $d_{\text{min\_B}\to\text{A}} \leftarrow d_{\text{min}}$    ▷ Update the max of minimum distances
    end if
end for
Return: $\max(d_{\text{min\_A}\to\text{B}}, d_{\text{min\_B}\to\text{A}})$
```

- G_i represents the ground truth nuclei,
- S_j represents the segmented nuclei,
- The numerator is the sum of intersections between matched ground truth and segmented nuclei,
- The denominator is the sum of their unions, plus any unmatched segmented nuclei (false positives).

Pseudo-code for calculation [18]: Algorithm 2,

Algorithm 2 Computing Aggregated Jaccard Index (AJI)

```
Input: A set of images with annotated nuclei $G_i$ indexed by $i$, and segmented nuclei $S_k$ indexed by $k$
Output: Aggregated Jaccard Index $A$
Initialize overall correct and union pixel counts: $C \leftarrow 0$, $U \leftarrow 0$
for each ground truth nucleus $G_i$ do
    $j \leftarrow \arg\max_k \left(\frac{|G_i \cap S_k|}{|G_i \cup S_k|}\right)$
    Update pixel counts: $C \leftarrow C + |G_i \cap S_j|$, $U \leftarrow U + |G_i \cup S_j|$
    Mark $S_j$ as used
end for
for each segmented nucleus $S_j$ do
    if $S_j$ is not used then
        $U \leftarrow U + |S_j|$
    end if
end for
$A \leftarrow C/U$
return $A$
```

In the case where multiple segmented nuclei have tied Jaccard indices for the same ground truth nucleus, the one with the largest intersection with the ground truth nucleus is selected.

3.4.3 About Panoptic Quality (PQ) [31]

Panoptic Quality (PQ) is a metric designed to evaluate panoptic segmentation—a term that refers to the attempt to aggregate semantic segmentation and instance segmentation into a single task, where certain classes are considered with semantic meaning (not countable), and others with instance-level meaning (countable). It is defined as follows:

$$PQ_{c,i} = \frac{\sum_{(g_k,p_l)\in \mathrm{TP}} \mathrm{IoU}(g_k,p_l)}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}$$

It can be decomposed into:
Recognition Quality (RQ) and Segmentation Quality (SQ):

$$RQ_{c,i} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}$$

$$SQ_{c,i} = \frac{\sum_{(g_k,p_l)\in \mathrm{TP}} \mathrm{IoU}(g_k,p_l)}{|\mathrm{TP}|}$$

So PQ will be:

$$PQ_{c,i} = SQ_{c,i} \times RQ_{c,i}$$

In the evaluation of frameworks that involve instance segmentation and classification for digital pathology, PQ is often employed. However, in this Master Thesis project PQ was not considered in the evaluation of metrics for selecting the optimal instance segmentation network configuration, but rather for potential comparisons with the state of the art. This is because, as demonstrated in [31], the use of this metric is inappropriate for several reasons.

The first reason lies in the fundamental difference between an instance segmentation task with concurrent classification and a panoptic segmentation task in the context of digital pathology.

Finally, the interpretation of results is further complicated by this PQ, as information of various types and of individual significance is aggregated into a single, more confusing index: the definition of PQ as a product of SQ and RQ (both ranging between 0 and 1) means that equal percentage variations in SQ or RQ will scale PQ in the same way. However, equal percentage variations in these quantities do not have the same significance in terms of error: SQ is related to object size, while RQ is related to class distribution, and an error of a given percentage in the first (SQ) is less significant than the same percentage error in the second (RQ). Therefore, the meaning of changes in SQ or RQ is obscured by their multiplication to generate PQ.

3.4.4 Classification Metrics

Classification performance was assessed through per-class Precision, Recall (see 3.4.1) and Specificity.

The formula for Specificity is:

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}}$$

It measures the ability to correctly identify negative cases, avoiding their misclassification into positives.

Chapter 4

Inference Results

4.1 Metrics Evaluation Results: HoVerNet

Hereby, the results of the metrics listed in Section 3.4 for the best HoVerNet training configurations for each dataset are reported. This quantitative analysis not only allows for the assessment of the absolute quality of the automatic segmentation but also helps identify its weaknesses and formulate strategies for resolution. When "Border" is indicated, it is always referred to Version 2 of the border generation.

4.1.1 MoNuSeg

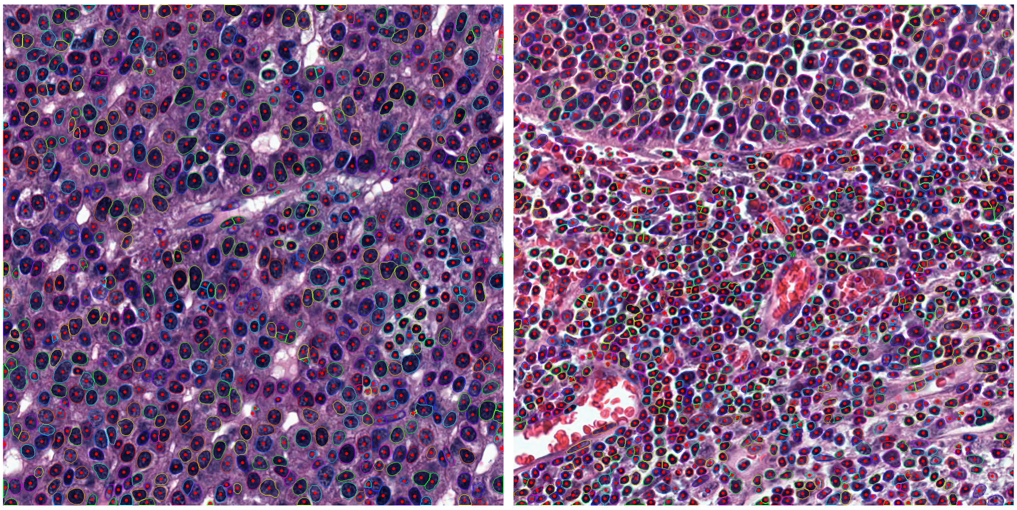


Figure 4.1: Examples of Segmetation on MoNuSeg Tiles from Test Sets 1 & 2 with Config. 3, Border Prediction and Softmax-Argmax Post Processing. Respectively: AJI = 0.7438 and AJI = 0.6457

Here TEST set is the union of TEST 1 and TEST 2.

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8429	0.8643	0.8531	0.8870	0.9163	0.9004	5.7063	0.6690
VAL	0.7851	0.9049	0.8392	0.8520	0.8583	0.8524	5.3236	0.6257
TEST	0.7844	0.8791	0.8231	0.8523	0.8961	0.8716	6.3361	0.6282

Table 4.1: Config. 3 – No Border – HoVer Post Processing – last checkpoint

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8655	0.8045	0.8320	0.8917	0.8905	0.8886	5.8210	0.6484
VAL	0.8126	0.8407	0.8228	0.8554	0.8407	0.8440	5.2374	0.6181
TEST	0.8098	0.8399	0.8195	0.8599	0.8946	0.8739	6.1805	0.6337

Table 4.2: Config. 3 – Border – Softmax-Argmax Post Processing – last checkpoint

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8631	0.8041	0.8302	0.8890	0.8922	0.8874	5.8227	0.6462
VAL	0.8115	0.8372	0.8204	0.8548	0.8406	0.8432	5.0978	0.6164
TEST	0.8068	0.8420	0.8189	0.8536	0.8987	0.8729	6.1034	0.6337

Table 4.3: Config. 4 – Border – Softmax-Argmax Post Processing – 50th checkpoint

Cfr. with State of Art

Study	**AJI**	**Hausdorff Distance**	**Dice Object**	**Dice Pixel**	**Recall**	**Precision**
Kumar et al. (2017)	0.5083	7.6615	0.7623	0.8267	–	–
Shen et al. (2019)	0.6110	–	–	0.829	0.833	0.826
Xie et al. (2020)	0.5664	–	–	–	–	–
Liu et al. (2021)	0.5803	8.2847	–	0.8022	–	–

Table 4.4: MoNuSeg Segmentation Performance Metrics from Various Studies

4.1.2 Giemsa Dataset

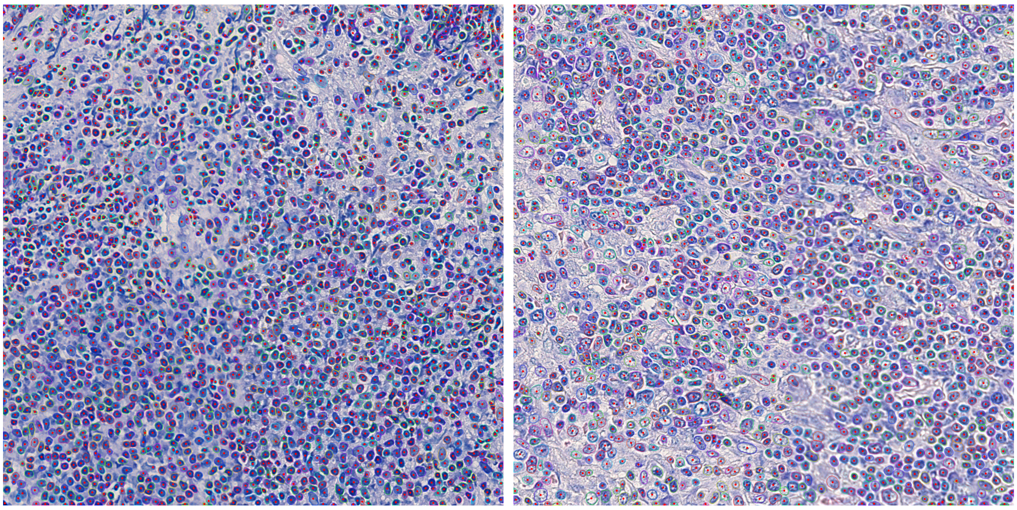


Figure 4.2: Examples of Segmetation on Giemsa Dataset Tiles from Test Set with Config. 3, Border Prediction and Softmax-Argmax Post Processing, 35th checkpoint. Respectively: AJI = 0.6762 and AJI = 0.6994

Subset	Pixel			Object				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8751	0.8655	0.8697	0.8900	0.8942	0.8906	4.5765	0.6914
VAL	0.9086	0.8797	0.8934	0.9230	0.9130	0.9165	4.3607	0.7184
TEST	0.8899	0.7745	0.8260	0.9130	0.8337	0.8695	5.1917	0.6276

Table 4.5: Config. 1 – No Border – HoVer Post Processing - 40th checkpoint

Subset	Pixel			Object				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8727	0.8661	0.8687	0.8803	0.8985	0.8876	4.6191	0.6885
VAL	0.9077	0.8818	0.8942	0.9179	0.9156	0.9153	4.3829	0.7172
TEST	0.8888	0.7878	0.8336	0.9068	0.8532	0.8776	5.1410	0.6377

Table 4.6: Config. 3 – No Border – HoVer Post Processing - 35th checkpoint

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8481	0.8929	0.8693	0.8766	0.9018	0.8878	4.4146	0.6920
VAL	0.8875	0.9013	0.8938	0.9131	0.9226	0.9170	4.0654	0.7265
TEST	0.8693	0.8029	0.8321	0.9097	0.8457	0.8746	4.8750	0.6385

Table 4.7: Config. 3 – Border – Softmax-Argmax Post Processing - last checkpoint

Debug Figures

As Giemsa Dataset is a private one and there are no other reference models in the literature trained on Giemsa-stained images, using debug figures can serve as a valuable tool for assessing the achieved performance and identifying areas for potential improvement. The following images are 500 × 500 sub-tiles extracted from the test set. On the left, there is an overlay of the sub-tile and the ground truth in red. On the right, the ground truth is shown in red, while the contours of the automatic segmentation output from the network are displayed in green. The configuration used for this inference is Config. 3, comprising border prediction, the 35th checkpoint was chosen and the post processing was brought on with Softmax - Argmax strategy.

In Figure 4.3, we observe an example of over-segmentation. The network segments structures (chromatin thickenings) that are not nuclei, likely due to a lack of examples in the training set to distinguish between nuclei and such structures. This issue could be addressed by including more examples of these patterns in the training set to teach the network to distinguish them from nuclei.

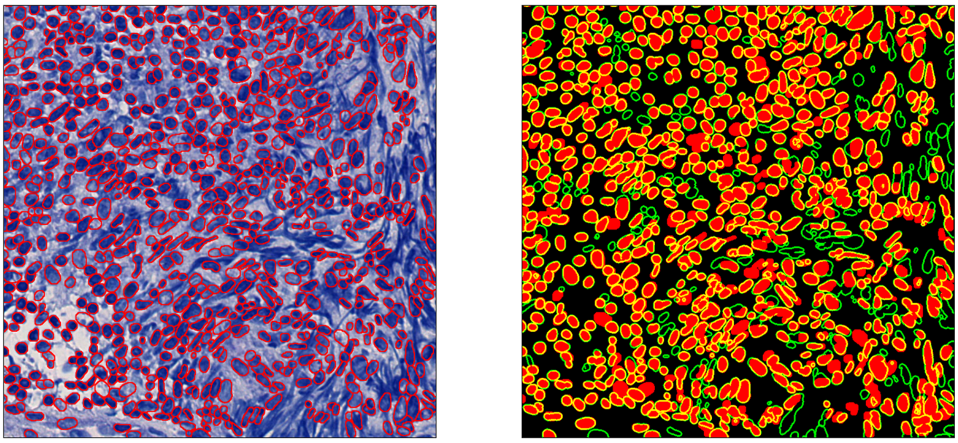


Figure 4.3: Debug Figure #1

In Figure 4.4, we observe a case of apparent over-segmentation; however, the network is correctly segmenting instances that were mistakenly left unsegmented in the manual ground truth. This issue can be resolved by refining the manual annotations. Nonetheless, this example demonstrates the network's ability to generalize its learned knowledge from the training phase.

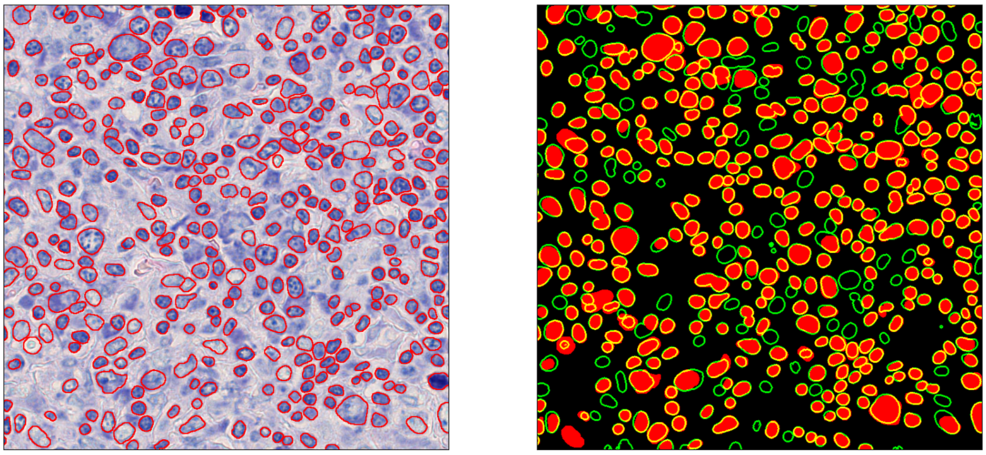


Figure 4.4: Debug Figure #2

In Figure 4.5, we observe a case of apparent under-segmentation. Within the orange circle, instances are marked that were incorrectly segmented manually by the operator. The network, correctly generalizing from its training, does not segment these instances during inference. This issue, once again, can be addressed by refining the annotations.

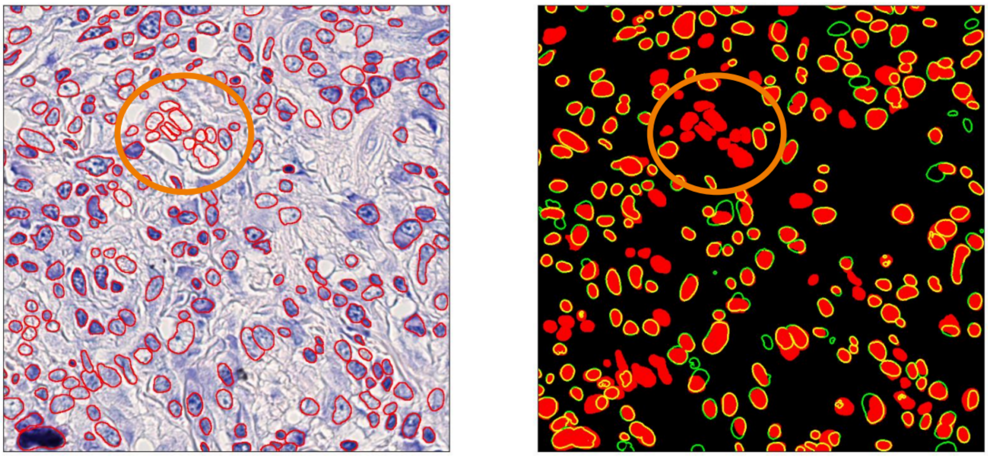


Figure 4.5: Debug Figure #3

Finally, in Figure 4.6, we present an example of successful segmentation. This example belongs to a patch with an AJI score of 0.6764.

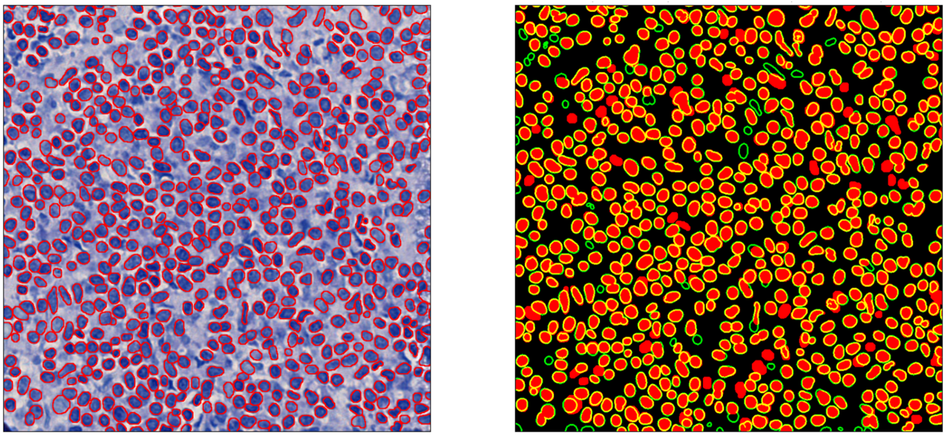


Figure 4.6: Debug Figure #4

4.1.3 BreastIHC

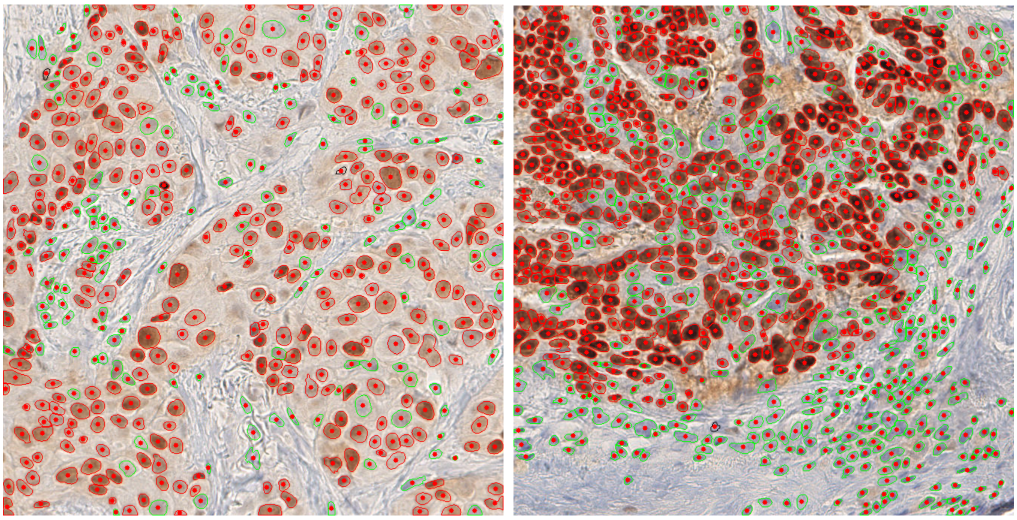


Figure 4.7: Examples of Segmetation and Classification on BreastIHC Tiles from Test Set with Config. 3. Respectively: AJI = 0.6249 and AJI = 0.5153

Subset	Pixel			Object				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8354	0.7887	0.8092	0.9100	0.8592	0.8801	3.4903	0.6428
VAL	0.8339	0.8213	0.8260	0.8994	0.8791	0.8865	3.4034	0.6610
TEST	0.8198	0.8139	0.8125	0.8929	0.8747	0.8771	3.7501	0.6338

Table 4.8: Config. 6 – No Border – HoVerNet Post Processing – last checkpoint

Cfr. with another Master Thesis work

Our results surpass the ones obtained in [29] using CDNet on the same dataset, without classification.

Subset	Pixel			Object				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8577	0.7223	0.7797	0.8785	0.8677	0.8677	3.8807	0.6037
VAL	0.8557	0.7701	0.8085	0.8696	0.8913	0.8777	3.7310	0.6304
TEST	0.8390	0.7577	0.7913	0.8568	0.8896	0.8661	4.0695	0.6074

Table 4.9: CDNet trained on BreastIHC dataset in [29] Master Thesis

For this dataset, as well as for the others where classification was performed, metrics are also reported for each class.

Subset	Pixel			Object				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8245	0.7910	0.7794	0.8899	0.8334	0.8289	3.2886	0.6325
VAL	0.7871	0.8616	0.7886	0.8491	0.8896	0.8310	3.3001	0.6447
TEST	0.8017	0.8126	0.7758	0.8627	0.8352	0.8135	3.6525	0.6073

Table 4.10: DAB

Subset	Pixel			Object				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.7887	0.6746	0.7182	0.8695	0.7828	0.8121	3.6856	0.5421
VAL	0.7498	0.6847	0.7092	0.8298	0.8025	0.8066	3.6618	0.5318
TEST	0.7397	0.6739	0.6937	0.8155	0.8049	0.7982	3.9876	0.5119

Table 4.11: HEM

4.1.4 LyNSeC 1

CD3

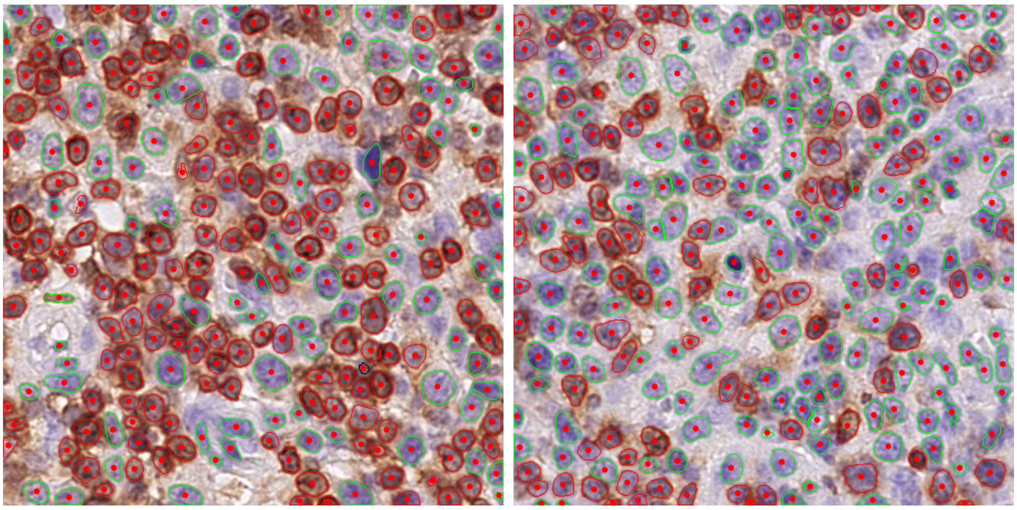


Figure 4.8: Examples of Segmetation and Classification on CD3 Tiles from Test Set with Config. 6. Respectively: AJI = 0.7521 and AJI = 0.7662

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8750	0.8794	0.8763	0.9187	0.9522	0.9342	3.1389	0.7689
VAL	0.8702	0.8729	0.8701	0.9132	0.9462	0.9280	3.1879	0.7600
TEST	0.8600	0.8774	0.8668	0.9132	0.9420	0.9259	3.2427	0.7542

Table 4.12: Config. 6 – No Border – HoVer-Net Post Processing – 35th checkpoint

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8571	0.8551	0.8548	0.9012	0.9334	0.9156	3.2617	0.7392
VAL	0.8466	0.8433	0.8418	0.8962	0.9232	0.9079	3.4191	0.7208
TEST	0.8372	0.8537	0.8434	0.8912	0.9182	0.9026	3.3305	0.7204

Table 4.13: DAB

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8602	0.8702	0.8640	0.8963	0.9316	0.9121	3.0508	0.7513
VAL	0.8523	0.8627	0.8562	0.8840	0.9225	0.9002	3.0071	0.7409
TEST	0.8444	0.8622	0.8508	0.8874	0.9190	0.9008	3.2016	0.7325

Table 4.14: HEM

Ki-67

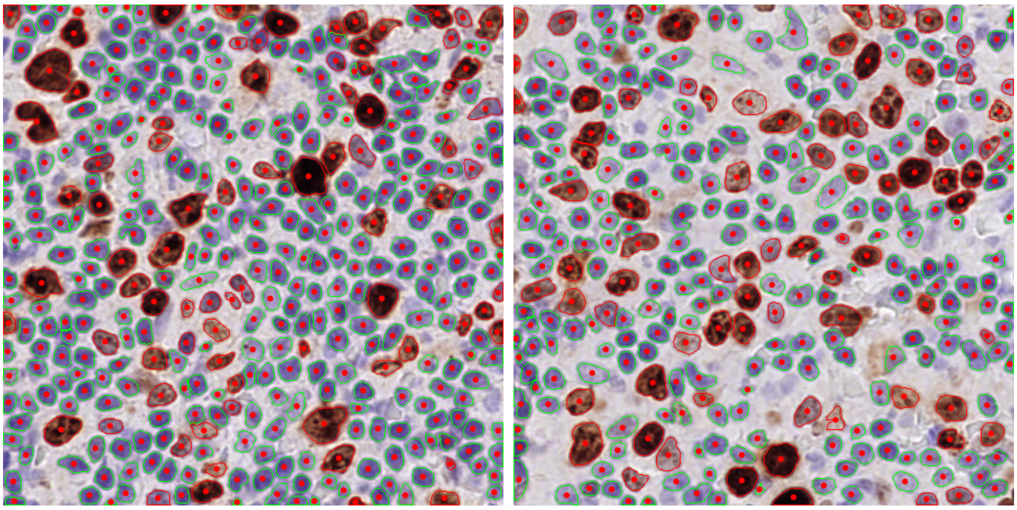


Figure 4.9: Examples of Segmetation and Classification on Ki-67 Tiles from Test Set with Config. 6. Respectively: AJI = 0.7975 and AJI = 0.8273

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.9022	0.9107	0.9061	0.9388	0.9622	0.9500	2.9345	0.8089
VAL	0.8947	0.9104	0.9021	0.9318	0.9607	0.9457	2.9560	0.8022
TEST	0.9137	0.9023	0.9072	0.9462	0.9498	0.9475	2.9705	0.8092

Table 4.15: Config. 6 – No Border – HoVer-Net Post Processing – 35th checkpoint

Cfr. with Ho-Ly Net

The results reported in Table 4.18 for training HoLy-Net on LyNSeC 1, which involved IHC images with CD3, Ki-67, and ERG markers simultaneously, show overall lower performance in terms of pixel-based Dice, object-based Dice, and AJI compared to the results obtained in our experiments. In our approach, models

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8978	0.9129	0.9048	0.9194	0.9542	0.9358	3.1217	0.8107
VAL	0.8793	0.9165	0.8972	0.9009	0.9606	0.9292	3.0235	0.7998
TEST	0.8879	0.9041	0.8952	0.9097	0.9388	0.9232	3.1106	0.7961

Table 4.16: DAB

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8748	0.8764	0.8749	0.9249	0.9404	0.9319	2.8241	0.7614
VAL	0.8734	0.8674	0.8696	0.9188	0.9272	0.9223	2.9035	0.7496
TEST	0.8895	0.8471	0.8658	0.9311	0.9122	0.9197	2.9050	0.7452

Table 4.17: HEM

were trained separately to recognise different types of markers (on-off or membrane markers), yielding superior performance. Although training multiple models incurs higher computational costs, this investment is offset by improved accuracy and the ability to segment different marker types with greater specificity.

Metric	**Value**
Dice Pixel Based	0.864
Dice Object Based	0.895
AJI	0.732

Table 4.18: HoLy-Net results on Lynsec 1 (IHC)

4.1.5 Comments on the results

The inclusion of the border class significantly improves the AJI metric for datasets such as MoNuSeg and Giemsa Dataset, where annotations are challenged by overlapping and overcrowded regions. In these datasets, the border class enhances the model's ability to separate closely packed and overlapping nuclei, leading to a more precise segmentation and effectively reducing errors where instances merge or intersect. This improvement in AJI indicates that the border class is particularly valuable for datasets with high instance density.

Conversely, in datasets like BreastIHC and LyNSeC 1, where instances are already well-separated, the addition of the border class does not yield any noticeable improvement. Since these datasets inherently have less overlap between nuclei or instances, the model's segmentation performance is already robust without the need for explicit border prediction. In these cases, the added complexity of the border class is unnecessary and does not contribute to further segmentation accuracy. This contrast highlights the importance of tailoring model configurations to dataset characteristics, particularly in cases where spatial separation of instances varies.

Overall, the results exceed most of the metrics of the state-of-the-art, and it was achieved using a single deep learning architecture, thus demonstrating its versatility across different tasks and stainings. The architecture itself is modified to address more complex instance segmentation challenges, proving once again to be adaptable and suitable for broad application.

4.2 Metrics Evaluation Results: CDNet

4.2.1 MoNuSeg

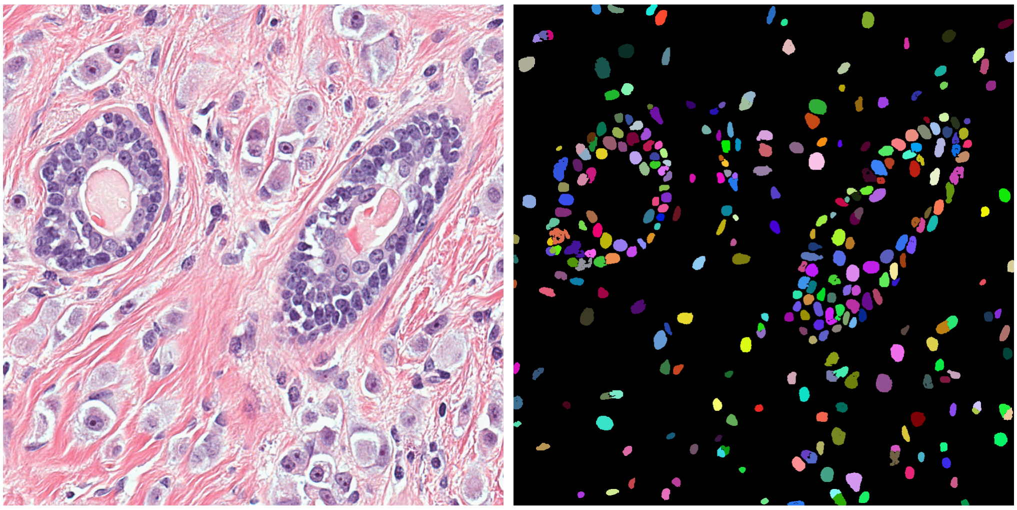


Figure 4.10: Examples of Original Image and corresponding Instance Segmentation with paper post processing

Here TEST set is the union of TEST 1 and TEST 2.

Subset	**Pixel**			**Object**				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.8833	0.7512	0.8104	0.8598	0.9021	0.8767	6.4693	0.6091
VAL	0.8574	0.7905	0.8203	0.8678	0.8579	0.8581	5.3780	0.6085
TEST	0.8704	0.7891	0.8246	0.8711	0.9027	0.8837	6.1940	0.6325

Table 4.19: CDNet training with paper post processing strategy

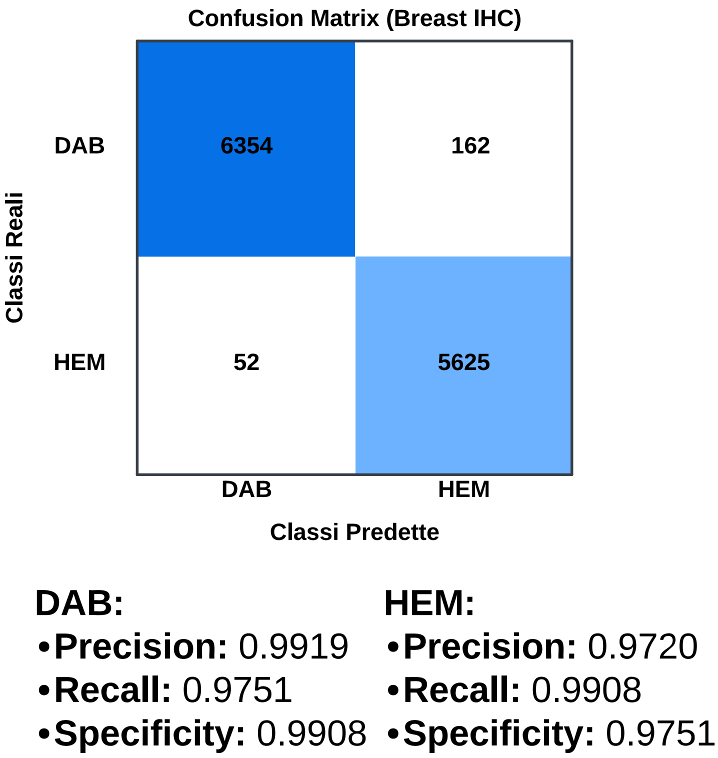
Subset	Pixel			Object				
	Precision	**Recall**	**Dice**	**Precision**	**Recall**	**Dice**	**HD**	**AJI**
TRAIN	0.7578	0.9112	0.8232	0.8536	0.9181	0.8833	6.4872	0.6237
VAL	0.7137	0.9471	0.8133	0.8381	0.8659	0.8490	5.6703	0.6005
TEST	0.7407	0.9230	0.8175	0.8701	0.8911	0.8783	6.6514	0.6136

Table 4.20: CDNet training with Softmax-Argmax post processing strategy

Although CDNet provides satisfactory segmentation results, HoVerNet was preferred due to its versatility, being able to perform both instance segmentation and classification tasks, and its ability to achieve higher AJI scores with the inclusion of border prediction on MoNuSeg with Config. 3.

4.3 Classification Results with HoVerNet

4.3.1 Breast IHC



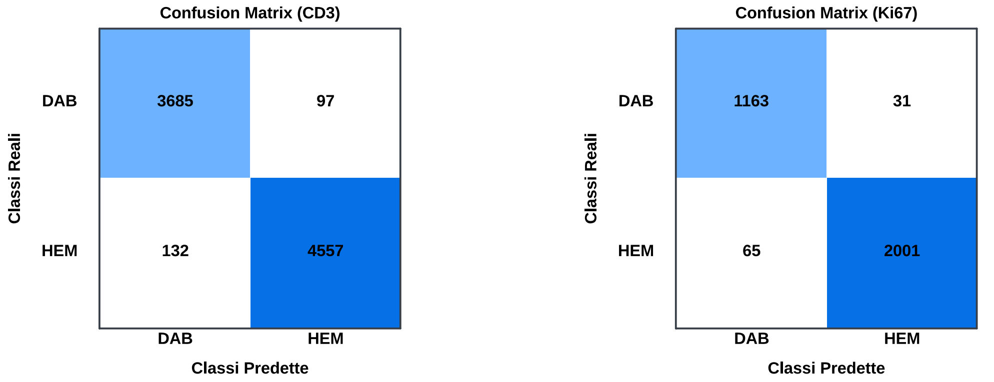
DAB:
- **Precision:** 0.9919
- **Recall:** 0.9751
- **Specificity:** 0.9908

HEM:
- **Precision:** 0.9720
- **Recall:** 0.9908
- **Specificity:** 0.9751

Figure 4.11: Confusion Matrix and Per-Class Metrics, Breast IHC

4.3.2 LyNSeC 1

CD3 and Ki-67



DAB:
- **Precision:** 0.9654
- **Recall:** 0.9744
- **Specificity:** 0.9718

HEM:
- **Precision:** 0.9792
- **Recall:** 0.9718
- **Specificity:** 0.9744

DAB:
- **Precision:** 0.9471
- **Recall:** 0.9740
- **Specificity:** 0.9685

HEM:
- **Precision:** 0.9847
- **Recall:** 0.9685
- **Specificity:** 0.9740

Figure 4.12: Confusion Matrix and Per-Class Metrics: CD3 (left), Ki-67 (right).

Chapter 5

Transfer Learning

5.1 Knowledge Transfer Across Datasets

5.1.1 Trained Models

Transfer learning is a strategy aimed at transferring knowledge from one task to a second task. In our case, transfer learning is not used to boost the performance of another model, as is typically done, but rather to transfer knowledge learned from one dataset to perform inference (instance segmentation and classification) on a different dataset of the same type but from a different source.

Specifically, the goal is to segment and classify cell nuclei in WSI stained with IHC markers from Giemsa Dataset, without being able to train a dedicated network due to the lack of annotations. To achieve this, we use a similar dataset, LyNSec 1, with which two networks have been trained, one with the CD3 marker and the other with the Ki-67 marker.

The CD3 marker is a membrane-type marker, meaning that the DAB stain binds to positive cells at the cell wall. Ki-67, on the other hand, is an on-off marker, also known as a nuclear marker, as it binds to the cell at the nucleus level.

After training these networks, knowledge transfer was performed by running inference on the tiles from a patient in Giemsa Dataset to segment and classify immunopositive and immunonegative nuclei. The patient used for this inference was selected for having a balanced number of IHC-stained slides with on-off and membrane markers. Additionally, the chosen patient has some more cytoplasmic-specific markers, which are listed in Table 5.1.

For each WSI of every marker listed in Table 5.1, 50 tiles were initially selected, from which a subset was chosen to retain tiles that were informative regarding the patient's immunopositivity/negativity status, resulting in a variable number of tiles per marker, ranging from 18 to 28.

Inference for membrane or cytoplasmic markers was performed using the model

Marker Type	**Markers**
Membrane Markers	CD3, CD10, CD20, PD1
On-Off Markers	BCL6, CD138, Ki-67, MUM1
Cytoplasm Markers	CD21-3, EBER, CXCL13

Table 5.1: Marker Classification for Patient #1

trained on the LyNSeC 1 – CD3 dataset, while the network trained on Ki-67 was used for inference on on-off markers.

5.1.2 Statistics

An hand-crafted function calculates and displays key statistics on nuclei segmentation and classification in an image, using data from a JSON file.

First, it counts the nuclei instances for two types: hematoxylin-stained (Hem) and DAB-stained (Dab), giving an overview of the distribution of each type in the image.

Next, the function calculates the total pixel count for each type by summing the pixels within the contours of all nuclei for each staining type. This gives the values for the total Hem pixels (Total Hem Pixels) and total Dab pixels (Total Dab Pixels).

The Positivity Score is then computed as the ratio of DAB-stained pixels to the total number of pixels (both Hem and Dab combined), which is then expressed as a percentage:

$$\text{Positivity Score} = \frac{\text{Total Dab Pixels}}{\text{Total Hem Pixels} + \text{Total Dab Pixels}} \times 100$$

This score indicates the level of immunopositivity in the image, providing a quantitative measure of DAB staining.

Using the image resolution of 0.2208 microns per pixel, the function converts pixel counts into area measurements in square microns. The conversion factor for each pixel to square microns is computed as:

$$\text{Micron Per Pixel}^2 = (0.2208)^2 \approx 0.0487\ \mu m^2$$

Using this factor, the DAB area and total area in square microns are calculated as follows:

$$\text{Dab Area } (\mu m^2) = \text{Total Dab Pixels} \times 0.0487$$

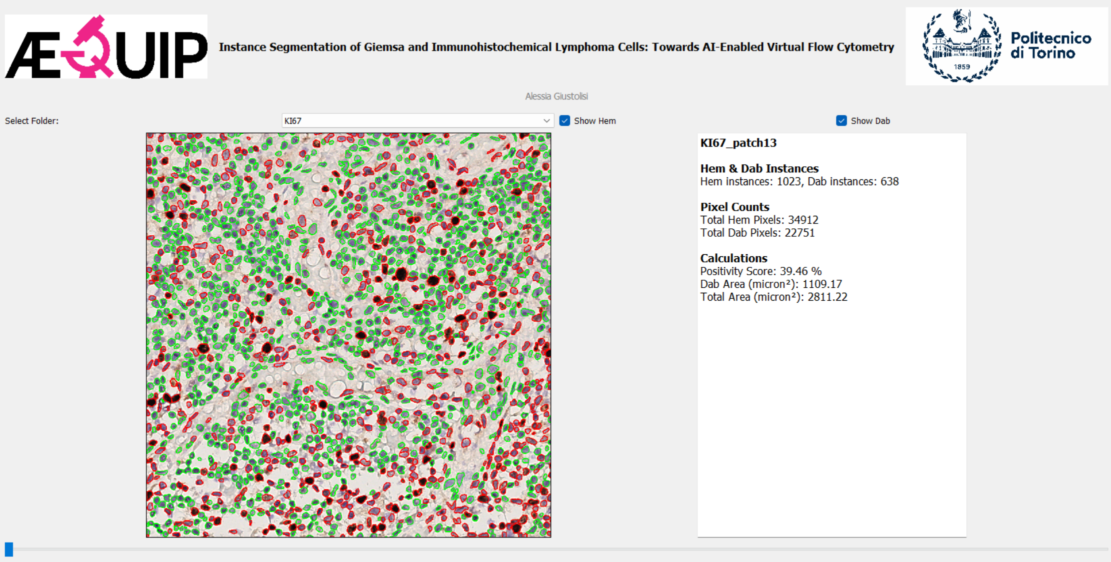
$$\text{Total Area } (\mu m^2) = (\text{Total Hem Pixels} + \text{Total Dab Pixels}) \times 0.0487$$

These area calculations allow for a spatial interpretation of the segmentation results in biologically relevant units.

Finally, this data, including positivity score, instance counts, pixel counts, and area measurements, is displayed within a GUI (Graphical User Interface, see 5.2.1). This provides users with a comprehensive view of immunopositivity distribution, staining type presence, and the spatial characteristics of segmented nuclei within each analyzed file.

5.2 Qualitative Results

5.2.1 Graphical User Interface



Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Alessia Giustolisi

Select Folder: KI67

☑ Show Hem

☑ Show Dab

KI67_patch13

Hem & Dab Instances
Hem instances: 1023, Dab instances: 638

Pixel Counts
Total Hem Pixels: 34912
Total Dab Pixels: 22751

Calculations
Positivity Score: 39.46 %
Dab Area (micron^2): 1109.17
Total Area (micron^2): 2811.22

Figure 5.1: Graphical User Interface

As previously mentioned, quantitative validation of the results is not feasible due to the lack of manual ground truth annotations. Consequently, a qualitative evaluation of the results was performed, and to streamline the process, a GUI was developed. In the GUI, the user can select the marker of interest and navigate through the tiles for that marker. Additionally, using overlay toggles, the user can choose to display the segmentation of instance boundaries separately for immunopositive and immunonegative cells. Quantitative metrics presented in Section 5.1.2 are also displayed within the interface.

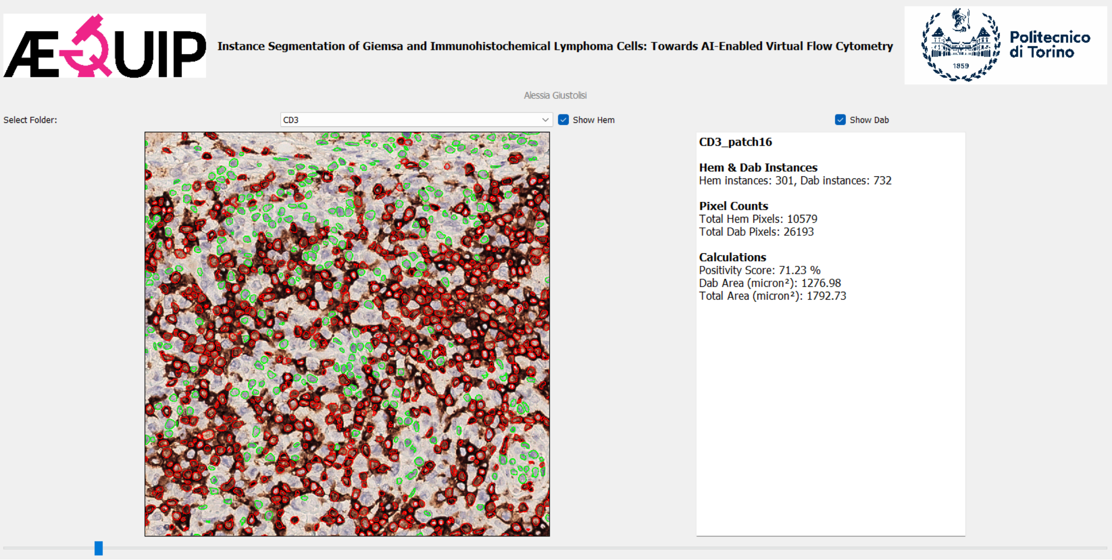
5.2.2 Overlays and Considerations

Membrane Markers

The results for membrane markers CD3, CD20, and PD1 are satisfactory, while those for CD10 are suboptimal. This may be due to a predominance of immunonegative cells, stained with haematoxylin, whose blue shade differs from what the network encountered during training. Consequently, the network struggles to generalize knowledge to these images.

This difficulty highlights a broader phenomenon in deep learning where a network is trained to recognize specific staining characteristics and color variations provided by the training dataset. When the test images exhibit a different staining intensity or shade not covered in the training set, the network may fail to generalize well. This limitation stems from the epistemic uncertainty of the model. Epistemic uncertainty, also referred as model uncertainty, reflects the network's lack of knowledge on data not seen during training, including variations in color or staining.

A practical approach to overcome this issue is to normalize the color of test images to match a target staining profile. Color normalization methods adjust the shades and intensities in test images to be consistent with the training data color distribution, thereby reducing the gap between the two datasets. This can significantly improve the model's robustness by reducing epistemic uncertainty and helping the network better recognize cells under a variety of staining conditions.



Select Folder: CD3

☑ Show Hem

☑ Show Dab

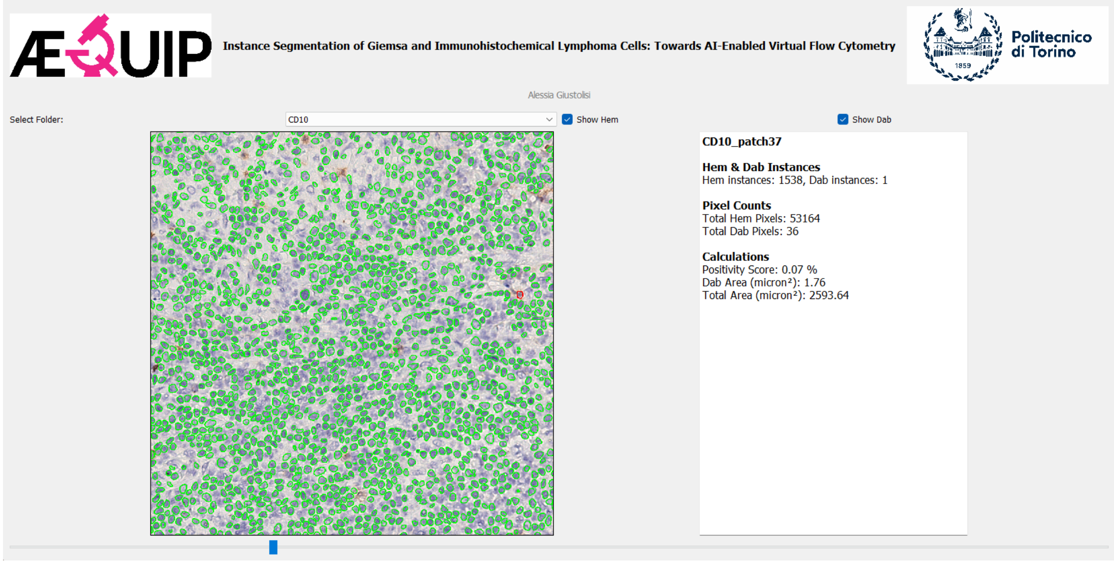
CD3_patch16

Hem & Dab Instances
Hem instances: 301, Dab instances: 732

Pixel Counts
Total Hem Pixels: 10579
Total Dab Pixels: 26193

Calculations
Positivity Score: 71.23 %
Dab Area (micron^2): 1276.98
Total Area (micron^2): 1792.73

Figure 5.2: CD3



Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Alessia Giustolisi

Select Folder: CD10 ☑ Show Hem ☑ Show Dab

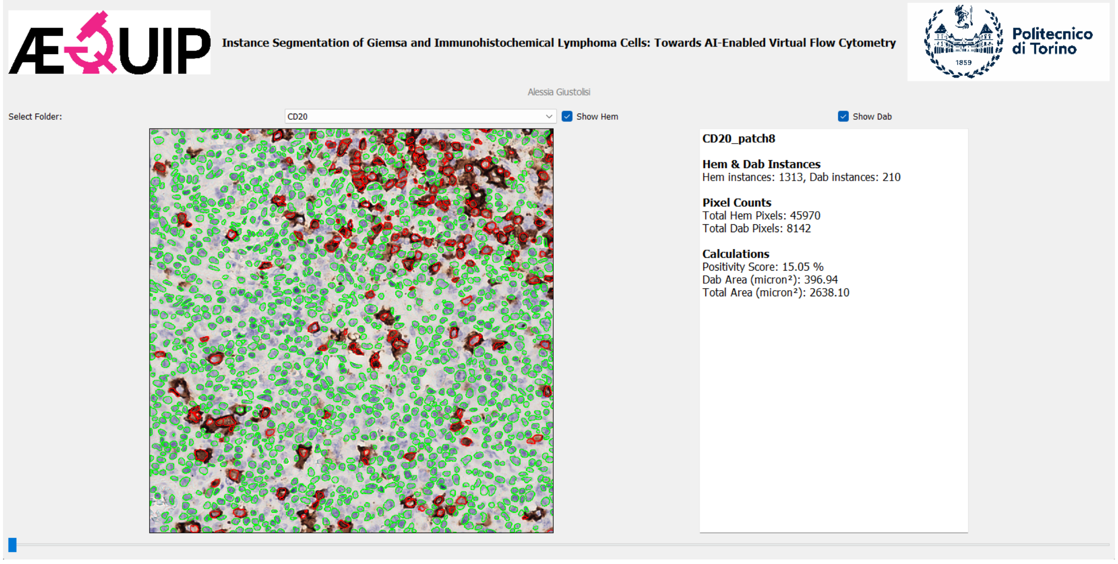
CD10_patch37

Hem & Dab Instances
Hem instances: 1538, Dab instances: 1

Pixel Counts
Total Hem Pixels: 53164
Total Dab Pixels: 36

Calculations
Positivity Score: 0.07 %
Dab Area (micron²): 1.76
Total Area (micron²): 2593.64

Figure 5.3: CD10



Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Alessia Giustolisi

Select Folder: CD20 ☑ Show Hem ☑ Show Dab

CD20_patch8

Hem & Dab Instances
Hem instances: 1313, Dab instances: 210

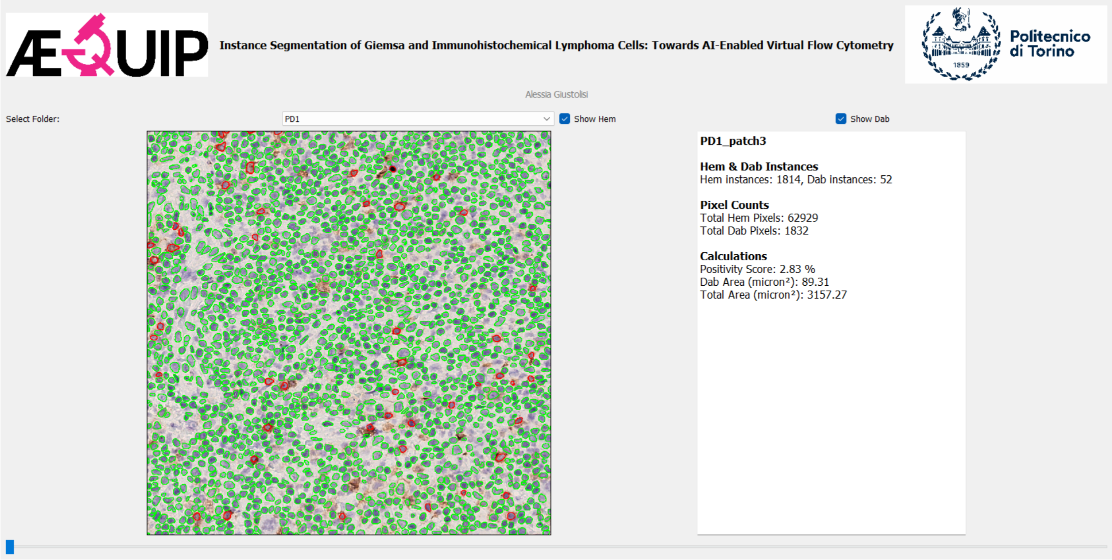
Pixel Counts
Total Hem Pixels: 45970
Total Dab Pixels: 8142

Calculations
Positivity Score: 15.05 %
Dab Area (micron²): 396.94
Total Area (micron²): 2638.10

Figure 5.4: CD20

On-Off Markers

The results for BCL6, CD138, Ki-67, and MUM1, which are nuclear markers, are also satisfactory. In general, this task is simpler than membrane marker segmentation, as the DAB staining is confined within the cell nucleus and not dispersed in the surrounding area. The challenge of identifying cells stained with



ÆQUIP Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Politecnico di Torino

Alessia Giustolisi

Select Folder: PD1 ☑ Show Hem ☑ Show Dab

PD1_patch3

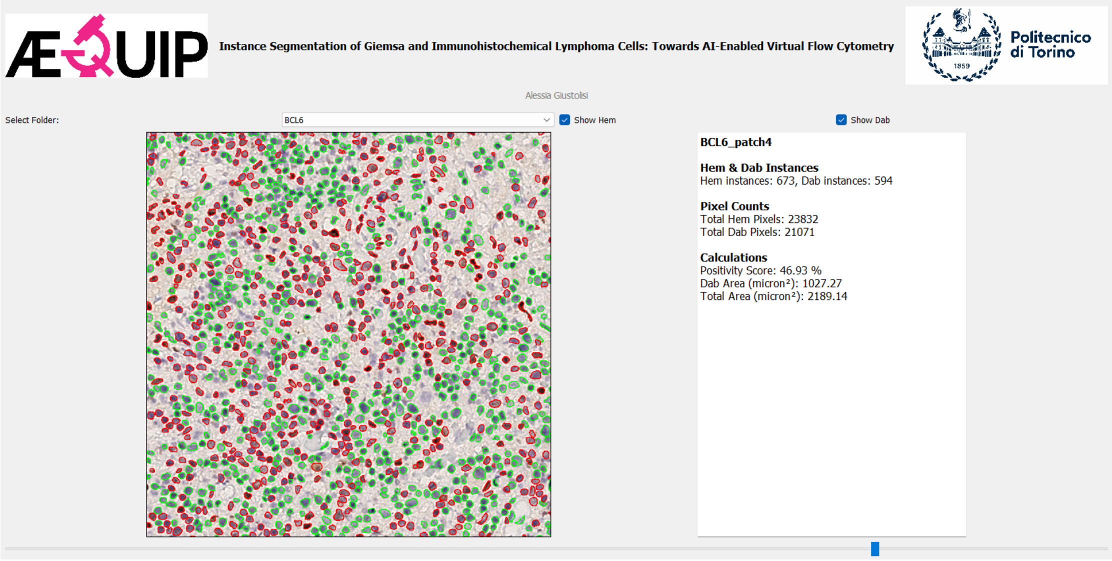
Hem & Dab Instances
Hem instances: 1814, Dab instances: 52

Pixel Counts
Total Hem Pixels: 62929
Total Dab Pixels: 1832

Calculations
Positivity Score: 2.83 %
Dab Area (micron^2): 89.31
Total Area (micron^2): 3157.27

Figure 5.5: PD1

haematoxylin remains but is less pronounced compared to membrane markers.



ÆQUIP Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Politecnico di Torino

Alessia Giustolisi

Select Folder: BCL6 ☑ Show Hem ☑ Show Dab

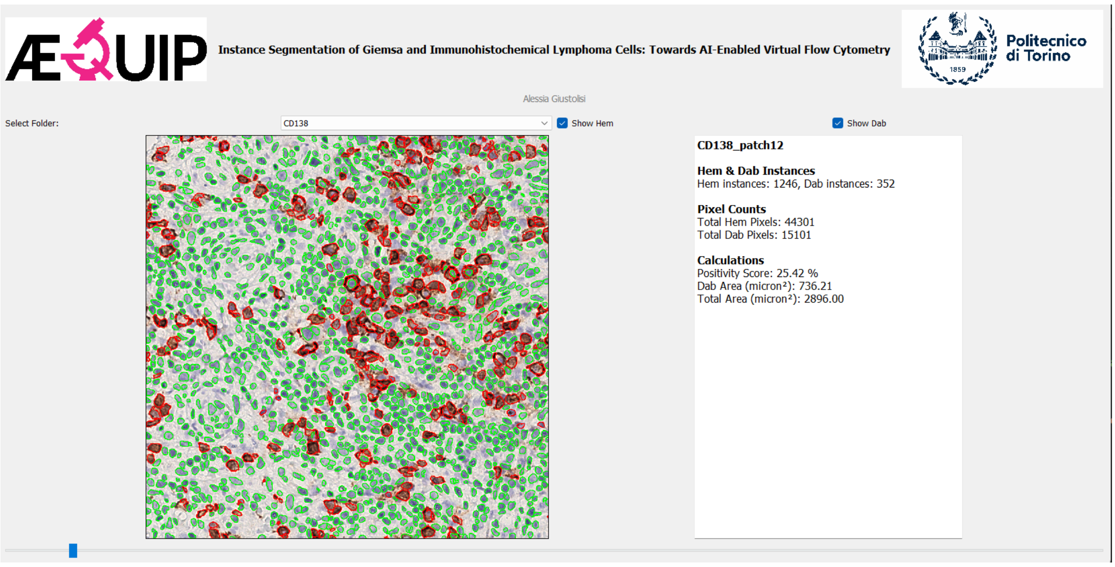
BCL6_patch4

Hem & Dab Instances
Hem instances: 673, Dab instances: 594

Pixel Counts
Total Hem Pixels: 23832
Total Dab Pixels: 21071

Calculations
Positivity Score: 46.93 %
Dab Area (micron^2): 1027.27
Total Area (micron^2): 2189.14

Figure 5.6: BCL6



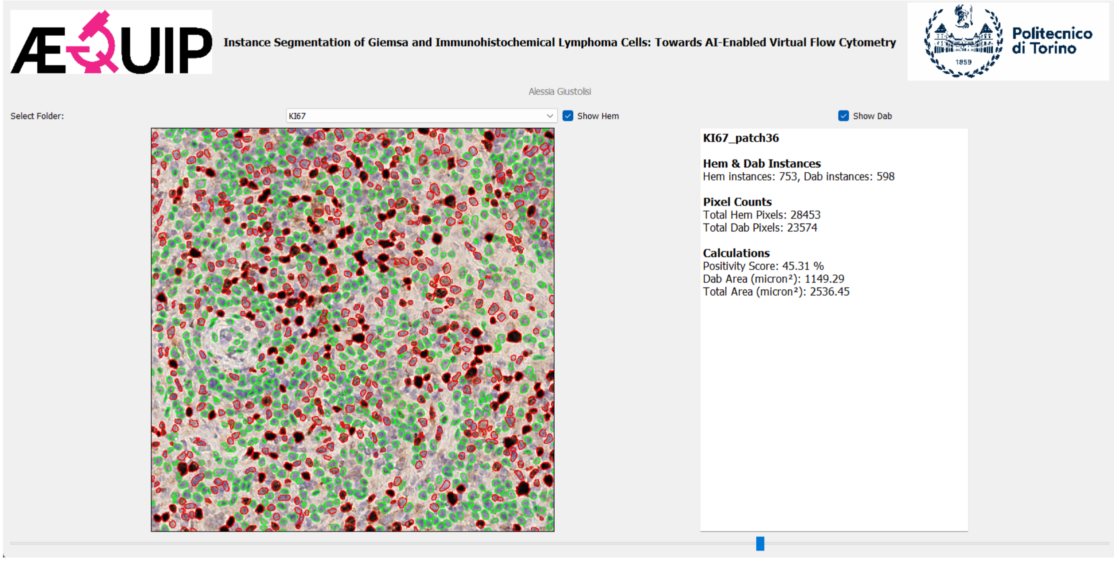
CD138_patch12

Hem & Dab Instances
Hem instances: 1246, Dab instances: 352

Pixel Counts
Total Hem Pixels: 44301
Total Dab Pixels: 15101

Calculations
Positivity Score: 25.42 %
Dab Area (micron^2): 736.21
Total Area (micron^2): 2896.00

Figure 5.7: CD138



KI67_patch36

Hem & Dab Instances
Hem instances: 753, Dab instances: 598

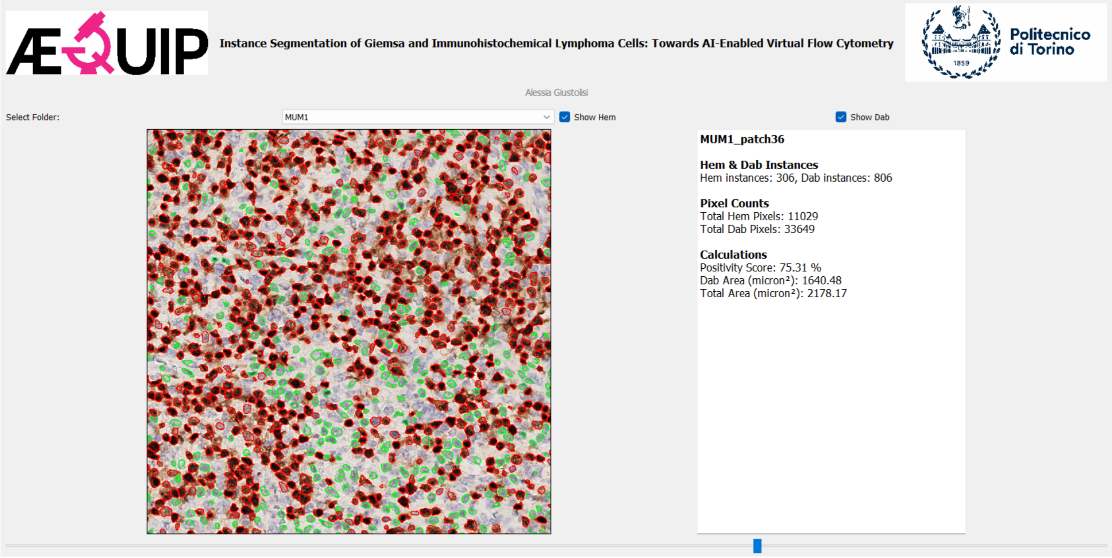
Pixel Counts
Total Hem Pixels: 28453
Total Dab Pixels: 23574

Calculations
Positivity Score: 45.31 %
Dab Area (micron^2): 1149.29
Total Area (micron^2): 2536.45

Figure 5.8: Ki-67

Cytoplasm Markers

Finally, for cytoplasmic markers such as CD21-3, EBER, and CXCL13, the results are poor. This outcome is expected, as inference was performed using a network trained to recognize membrane markers, not cytoplasmic ones (CD3). It is evident



Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Alessia Giustolisi

Select Folder: MUM1

☑ Show Hem ☑ Show Dab

MUM1_patch36

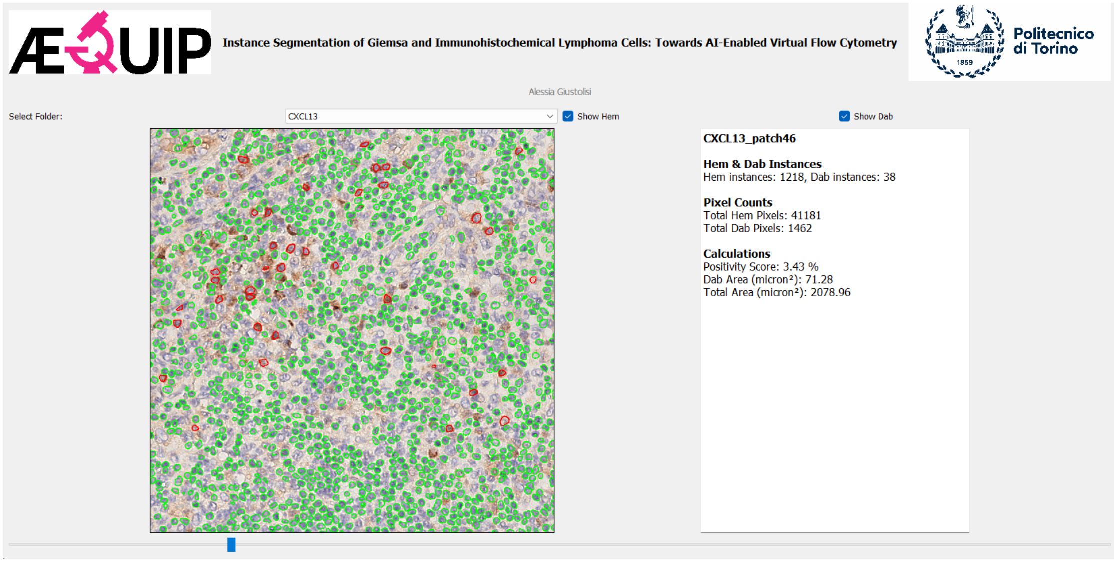
Hem & Dab Instances
Hem instances: 306, Dab instances: 806

Pixel Counts
Total Hem Pixels: 11029
Total Dab Pixels: 33649

Calculations
Positivity Score: 75.31 %
Dab Area (micron²): 1640.48
Total Area (micron²): 2178.17

Figure 5.9: MUM1

that not only is recognizing haematoxylin problematic due to the issues previously described, but in some cases, even immunopositive cells are not correctly identified. A potential solution would be to train a network specifically to recognize strictly cytoplasmic markers. In any case, the traditional calculation of immunopositivity on images stained with cytoplasmic markers is carried out as a percentage of total immunopositivity, therefore without the need to divide between instances.



Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Alessia Giustolisi

Select Folder: CXCL13

☑ Show Hem ☑ Show Dab

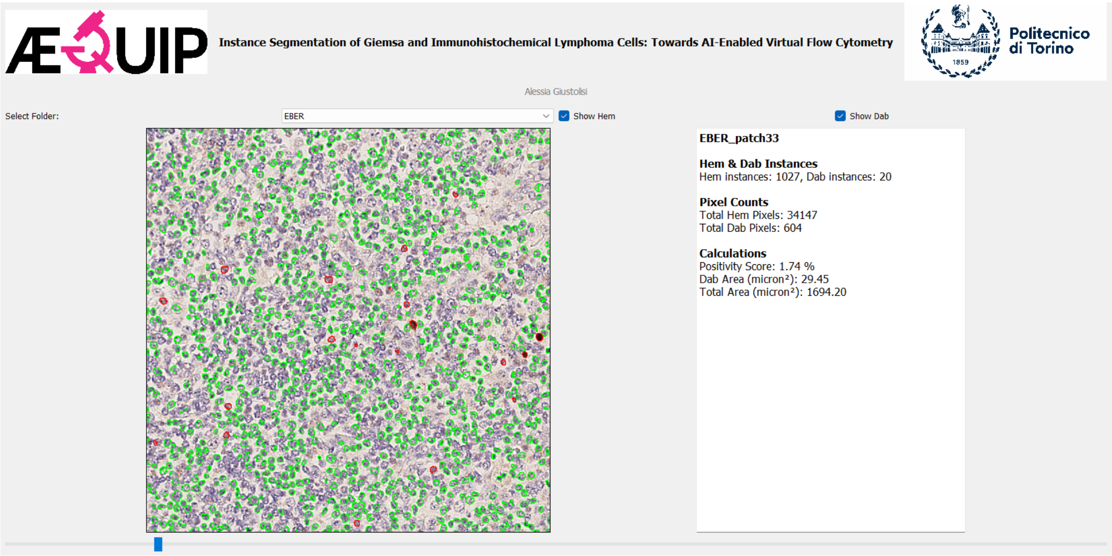
CXCL13_patch46

Hem & Dab Instances
Hem instances: 1218, Dab instances: 38

Pixel Counts
Total Hem Pixels: 41181
Total Dab Pixels: 1462

Calculations
Positivity Score: 3.43 %
Dab Area (micron²): 71.28
Total Area (micron²): 2078.96

Figure 5.10: CXCL13



Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Alessia Giustolisi

Select Folder: EBER

☑ Show Hem

☑ Show Dab

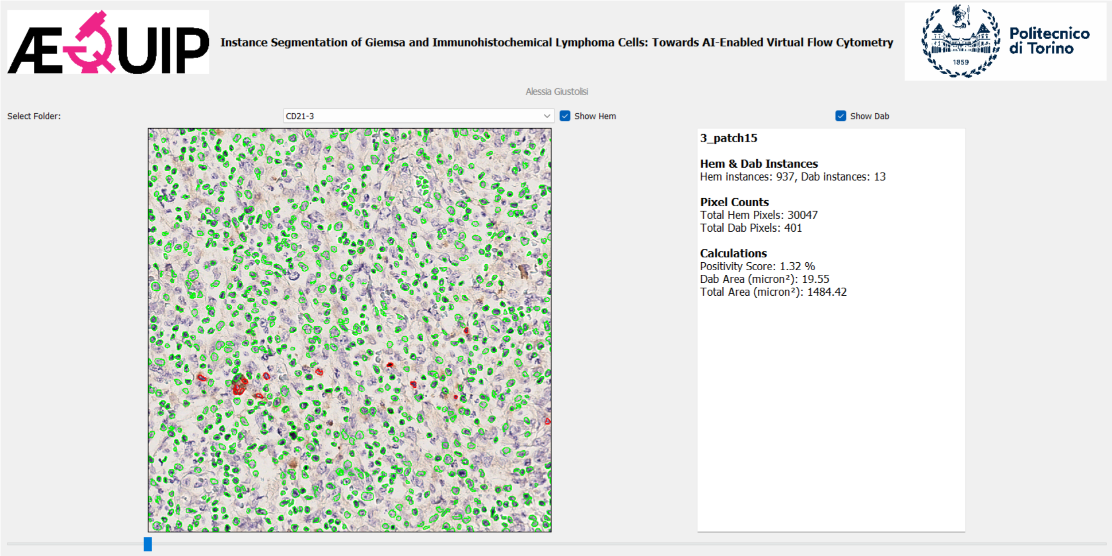
EBER_patch33

Hem & Dab Instances
Hem instances: 1027, Dab instances: 20

Pixel Counts
Total Hem Pixels: 34147
Total Dab Pixels: 604

Calculations
Positivity Score: 1.74 %
Dab Area (micron²): 29.45
Total Area (micron²): 1694.20

Figure 5.11: EBER



Instance Segmentation of Giemsa and Immunohistochemical Lymphoma Cells: Towards AI-Enabled Virtual Flow Cytometry

Alessia Giustolisi

Select Folder: CD21-3

☑ Show Hem

☑ Show Dab

3_patch15

Hem & Dab Instances
Hem instances: 937, Dab instances: 13

Pixel Counts
Total Hem Pixels: 30047
Total Dab Pixels: 401

Calculations
Positivity Score: 1.32 %
Dab Area (micron²): 19.55
Total Area (micron²): 1484.42

Figure 5.12: CD21-3

Chapter 6

Conclusions and Further Developments

6.1 Summary of Findings

In summary, we have successfully developed models, using a single architecture, that equal or surpass the state-of-the-art in instance segmentation and classification across various datasets and stainings, as well as images at different magnification levels. This architecture can be adapted to predict or omit the border class based on the dataset's specific characteristics and nuclear density, allowing for customizable post-processing accordingly.

Additionally, we demonstrated that by training networks on specific classes of IHC markers, we can generalize the learning to unseen datasets, paving the way toward a tool capable of parallel analysis of different stainings and markers. This would support pathologists by enabling them to assess the clinical status of the patient with greater precision and speed.

6.2 Limits and Future Objectives

Clearly, there are limitations to this study, which open the door to future developments aimed at further optimizing results.

6.2.1 Giemsa Nuclei Classification

Currently, no publicly available Giemsa-stained datasets exist with annotations for instance segmentation and classification; Giemsa Dataset includes annotations only for instance segmentation. This is a limitation, as Giemsa staining is particularly effective for distinguishing between different classes of lymphocytes, meaning that valuable information is left untapped when limiting analysis to instance segmentation alone without distinguishing between nuclei types.

Specifically, nuclei could be differentiated into classes such as small and uniform, large and uniform, small with scattered chromatin and nucleoli, or large with numerous nucleoli. Access to such detailed annotations would enable HoVerNet to be trained for classification tasks as well, sparing pathologists the manual classification of nuclei into these categories and greatly enhancing the clinical utility of the segmentation tool.

6.2.2 Registration of WSIs

A current limitation of this approach is the lack of registration between consecutive WSI images of the same patient, meaning there is no alignment in the z-axis direction across successive WSIs. Such a task is highly complex and requires advanced mathematical tools as well as significant computational power. However, the ability to align slides would enable identification of the same cell nucleus across consecutive slices, allowing for a combined analysis of Giemsa staining patterns and nuclear immunopositivity/negativity across multiple markers. This would shift the analysis from a tissue- or patient-based level to a nucleus-by-nucleus basis, providing a more detailed and precise diagnostic insight. This would allow to develop a Virtual Multiplex tool.

6.2.3 WSI-based Inference

An additional step, particularly useful in itself and even more so if paired with successful registration, would be to perform inference directly on entire WSIs rather than on extracted tiles. This approach would facilitate virtual navigation through the tissue and provide a comprehensive overview.

6.2.4 Other Markers

Finally, it should be noted that the list of markers shown in 3.1 is much longer than those used for Transfer Learning. A natural next step would be to extend the analysis to additional markers and to conduct training with cytoplasmic markers as well.

Bibliography

[1] Elizabeth J. Taylor. *Dorland's Illustrated Medical Dictionary.* 29th. Philadelphia: Saunders, 2000, p. 1038. ISBN: 0-7216-6254-4 (cit. on p. 1).

[2] J. Ferlay et al. *Global Cancer Observatory: Cancer Today.* Lyon, France: International Agency for Research on Cancer. Accessed: [DD Month YYYY]. 2024. URL: `https://gco.iarc.who.int/today` (cit. on p. 1).

[3] American Cancer Society. *Cancer Facts & Figures 2024.* Atlanta, GA, 2024 (cit. on p. 2).

[4] Leukemia & Lymphoma Society. *Leukemia & Lymphoma Society.* Accessed: 4 November 2024. 2024. URL: `https://www.lls.org/` (cit. on p. 2).

[5] K. M. McKinnon. «Flow Cytometry: An Overview». In: *Current Protocols in Immunology* 120 (Feb. 2018), pp. 5.1.1–5.1.11. DOI: `10.1002/cpim.40` (cit. on pp. 2–4).

[6] H. Drescher, S. Weiskirchen, and R. Weiskirchen. «Flow Cytometry: A Blessing and a Curse». In: *Biomedicines* 9.11 (2021), p. 1613. DOI: `10.3390/biomedicines9111613`. URL: `https://doi.org/10.3390/biomedicines9111613` (cit. on pp. 2–4).

[7] Carla Valtriani and Carlo Hurle. *Citofluorimetria a flusso: aspetti generali con riferimento particolare allo studio di alcune malattie ematologiche.* Stampato a Genova. Viale Alfieri 36, 57100 Livorno, Italia: Caleidoscopio, Ospedali Riuniti Livorno - Azienda USL 6 - U.O. Immunologia-Allergologia Livorno, 1997 (cit. on p. 2).

[8] Junhun Cho. «Basic immunohistochemistry for lymphoma diagnosis». In: *Blood Research* 57.S1 (2022). Published online April 30, 2022. DOI: `10.5045/br.2022.2022037`. URL: `https://doi.org/10.5045/br.2022.2022037` (cit. on pp. 3, 4, 8).

[9] Jeffrey C. Pommerville. *Fundamentals of Microbiology.* Vol. I. Jones Bartlett Learning, 2017, pp. 248, 249. ISBN: 978-1-284-10095-2 (cit. on p. 4).

[10] A. H. Fischer, K. A. Jacobson, J. Rose, and R. Zeller. «Hematoxylin and Eosin Staining of Tissue and Cell Sections». In: *Cold Spring Harbor Protocols* 2008.6 (2008). DOI: `10.1101/pdb.prot4986` (cit. on p. 5).

[11] Juan José Barcia. «The Giemsa Stain: Its History and Applications». In: *International Journal of Surgical Pathology* 15.3 (2007), pp. 292–296. DOI: `10.1177/1066896907302239`. URL: `http://ijsp.sagepub.com` (cit. on p. 6).

[12] E. M. Brey, Z. Lalani, C. Johnston, M. Wong, L. V. McIntire, P. J. Duke, and C. W. Patrick Jr. «Automated selection of DAB-labeled tissue for immunohistochemical quantification». In: *Journal of Histochemistry Cytochemistry* 51.5 (May 2003), pp. 575–584. DOI: `10.1177/002215540305100503` (cit. on p. 7).

[13] Nicola Michielli. «Analysis of Signal Decomposition and Stain Separation Methods for Biomedical Applications». Supervisor: Full Professor Filippo Molinari. PhD thesis. Politecnico di Torino: Doctoral Program in Bioengineering and Medical-Surgical Sciences (33rd Cycle), 2024 (cit. on p. 7).

[14] The Royal College of Pathologists. *Digital pathology.* Accessed: 10 October 2024. 6 Alie Street, London E1 8QT, UK: The Royal College of Pathologists, 2024. URL: `https://www.rcpath.org/` (cit. on p. 7).

[15] AEQUIP s.r.l. *La sfida dell'Anatomia Patologica.* Accessed: 10 October 2024. Corso Castelfidardo 30/A, 10129 Torino, Italy, 2024. URL: `https://www.aequip.it/` (cit. on p. 7).

[16] Steven H. Swerdlow, Elias Campo, Nancy Lee Harris, Elaine S. Jaffe, Stefano A. Pileri, Harald Stein, and Jürgen Thiele, eds. *WHO Classification of Tumours of Haematopoietic and Lymphoid Tissues.* WHO Classification of Tumours, Revised 4th Edition, Volume 2. International Agency for Research on Cancer, 2017. ISBN: 9789283244943 (cit. on pp. 8, 12).

[17] W. W. Choi et al. «A new immunostain algorithm classifies diffuse large B-cell lymphoma into molecular subtypes with high accuracy». In: *Clinical Cancer Research* 15.17 (Sept. 2009). Epub 2009 Aug 25, pp. 5494–5502. DOI: `10.1158/1078-0432.CCR-09-0113` (cit. on p. 9).

[18] 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». In: *IEEE Transactions on Medical Imaging* 36.7 (July 2017). Epub 2017 Mar 6, pp. 1550–1560. DOI: `10.1109/TMI.2017.2677499` (cit. on pp. 17, 20, 49, 51).

[19] P. Naylor, M. Lae, F. Reyal, and T. Walter. «Segmentation of Nuclei in Histopathology Images by Deep Regression of the Distance Map». In: *IEEE Transactions on Medical Imaging* 38.2 (Feb. 2019), pp. 448–459. DOI: `10.1109/TMI.2018.2865709` (cit. on p. 18).

[20] Qingbo Kang, Qicheng Lao, and Thomas G. Fevens. «Nuclei Segmentation in Histopathological Images Using Two-Stage Learning». In: *Proceedings of the 22nd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2019), Part I*. Shenzhen, China: Springer, 2019, updating based on correct page range if available. DOI: `10.1007/978-3-030-32239-7_78` (cit. on p. 18).

[21] Lipeng Xie, Jin Qi, Lili Pan, and Samad Wali. «Integrating deep convolutional neural networks with marker-controlled watershed for overlapping nuclei segmentation in histopathology images». In: *Neurocomputing* 376 (2020), pp. 166–179. ISSN: 0925-2312. DOI: `10.1016/j.neucom.2019.09.083`. URL: `https://doi.org/10.1016/j.neucom.2019.09.083` (cit. on p. 18).

[22] Xiaoming Liu, Zhengsheng Guo, Jun Cao, and Jinshan Tang. «MDC-net: A new convolutional neural network for nucleus segmentation in histopathology images with distance maps and contour information». In: *Computers in Biology and Medicine* 135 (2021), p. 104543. ISSN: 0010-4825. DOI: `10.1016/j.compbiomed.2021.104543`. URL: `https://doi.org/10.1016/j.compbiomed.2021.104543` (cit. on p. 18).

[23] Amirreza Mahbod, Gerald Schaefer, Georg Dorffner, Sepideh Hatamikia, Rupert Ecker, and Isabella Ellinger. «A dual decoder U-Net-based model for nuclei instance segmentation in hematoxylin and eosin-stained histological images». In: *Frontiers in Medicine* 9 (Nov. 2022). Section: Precision Medicine, p. 978146. DOI: `10.3389/fmed.2022.978146`. URL: `https://doi.org/10.3389/fmed.2022.978146` (cit. on p. 18).

[24] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. «Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images». In: *Medical Image Analysis* 58 (2019), p. 101563. ISSN: 1361-8415. DOI: `10.1016/j.media.2019.101563`. URL: `https://doi.org/10.1016/j.media.2019.101563` (cit. on pp. 18, 19, 25).

[25] Hongliang He, Zhongyi Huang, Yao Ding, Guoli Song, Lin Wang, Qian Ren, Pengxu Wei, Zhiqiang Gao, and Jie Chen. «CDNet: Centripetal Direction Network for Nuclear Instance Segmentation». In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 4006–4015. DOI: `10.1109/ICCV48922.2021.00399` (cit. on pp. 18, 19, 39–41).

[26] Hussein Naji, Lucas Sancere, Adrian Simon, Reinhard Büttner, Marie-Lisa Eich, Philipp Lohneis, and Katarzyna Bożek. «HoLy-Net: Segmentation of histological images of diffuse large B-cell lymphoma». In: *Computers in Biology and Medicine* 170 (2024), p. 107978. ISSN: 0010-4825. DOI: `10.`

1016/j.compbiomed.2024.107978. URL: https://doi.org/10.1016/j.compbiomed.2024.107978 (cit. on pp. 18, 25).

[27] N. Kumar et al. «A Multi-Organ Nucleus Segmentation Challenge». In: *IEEE Transactions on Medical Imaging* 39.5 (May 2020). Epub 2019 Oct 23, pp. 1380–1391. DOI: 10.1109/TMI.2019.2947628 (cit. on p. 20).

[28] PathologyOutlines.com, Inc. *Stains (IHC & special) & CD markers*. Last staff update: 11 October 2024. Editorial Board oversight by Christian M. Schürch, M.D., Ph.D. and Brandon Umphress, M.D. (last reviewed January 2023). 30150 Telegraph Road, Suite 119, Bingham Farms, Michigan 48025, USA, 2024. URL: https://www.pathologyoutlines.com/ (cit. on p. 22).

[29] Sara Siragusa. «Proliferation Index Estimation in Immunohistochemical Images via Deep Learning: Stain Normalization Impact Assessment in Nuclei Segmentation and Classification». Master's Degree in Biomedical Engineering. Supervisors: Prof. Massimo Salvi, Ing. Nicola Michielli. MA thesis. Turin, Italy: Politecnico di Torino, 2024 (cit. on pp. 24, 60).

[30] Massimo Salvi, Nicola Michielli, Kristen M. Meiburger, Cristina Cattelino, Bruna Cotrufo, Matteo Giacosa, Chiara Giovanzana, and Filippo Molinari. «cyto-Knet: An instance segmentation approach for multiple myeloma plasma cells using conditional kernels». In: *International Journal of Imaging Systems and Technology* 34.1 (2023). First published: 25 October 2023, e22984. DOI: 10.1002/ima.22984. URL: https://doi.org/10.1002/ima.22984 (cit. on p. 27).

[31] 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». In: *Scientific Reports* 13 (2023), p. 8614. DOI: 10.1038/s41598-023-35605-7. URL: https://doi.org/10.1038/s41598-023-35605-7 (cit. on pp. 51, 52).