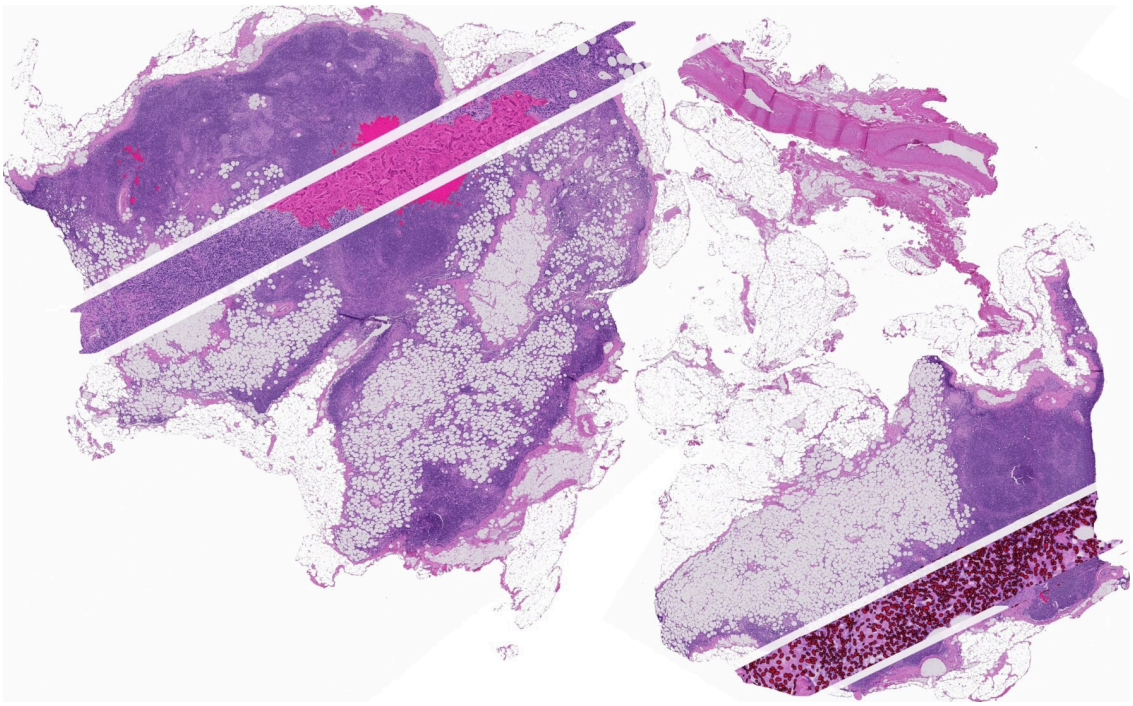
Dissertation
Fakultät für Physik

FABIAN HÖRST

BRIDGING SCALES IN DIGITAL PATHOLOGY:

Modern Computer Vision Algorithms for Automated Tissue and Cell Segmentation



technische universität dortmund

Universitätsmedizin Essen
Institut für KI in der Medizin (IKIM)

technische universität dortmund

Universitätsmedizin Essen
Institut für KI in der Medizin (IKIM)

Arbeit zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

Bridging Scales in Digital Pathology:

Modern Computer Vision Algorithms for Automated Tissue and Cell Segmentation

Fabian Hörst

Arbeitsgruppe Medizinische und Biologische Physik
Fakultät Physik
Technische Universität Dortmund

und

Medical Machine Learning
Institut für künstliche Intelligenz in der Medizin
Universitätsklinikum Essen

März 2025

Dissertationsschrift
Vorgelegt an der Fakultät für Physik
der Technischen Universität Dortmund
Dortmund

Arbeitsgruppe Medizinische und Biologische Physik
Fakultät Physik
Technische Universität Dortmund
Dortmund

Medical Machine Learning
Institut für künstliche Intelligenz in der Medizin
Universitätsklinikum Essen
Essen

Erstgutachter:	Prof. Dr. Dr. Jens Kleesiek
Zweitgutachter:	Prof. Dr. Matthias Schneider
Drittgutachter:	Prof. Dr. Daniel Rückert
Abgabedatum:	31.03.2025
Datum des Rigorosums:	05.11.2025
Ort:	Bochum/Essen/Dortmund

Bridging Scales in Digital Pathology:
Modern Computer Vision Algorithms for Automated Tissue and Cell Segmentation

Cover: The figure shows a lymph node sample from the CAMELYON16 dataset (Tumor-084). Two zoomed-in regions are highlighted. In the first, metastatic areas are annotated with a pink overlay. In the second, cell nuclei are segmented (red overlay), with the segmentation masks generated by our CellViT algorithm, a key component of this work.

Typeset using LaTeX.

Bochum/Essen, 2025

Abstract

Histopathology is a cornerstone of disease diagnosis and treatment, traditionally relying on manually assessing tissue specimens under a microscope. However, the advent of slide scanners to produce digital tissue representations, so-called whole-slide images (WSI), has enabled computational pathology to perform quantitative and automated tissue analysis. Current developments in Artificial Intelligence, particularly Deep Learning, have accelerated the progress in this field.

This thesis proposes a comprehensive Deep Learning pipeline for quantitative histopathological image analysis, integrating WSI preprocessing, algorithm development for tissue and cell-level segmentation, and clinical application in an end-to-end workflow. The approach not only improves the quantitative evaluation of WSI but also extracts diagnostic and prognostic markers while automatically characterizing tissue dynamics through morphological tissue features.

Segmenting entire tissue sections into classes like tumorous or non-tumorous requires the consideration of global tissue patterns as well as local cell morphologies. Following this, we introduce the Memory Attention Framework that can be incorporated into any encoder-decoder segmentation architecture. This framework enables the adaptive incorporation of tissue context during fine-grained local segmentation. The method was evaluated on two public datasets (breast, liver) and an internal kidney cancer dataset, demonstrating superiority over non-context and multiscale segmentation approaches. Notably, the approach reduced the number of false-positive tumor regions. Building on this, we applied the framework to a pancreatic cancer cohort consisting of 400 internal and 182 external patients to quantify the tumor microenvironment and correlate it with patient outcomes. In doing so, we were able to stratify patients into two risk groups based on tissue composition and spatial tumor-stroma distribution, which showed significant ($p < 0.05$) differences in their survival probabilities.

Next to tissue analysis, segmentation on the cellular level is crucial to uncover the cellular composition of tissue samples. While convolutional neural networks have been extensively used for this task, we evaluate the capabilities of Transformer-based networks and incorporate so-called foundation models to improve accuracy compared to existing solutions. The proposed CellViT and CellViT^{++} models have proven to achieve State-of-the-Art results on several benchmark datasets, covering a broad spectrum of tissue types and cell classes, bringing cell segmentation solutions closer to clinical practice. The models require minimal data for fine-tuning and exhibit remarkable zero-shot cell segmentation quality. This capability allows for a considerably faster adaptation to new research hypotheses without the need for extensive development time.

In summary, this work presents Deep Learning techniques for quantifying tissue at both the macro and micro levels, enhancing diagnostic workflows, and identifying prognostic markers.

Kurzzusammenfassung

Die Histopathologie ist ein wichtiger Bestandteil der Diagnose und Behandlung von Krankheiten und basiert traditionell auf der manuellen Beurteilung von Gewebeproben. Mit der Entwicklung von Whole-Slide-Images (WSI) und Deep Learning-Techniken können automatisierte und quantitative Gewebeanalysen durchgeführt werden.

In dieser Arbeit wird eine umfassende Deep Learning-Pipeline für die quantitative histopathologische Bildanalyse vorgeschlagen, die von der Datenverarbeitung der WSI über die Entwicklung von Algorithmen für die Segmentierung auf Gewebe- und Zellebene bis hin zur klinischen Anwendung in einem End-to-End-Workflow reicht. Der Ansatz verbessert nicht nur die quantitative Auswertung von WSI, sondern extrahiert auch diagnostische und prognostische Marker und charakterisiert automatisch die Gewebedynamik durch morphologische Gewebemerkmale.

Bei der Segmentierung von Gewebeschnitten in Entitäten wie tumorös oder nicht-tumorös müssen sowohl globale Gewebemuster als auch lokale Zellmorphologien berücksichtigt werden. In dieser Arbeit stellen wir das Memory Attention Framework vor, welches die adaptive Einbeziehung des Gewebekontexts während der feingranularen lokalen Segmentierung ermöglicht. Die Methode wurde auf zwei öffentlichen Datensätzen (Brust, Leber) und einem internen Nierenkarzinomdatensatz evaluiert, wobei eine Verbesserung gegenüber existierenden Segmentierungsansätzen nachgewiesen werden konnte. Insbesondere konnte die Zahl der falsch-positiven Tumorregionen reduziert werden. Darauf aufbauend haben wir die Methode auf einer Kohorte von 400 internen und 182 externen Patienten mit Pankreaskarzinomen angewandt, um die Mikroumgebung des Tumors zu quantifizieren und mit der Überlenszeit der Patienten zu korrelieren. Auf diese Weise konnten wir die Patienten anhand der Gewebezusammensetzung und der räumlichen Tumor-Stroma-Verteilung in zwei Risikogruppen einteilen, die signifikante Unterschiede ($p < 0.05$) in ihrer Überlebenswahrscheinlichkeit aufwiesen.

Neben der Gewebesegmentierung ist auch die Segmentierung auf zellulärer Ebene notwendig, um die Gewebezusammensetzung auch auf mikroskopischer Ebene zu analysieren. Die vorgeschlagenen Modelle CellViT und CellViT^{++} konnten zeigen, dass sie auf mehreren Benchmark-Datensätzen, die ein breites Spektrum von Gewebetypen und Zellklassen abdecken, herrausragende Ergebnisse erzielen. Sie erfordern nur minimale Datenmengen für die Anpassung an neue Zellklassen. Diese Fähigkeit ermöglich eine wesentlich schnelle Anpassung des Netzwerkes an neue Taxonomien.

Zusammenfassend zeigt diese Arbeit das Potenzial von Deep-Learning-Techniken zur Quantifizierung von Gewebe sowohl auf Makro- als auch auf Mikroebene, um diagnostische Arbeitsabläufe zu verbessern und prognostische Marker zu identifizieren.

Acknowledgements

This journey over the past three years has been filled with inspiring collaborations, intensive discussions, and moments of joy, but also with considerable challenges and hard work. I am deeply grateful for the incredible people I have met along the way, especially within IKIM. This chapter of my life would not have been possible without the colleagues who accompanied me through both scientific and personal challenges. I have grown not only as a researcher but also gained valuable personal experiences.

I would like to express my heartfelt gratitude to Prof. Dr. Dr. Jens Kleesiek for his unwavering support, invaluable advice, and continuous encouragement. His mentorship provided me with the necessary guidance and freedom to develop my work, and his dedication to the institute created the environment and resources that made this research possible. Thank you for your trust and the time we shared throughout this journey.

To all the members of our research group, thank you for your constructive feedback and for fostering such a pleasant working environment. I still owe a rematch or two at the soccer table with some of you. A special thanks goes to the DevOps team (Enrico, Fin, Lukas, Sameh, and Kajepaan) for always providing technical support and maintaining an outstanding infrastructure. My gratitude also extends to all my collaborators who enriched my work through their feedback, medical expertise, project discussions, and contributions of datasets. Saskia Ting and Jens Siveke, your input significantly shaped my first publication and guided me toward the field of digital pathology, thank you for that. Dear Barbara, thank you as well for the engaging discussions and our collaboration within your pancreatic cancer project idea. To all my co-authors on the papers published during my PhD, I greatly appreciate your feedback and insights. A special 'thank you' goes to Prof. Dr. Dr. Jan Egger. Although we collaborated only occasionally on scientific projects, you always generously reviewed my manuscripts and provided valuable feedback. I am also grateful to my co-supervisor, Prof. Dr. Matthias Schneider, for taking on the role of the co-advisor during my doctoral studies. I would like to give heartfelt thanks to Lukas, Moritz, and Helmut. I truly enjoyed the time we spent together outside of work, which provided balance and support throughout this journey. It was always a pleasure to share the office with Moritz and Helmut, whose company made even the most intense workdays enjoyable.

To my family and friends, thank you for your understanding and support over the past three years. I am especially grateful for your patience when I was once again buried in work over the weekends or showed up exhausted from long hours at my desk. I am profoundly thankful to my parents, Sabine and Martin, who have always encouraged me to pursue my academic path and provided me with every possible resource along the way. While I wish it were otherwise, the influence of parents on education in Germany remains significant, and I deeply appreciate all

the opportunities you made possible for me. Finally, my deepest gratitude goes to my beloved wife, Kim, who often had to make sacrifices but never failed to show understanding. At the beginning of my PhD, you warned me not to spend too much time at my desk again, and I fear I broke that promise far too often. Yet, your constant support and encouragement were invaluable, and I could not have done this without you. I am proud of you!

A final thanks goes to the Cancer Research Center Cologne Essen (CCCE) for funding my research, to University Hospital Essen as my employer, to TU Dortmund University, particularly the Dortmund Graduate School of Physics, where I was enrolled as a doctoral student, and of, course, to all the people at the IKIM and the MML group.

List of Publications

This doctoral research resulted in the following thesis-related publications (sorted by publication date):

Publication 1. Oliver Ester, **Fabian Hörst**, Constantin Seibold, Julius Keyl, Saskia Ting, Nikolaos Vasileiadis, Jessica Schmitz, Philipp Ivanyi, Viktor Grünwald, Jan Hinrich Bräsen, Jan Egger, and Jens Kleesiek. *Valuing vicinity: Memory attention framework for context-based semantic segmentation in histopathology.* In: *Computerized Medical Imaging and Graphics* 107 (July 2023), p. 102238. ISSN: 0895-6111. DOI: 10.1016/j.compmedimag.2023.102238.

Publication 2. **Fabian Hörst**, Sajad H. Schaheer, Giulia Baldini, Fin H. Bahnsen, Jan Egger, and Jens Kleesiek. *Accelerating Artificial Intelligence-based Whole Slide Image Analysis with an Optimized Preprocessing Pipeline.* In: *Bildverarbeitung für die Medizin 2024*. Springer Fachmedien Wiesbaden (February 2024), pp. 356-361. ISBN: 978-3-658-44037-4. DOI: 10.1007/978-3-658-44037-4_91.

Publication 3. **Fabian Hörst**, Moritz Rempe, Lukas Heine, Constantin Seibold, Julius Keyl, Giulia Baldini, Selma Ugurel, Jens Siveke, Barbara Grünwald, Jan Egger, and Jens Kleesiek. *CellViT: Vision Transformers for precise cell segmentation and classification.* In: *Medical Image Analysis* 94 (May 2024), p. 103143. ISSN: 1361-8415. DOI: 10.1016/j.media.2024.103143.

Publication 4. **Fabian Hörst**, Moritz Rempe, Helmut Becker, Lukas Heine, Julius Keyl, and Jens Kleesiek. *CellViT++: Energy-Efficient and Adaptive Cell Segmentation and Classification Using Foundation Models* (January 2025) DOI: 10.48550/arXiv.2501.05269. (Preprint).

The following publications have been authored or co-authored by Fabian Hörst during his doctoral research, but have not been incorporated into this thesis:

1. **Fabian Hörst**, Saskia Ting, Sven-Thorsten Liffers, Kelsey L Pomykala, Katja Steiger, Markus Albertsmeier, Martin K Angele, Sylvie Lorenzen, Michael Quante, Wilko Weichert, Jan Egger, Jens T Siveke, and Jens Kleesiek. *Histology-Based Prediction of Therapy Response to Neoadjuvant Chemotherapy for Esophageal and Esophagogastric Junction Adenocarcinomas Using Deep Learning.* In: *JCO Clinical Cancer Informatics* 7 (August 2023). ISSN: 2473-4276. DOI: 10.1200/cci.23.00038.
2. Lukas Heine, **Fabian Hörst**, Enrico Nasca, Jan Egger, Jens T. Siveke, Moon Kim, Jens Kleesiek, and Fin H. Bahnsen. *Lean Study Host: Towards an Automated Pipeline for Multi-Center Study Hosting.* In: *Proceedings of the 57th Hawaii International Conference on System Sciences* (January 2024). ISBN: 978-0-9981331-7-1. DOI: 10125/107396
3. Moritz Rempe, **Fabian Hörst**, Constantin Seibold, Boris Hadaschik, Marco Schlimbach, Jan Egger, Kevin Kröninger, Felix Breuer, Martin Blaimer, and Jens Kleesiek. *Tumor likelihood estimation on MRI prostate data by utilizing k-Space information*. (June 2024). DOI: 10.48550/arXiv.2407.06165. (Preprint - Accepted as oral presentation on the 2025 annual meeting of the International Society for Magnetic Resonance in Medicine).
4. Johannes Raufeisen, Kunpeng Xie, **Fabian Hörst**, Till Braunschweig, Jianning Li, Jens Kleesiek, Rainer Röhrig, Jan Egger, Bastian Leibe, Frank Hölzle, Alexander Hermans, and Behrus Puladi. *Cyto R-CNN and CytoNuke Dataset: Towards reliable whole-cell segmentation in bright-field histological images*. In: *Computer Methods and Programs in Biomedicine* 252 (July 2024), p. 108215. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2024.108215.
5. Hamza Kalisch, **Fabian Hörst**, Ken Herrmann, Jens Kleesiek, and Constantin Seibold. *Autopet III challenge: Incorporating anatomical knowledge into nnUNet for lesion segmentation in PET/CT*. (September 2024). DOI: 10.48550/arXiv.2409.12155. (Preprint - Oral presentation by Fabian Hörst during the AutoPET III workshop at MICCAI 2024).
6. Lukas Heine, **Fabian Hörst**, Jana Fragemann, Gijs Luijten, Miriam Balzer, Jan Egger, Fin H. Bahnsen, M. Saquib Sarfraz, Jens Kleesiek, and Constantin Seibold. *Spacewalker: Traversing Representation Spaces for Fast Interactive Exploration and Annotation of Unstructured Data*. (September 2024). DOI: 10.48550/arXiv.2409.16793. (Preprint).

7. Georg C. Lodde, Fang Zhao, Rudolf Herbst, Patrick Terheyden, Jochen Utikal, Claudia Pföhler, Jens Ulrich, Alexander Kreuter, Peter Mohr, Ralf Gutzmer, Friedegund Meier, Edgar Dippel, Michael Weichenthal, Philipp Jansen, Bernd Kowall, Wolfgang Galetzka, **Fabian Hörst**, Jens Kleesiek, Birte Hellwig, Jörg Rahnenführer, Luisa Rajcsanyi, Triinu Peters, Anke Hinney, Jan-Malte Placke, Antje Sucker, Annette Paschen, Jürgen C. Becker, Elisabeth Livingstone, Lisa Zimmer, Alpaslan Tasdogan, Alexander Roesch, Eva Hadaschik, Dirk Schadendorf, Klaus Griewank, and Selma Ugurel. *Early versus late response to PD-1-based immunotherapy in metastatic melanoma*. In: *European Journal of Cancer* 210 (October 2024), p. 114295. ISSN: 0959-8049. DOI: 10.1016/j.ejca.2024.114295.
8. Moritz Rempe, Lukas Heine, Constantin Seibold, **Fabian Hörst**, and Jens Kleesiek. *De-Identification of Medical Imaging Data: A Comprehensive Tool for Ensuring Patient Privacy*. (October 2024). DOI: 10.48550/arXiv.2410.12402. (Preprint).
9. Jianning Li, Zongwei Zhou, Jiancheng Yang, Antonio Pepe, Christina Gsaxner et al. *MedShapeNet - A Large-Scale Dataset of 3D Medical Shapes for Computer Vision.* In: *Biomedical Engineering / Biomedizinische Technik* (December 2024). 70(1), 71-90 . DOI: 10.1515/bmt-2024-0396.
10. Negar Shahamiri, Moritz Rempe, Lukas Heine, Jens Kleesiek, **Fabian Hörst**. *Cracking the PUMA Challenge in 24 Hours with CellViT++ and nnU-Net*. (March 2025). DOI: 10.48550/arXiv.2503.12269. (Preprint).

Personal Contribution Statement

Publication 1. Oliver Ester is the first author of this work, responsible for the conceptualization, methodology, formal analysis, and drafting of the initial manuscript. He came up with the initial idea for the memory attention framework and the concept for the project. He was also primarily responsible for the implementation of the project, but in collaboration with Fabian Hörst (30% total contribution). Fabian Hörst was mainly included in the execution and implementation of the project after the project handover in October 2022. He provided significant methodological input for the initial submission (MICCAI 2022), managed all subsequent re-submissions (Medical Image Analysis, IEEE Transactions on Medical Imaging, Computerized Medical Imaging and Graphics as corresponding author) and revisions of the manuscript, and took over responsibility for the project from Oliver Ester. The mathematical formulation of the publication was mainly developed by Fabian Hörst. Both authors jointly designed the experiments on the RCC and CY16 dataset (30% contribution), and Fabian Hörst solely conducted new experiments on the Paip 2019 dataset and the Selocan cohort. Fabian Hörst made code improvements and adapted the manuscript based on reviewer comments. The entire CRediT authorship contribution statement can be found in the peer-reviewed journal submission.

Publication 2. Fabian Hörst was responsible for the conceptualization, data curation, formal analysis, methodology, software development, writing, reviewing, and visualization. Giulia Baldini and Sajad H. Schaheer provided initial help for the software development. Fin H. Bahnsen, Jan Egger and Jens Kleesiek supervised the publication.

Publication 3. Fabian Hörst was responsible for the conceptualization, data curation, formal analysis, and investigation of the project. He contributed significantly to the methodology, software development, and visualization. Additionally, he authored the original draft of the manuscript and managed both the review and editing processes. The entire CRediT authorship contribution statement can be found in the peer-reviewed journal submission.

Publication 4. Fabian Hörst was responsible for the conceptualization, data curation, formal analysis, and investigation of the project. He contributed significantly to the methodology, software development, and visualization. Additionally, he authored the original draft of the manuscript and managed both the review and editing processes. The entire CRediT authorship contribution statement can be found in the preprint publication.

Dedicated to my beloved wife, Kim,
and my dear parents, Sabine and Martin

"Science cannot solve the ultimate mystery of nature. And that is because, in the last analysis, we ourselves are a part of the mystery that we are trying to solve."

— Max Planck

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
AJI	Averaged Jacard Index
CLI	Command Line Interface
CNN	Convolutional Neural Network
CPath	Computational Pathology
CPU	Central Processing Unit
CT	Computed Tomography
CV	Computer Vision
DAPI	4′,6-Diamidino-2-Phenylindole
DL	Deep Learning
DQ	Detection Quality
FDA	Food and Drug Administration
FHIR	Fast Healthcare Interoperability Resources
FN	False Negative
FOV	Field of View
FP	False Positive
GPU	Graphics Processing Unit
GT	Ground Truth
HE	Hematoxylin and Eosin
IF	Immunofluorescence

IHC Immunohistochemistry

IoU Intersection-over-Union

MACs Multiply-Accumulate Operations

MAF Memory Attention Framework

MHA Multi-Head Attention

ML Machine Learning

MLP Multi-Layer Perceptron

MRI Magnetic Resonance Imaging

NMS Non Maximum Suppression

PAAD Pancreatic Adenocarcinoma

PQ Panoptic Quality

QC Quality Control

RAM Random-Access Memory

RCC Renal Cell Carcinoma

ROI Region of Interest

SAM Segment Anything Model

SOTA State-of-the-Art

SQ Segmentation Quality

SSL Self-Supervised Learning

TCGA The Cancer Genome Atlas

TILs Tumor Infiltrating Lymphocytes

TP True Positive

UMAP Uniform Manifold Approximation and Projection

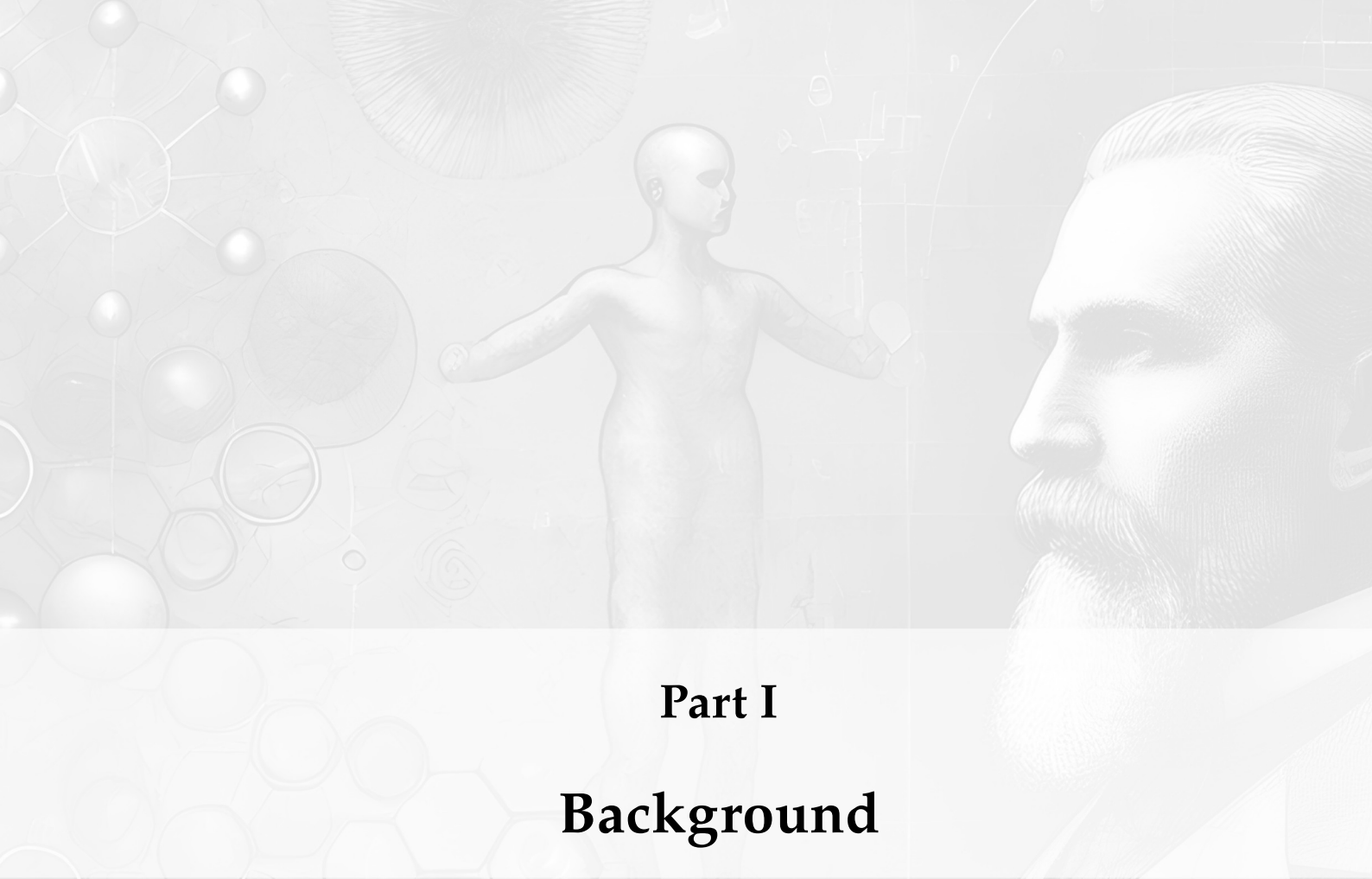
UK United Kingdom

US Ultrasound

ViT Vision Transformer

WSI Whole-slide image

xAI Explainable AI



Part I

Background

1 Introduction

In this thesis, we propose methods for the quantification of histopathological images in order to support pathologists in their diagnostic workflows and predictive analyses. To achieve this, both macro (tissue) and micro-level (cell) segmentations are necessary, requiring the adaptation of computer vision algorithms into digital pathology. In particular, we propose a new preprocessing pipeline (1) and review tissue (2) and cell (3) segmentation techniques. As many of the recent segmentation methods require extensively annotated datasets, we concentrate on data-efficient fine-tuning for our cell segmentation models (4). We provide algorithms that quantitatively assess histopathological images.

1.1 Modern Histopathology: From Microscopy to AI

1.1.1 Pathology as a Cornerstone of Medical Diagnostics

Medicine, as defined by the Cambridge dictionary, "is the science dealing with the preserving of health and with preventing and treating disease or injury" [27]. One of the cornerstones of modern medicine is pathology, particularly histopathology. The term pathology originates from the Greek words pathos (suffering) and logos (study) [195, p. 1]. Pathology investigates structural and functional abnormalities in organs, tissues, and cells that arise in response to disease. Historically, the field was primarily an autopsy-based retrospective discipline [195, p. 5]. This changed

in the late 19th century by Rudolf Virchow, who established the so-called "modern pathology" with the introduction of microscopic tissue analysis, now referred to as histopathology (histos = tissue) [94, 195]. Histopathology is widely considered the gold standard for diagnosing the presence, type, and progression of several diseases. Through the systematic examination of tissues and cellular structures, pathologists provide insights that guide clinical decisions and treatment strategies for a range of conditions, including cancer, infections, autoimmune diseases, genetic disorders, neurodegenerative diseases, cardiovascular diseases, and even transplant rejection [116]. Diagnostic findings are derived primarily from the visual observation and interpretation of patterns of the specimens [152]. Pathologists address questions such as:

Does this biopsy have any signs of tumor cells?
What is the histopathological classification of the tumor identified?
Are there any pathological changes in this tissue sample indicative of autoimmune diseases?
Considering the patient's increased infection markers, does this tissue sample show any histological evidence of an infection?

Looking at some statistics, the importance of pathology becomes more pronounced: In 2006, the National Health Service in England reported that 70% of all medical diagnoses involved consultations with pathologists [35, 203]. Approximately 20 million slides are examined annually in the United Kingdom (UK), with demand increasing at an estimated rate of 4.5% per year between 2007 and 2023 [25]. Similarly, in Germany, approximately 1,000 slides per day are processed in an average pathology department [216, p. 30].

In contrast to the increasing number of diagnostic inquiries and the demand for more comprehensive diagnoses, first analyses indicate an emerging staff shortage within pathology departments [152]. In Germany, pathologists are among the oldest medical specialists [216, p. 25], and in the UK, about 19% of histopathologists will retire in the next 5 years [261], contradictory to the predicted increase of 45% needed by 2029 to handle the growing demand [86, p. 11]. Without the implementation of countermeasures, including the integration of digitization, the quality of care will decline in the next few years [203].

1.1.2 Impact of Digitization on Medical Imaging

Currently, the routine pathology workflow is still based on manual tissue processing steps with a microscopy-based diagnosis. The introduction of tissue scanners in the last decade has enabled the creation of a digital representation of the tissue sample, referred to as whole-slide image (WSI) [297]. In 2017, the first FDA-approved scanner was introduced, with FDA clearance based on a study demonstrating that

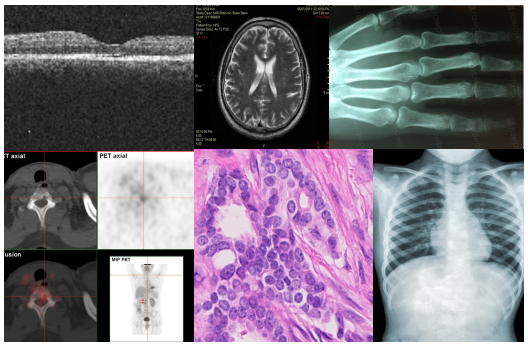


Figure 1.1: Examples of some medical imaging domains in which DL has achieved human-like performance. Adapted from [171]

diagnoses derived from digital images were consistent with those from microscopy-based evaluation [77, 201, 213]. Together with advances in computer vision (CV) and information technology, as well as hardware developments, including affordable storage solutions and processing units, the field of histopathology is now undergoing a digital transformation, which fosters computational pathology (CPath) [206, 248]. In parallel, milestones in artificial intelligence (AI), particularly Deep Learning (DL), significantly improved the algorithmic evaluation of medical imaging data [206] (examples in Figure 1.1).

Radiology exemplifies the possible influence of digital transformation in medical imaging. Due to the digital nature of radiology, with digital imaging techniques such as ultrasound (US), magnetic resonance imaging (MRI), computed tomography (CT), and X-rays, a large number of algorithms have been developed. Several tools are already available for research and routine clinical use [15, 108, 113, 128, 276]. Such applications include segmentation of anatomical structures and body composition in CT images [108, 276], or of brain tumors in MRI images [106, 180]. According to an analysis in 2023, a total of 190 AI-based tools had been approved for chest radiology by 2022 [192]. Caused by several reasons, including the late introduction of WSI scanners (2017 [77] versus the first CT scanner in 1973 [63]) and the inherent complexity of the data, the adoption of DL algorithms in pathology is lacking behind radiology. Pathological images are significantly larger than radiological images, miss anatomical orientation, encompass multiscale information at both the tissue and cell level, and are typically stained with various techniques, including hematoxylin and eosin (H&E) or immunohistochemical (IHC) stainings [70, 206]. The H&E stain is considered the workhorse of pathology. To highlight the difference in size and information density, Figure 1.2a presents a size comparison between a WSI and a 3D chest CT image [70].

1.1.3 Opportunities of Digital Pathology

Despite these challenges, the potential of DL in pathology is significant and has already been demonstrated in various works [70, 248]. The field of application includes both basic image analysis tasks, including tumor detection and tumor subtyping [29, 176], as well as more advanced applications such as survival time analysis [82, 139, 288], treatment response prediction [118, 161], biomarker exploration [72, 73, 104], and detection of genetic alterations [138]. Many of these AI applications are based on classification networks that predict clinical endpoints from

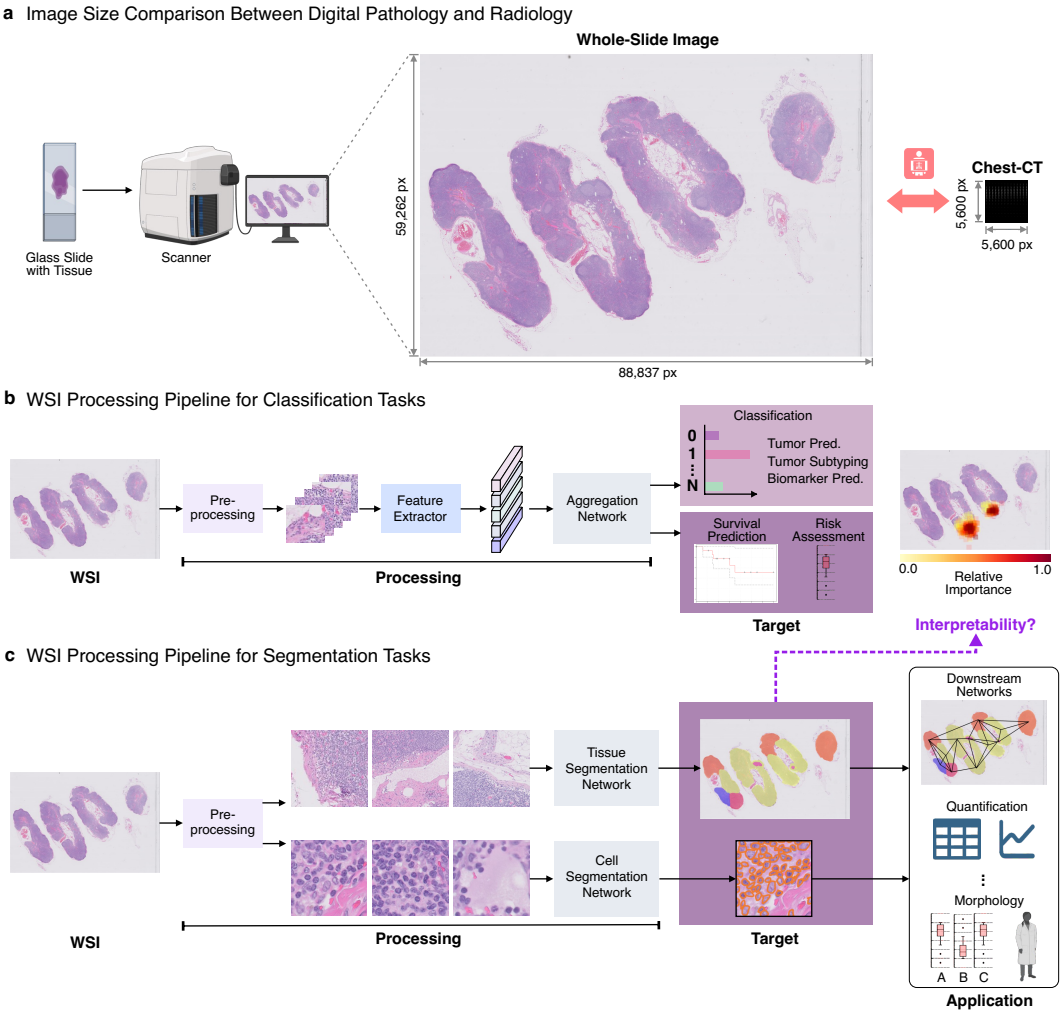


Figure 1.2: Overview of histopathological data processing.
a) Size comparison between a digitized tissue specimen vs. a 3D chest CT (illustrated as flattened 2D image) [70]. **b**) Processing pipeline for classification tasks with optional heatmaps. **c**) Processing pipeline for segmentation tasks with exemplary downstream applications. Created with [57].

tissue morphology in an end-to-end setting. For example, Campanella et al. [29] demonstrated that their DL-based algorithms for cancer classification in prostate cancer, basal cell carcinoma, and breast cancer achieved clinical-grade performance. Their algorithms enabled pathologists to exclude $65 - 75\%$ of slides without cancer while maintaining 100% sensitivity [29]. This illustrates the potential of CPath to improve workflow efficiency, an essential factor in the context of the contrasting trends of decreasing workforce and an increasing workload [83]. An example of an end-to-end pipeline for WSI classification is shown in Figure 1.2b.

However, DL-based classification algorithms have a significant limitation: their lack of explainability (xAI) and the inherent black-box nature [79, 152, 248]. This poses two key challenges: first, we are unable to fully understand "what the algorithm learns", and second, it complicates the discovery of new biomarkers because the results cannot be mapped back to biological processes. Although external validation may be sufficient for clinical applicability in tasks such as tumor detection or subtyping, the trust issue remains from the clinician's perspective. During my research at the University Hospital Essen, I frequently encountered critical questions from clinicians about the explainability of these models. Either they wanted to understand how the algorithm derived its decision, or there was a lack of visual feedback. My subjective observations are also supported by several studies that highlight the need for xAI in the context of histopathology [19, 26, 78, 79, 133, 221]. Although attention-based networks [123, 176] exhibit a kind of interpretability by highlighting regions of interest (Figure 1.2b), they only provide coarse heatmaps at a macroscopic level, and in-depth analysis is still needed [133].

Next to classification algorithms, segmentation algorithms are important in CPath. During the segmentation process, the WSI is divided into different components. At the macroscopic tissue level, this includes the detection of tissue compartments such as tumor or stromal areas [248]. The microscopic tissue level includes nuclei or glands, as illustrated in Figure 1.2c [248]. Segmentation is helpful for clinical decision-making by objectively and quantitatively measuring tissue composition and cellular entities, accessing morphological information, and correlating them with clinical endpoints such as diagnosis or response information [248]. Exemplary, Sun et al. [252] have shown that segmentation networks can be used to standardize the assessment of tumor-infiltrating lymphocytes (TILs) [4, 236], a prognostic and predictive biomarker in breast cancer. Geessink et al. [85] automatically evaluated the tumor-stroma ratio as an independent prognostic factor in colorectal cancer using DL-based segmentation networks. These tasks would not be feasible for pathologists on a large scale due to their time-consuming nature and high degree of intra- and inter-observer variability [119]. In addition to providing prognostic information about cell and tissue distributions within samples, segmentation algorithms can also be used for downstream algorithms such as cell graphs [37, 178, 218] or tissue graphs [160, 226]. Compared to classification algo-

rithms, segmentation networks offer the advantage of being directly connected to pathological knowledge and morphologies. When combined with attention-based classification methods, segmentation algorithms enable the objective quantification of established biomarkers and the discovery of unknown biomarkers by quantifying high-impact regions [152]. This represents an important step towards xAI (Figures 1.2b and 1.2c).

Although many algorithms have been proposed for WSI segmentation, we observe a lack of domain-specific solutions. Either CV algorithms are adapted for a specific use case (isolated solutions), which means that current developments in the CV field are integrated with a delay, or the domain-specific properties of WSI (high resolutions, information density) are not taken into account. In addition, these solutions are mainly based on manually annotated datasets, for which pathologists have to create ground truth tissue segmentations. Creating annotations is time-consuming and costly, a factor that should not be underestimated given the current staff shortage (Section 1.1.1) [248].

1.1.4 Problem Statement

When reviewing the previous subsections (1.1.1-1.1.3), the current challenges in (digital) pathology can be summarized into the following key issues:

Staff shortages necessitate digitization: The shortage of pathologists makes digital transformation vital. This has multiple downstream effects, including the need to re-design workflows and enable automation using AI.

Limitations of classification networks: Although classification algorithms have the potential to significantly reduce workload by automating workflows, critical gaps remain. Clinical validation is still lacking, and the absence of thorough research on algorithm explainability limits its clinical translation (spurious correlations, biomarker discovery).

Segmentation algorithms are underdeveloped: Segmentation algorithms offer opportunities to compute quantifiable parameters and shed light on the tissue composition. Still, their development lags behind advancements in CV and holistic frameworks are missing, such as the nnUNet [128] for radiology.

Reliance on expert-level annotations: The training of supervised algorithms depends on large annotated datasets. These datasets are difficult to obtain and requires pathologists.

In our work, we focus on the challenge of segmenting tissue structures and cells in WSI to automate pathology tasks, e.g., cell counting, improve algorithm interpretability, and support pathologists' hypothesis testing. The aim is to develop holistic quantitative analysis methods while addressing the current research gap between CV and CPath. This involves creating a preprocessing pipeline and designing algorithms that utilize the multiscale structure of WSIs.

To put it simply, rather than answering the typical qualitative questions about what can be seen in a tissue sample as given in Section 1.1.1, our aim is to answer more precise, quantitative questions like:

How many tumor cells are present in this sample?
What is the ratio of lymphocytes to tumor cells in this sample?
How does the distribution of the tumor microenvironment compare with the total tumor amount?

1.2 Thesis Roadmap and Contributions

This dissertation focuses on semantic and instance segmentation algorithms for digital pathology. Based on the current research gap discussed in Chapter 2, we initially introduce a preprocessing framework to deal with digital pathology data (Chapter 3). Starting from a high-level tissue perspective, we propose a tissue segmentation framework in Chapter 4, that efficiently takes the multiscale information of WSI into account to improve segmentation quality and consider tissue context within a specimen. Moving forward to higher resolutions, we introduce a new DL network architecture for cell instance segmentation (Chapter 5). Considering the lack of extensively annotated ground truth segmentation datasets, we extend our cell segmentation network to build an entire framework which can deal with limited annotated data (few-shot), is easy to use, and requires minimal computational resources. Finally, we set our solutions into a broader context in Chapter 6 and discuss potential future directions in Chapter 7.

A Comprehensive Framework for Whole-Slide Image Preprocessing

Digital image data in pathology is stored in pyramid images, consisting of tiled images with increasing resolution on different levels. Processing these images is nontrivial since the file format is not standardized yet and different DL algorithm requirements have to be considered. In this Chapter (Chapter 3), we present a preprocessing pipeline for CPath. We focus on (1) interoperability and (2) processing speed. Our goal is to provide a preprocessing library compatible with various proprietary and open-source file formats. The functionality should be agnostic to the subsequent DL algorithm. We want to provide a pipeline for both classification and segmentation algorithms. Due to the large file size of WSIs and the resulting slow processing time of existing solutions, we prioritized processing speed during development. This work was presented at the *BVM Conference 2024* [117].

Whole Tissue Segmentation

To answer questions like "*How does the distribution of the tumor microenvironment compare with the total tumor amount?*", a tissue segmentation algorithm is required.

Due to hardware constraints, an entire WSI cannot be processed contiguously. The standard procedure relies on segmenting small regions of the WSI and then reassembling the results during postprocessing. In Chapter 4, we explore how the surrounding tissue context can be taken into account when segmenting only parts of a specimen. Our method, which we call "Valuing Vicinity", is inspired how pathologists analyze a tissue sample (zooming out and considering the surrounding tissue context). The method is designed to be integrated into any encoder-decoder segmentation network to ensure that new network architectures and developments in the CV field can be quickly translated. This work was published in *Computerized Medical Imaging and Graphics* in 2023 [76].

Enhancing Cell-Level Analysis: CellViT and Beyond

Inspired by the work of Chen et al. [44], who introduced a self-supervised trained Vision Transformer that learned cell representations, we propose a Vision Transformer-based architecture for instance-level cell segmentation, called CellViT (Chapter 5.2). We demonstrate that CellViT achieves state-of-the-art (SOTA) performance on the challenging pan-cancer PanNuke [84] dataset. Based on this, we extend CellViT into a comprehensive framework for cell segmentation in H&E-stained tissue samples (Chapter 5.3). The enhanced framework, CellViT^{++}, enables data-efficient fine-tuning for new cell classes, offers faster performance compared to CellViT, and can be trained on limited hardware. The initial version of CellViT was published in *Medical Image Analysis* in 2024 [119], while CellViT^{++} is currently under review and published as a preprint [116].

2 Background and Related Works

The methods proposed in this thesis fall into the intersection of computer vision and Artificial Intelligence. Following an introduction to the digital pathology workflow in Section 2.1, which covers sample acquisition, WSI structure, and file formats, we provide a concise overview of Deep Learning (Section 2.2). Although classification algorithms have already been introduced as a component of computational pathology pipelines, this thesis focuses on segmentation algorithms for tissue and nuclei, with background information given in Section 2.3. In addition, we explore the emergence of foundation models and their relevance to the field in Section 2.4.

2.1 The Digital Pathology Workflow

2.1.1 Whole-Slide Image Acquisition

The traditional pathology workflow as shown in Figure 2.1a involves a series of manual preparation steps, starting from tissue acquisition to the final diagnosis. First, a tissue sample must be acquired by biopsy or surgical resection [233]. The tissue is then cut and fixed to ensure that the structure is preserved over time [233]. This step includes fixing the tissue with formalin, dehydrating the tissue in ethanol to remove fluids, and lastly embedding the tissue in paraffin. The entire preparation

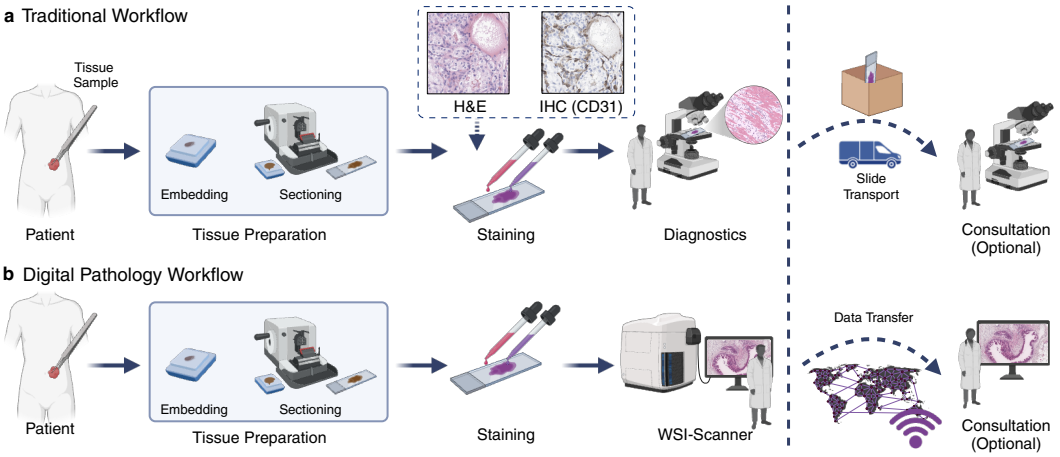


Figure 2.1: Comparison between the traditional pathology workflow and the digital pathology workflow.
a) Traditional pathology workflow with manual tissue preparation. Diagnostics under a microscope. Remote consultation involves the transport of glass slides. **b**) Digital pathology workflow. An additional digitization step is required, but data transfer over the internet is possible. Created with [57].

can take between 6-24 hours [103, 233]. In urgent cases, the tissue may also be frozen. Subsequently, the prepared tissue block is cut into $4\,\mu m$ to $5\,\mu m$ thick slices and placed on glass slides [103]. The tissue on the glass slide is then stained to increase the contrast or mark specific molecules like antibodies. Standard staining consists of hematoxylin and eosin (H&E). Hematoxylin stains basophilic structures, such as DNA in cell nuclei, RNA, and ribosomes, in a blue-violet hue [103]. Eosin stains acidophilic structures, such as the cytoplasm, in pink or red tones [103]. Combining these stainings results in color differentiation between cellular structures and the cytoplasm. An alternative to this is immunohistochemical staining (IHC), which is used to label proteins or antigens through the interaction of antigens with antibodies [135, p.3]. Examples include KI67 for the detection of cell proliferation, commonly applied in breast tumor diagnostics [296], and CD31 to visualize platelets, leukocytes, and endothelial cells [58, 257]. In the upper part of Figure 2.1a, an H&E and CD31 stained tissue sample is presented. Despite these advanced staining techniques, H&E remains the gold standard for diagnosis. The final slides are reviewed by a pathologist using a microscope. These microscopes usually offer a magnification of $\times 400$ (using $\times 40$ objective lens and $\times 10$ eyepiece) [202]. When referring to magnification, only the objective lens magnification is usually stated.

In comparison, the digital pathology workflow, shown in Figure 2.1b, consists of the same processing steps, with the addition of a scanning step to generate WSI. Scanners create high-resolution digital representations of tissue sections, often

exceeding a billion pixels (gigapixel scale). The evaluation of digital tissue samples can be performed directly on a computer monitor [297]. Digital pathology also enables functionalities such as algorithmic analysis, telepathology, rapid slide archiving, standardized education using reference WSI, or virtual second opinion consultations [16, 152, 206, 297]. Unlike physical slides that require shipping, WSIs can be shared over the internet (Figure 2.1). These digital images, however, present unique challenges regarding data formats and processing, as discussed in the following Sections 2.1.2-2.1.4.

2.1.2 Image Pyramids

Assuming a typical tissue size of $20 \times 15\,\mathrm{mm}$ and a scanner that scans with a resolution of $0.25\,\mu\mathrm{m/px}$, the scanned image would have a resolution of $80{,}000 \times 60{,}000\,\mathrm{px}$ [202]. With an encoding of 8 bits per color channel and three color channels (red, green, blue = RGB), the file size would be $1.152{\times}10^{11}\,\mathrm{bit} \approx 14.4\,\mathrm{GB}$ [202]. Under a microscope, a pathologist examines the tissue at different magnifications and field of view by zooming and moving the tissue in the x- and y-directions [202]. Replicating this behavior with a digital tissue scan involves two key challenges. (1) At low magnification levels, a large area of the tissue scan can be viewed at reduced resolution, allowing visualization of the tissue topology. However, since computer monitors have limited spatial resolution, e.g., $1{,}920 \times 1{,}080\,\mathrm{px}$, displaying the scan requires loading and downscaling the entire image. This is both computationally expensive and time-consuming [202]. (2) At high magnification levels, another property comes into place, as only a small part of the entire WSI can be displayed [202]. Continuing with the example above, only a $1{,}920 \times 1{,}080\,\mathrm{px}$ section of the entire $80{,}000 \times 60{,}000\,\mathrm{px}$ image would be displayed.

As a result, the so-called pyramid image format is used to save these images. An example of a WSI stored as an image pyramid is shown in Figure 2.2. In this format, the image is saved at multiple magnification levels [202]. Each pyramid layer represents a different magnification of the scanned tissue such that only the appropriate pyramid layer must be loaded during visualization [248]. Additionally, each layer is divided into tiles (tiled representation). Due to this, only tiles from the currently visible image section must be loaded when viewing and not the entire pyramid layer [202]. Although this is counterproductive in terms of file sizes, it reduces the loading and processing time when viewing the images [202, 248]. The file size is reduced by applying compression algorithms such as JPEG or JPEG2000 to the individual tiles [22]. In our daily work, depending on the tissue amount on the glass slide, typical WSI file sizes range between 1 and $3\,\mathrm{GB}$.

2.1.3 File Formats and Standardization

In radiology, the introduction of DICOM (Digital Imaging and Communication in Medicine) introduced a vendor-agnostic standard for image file formats [193].The

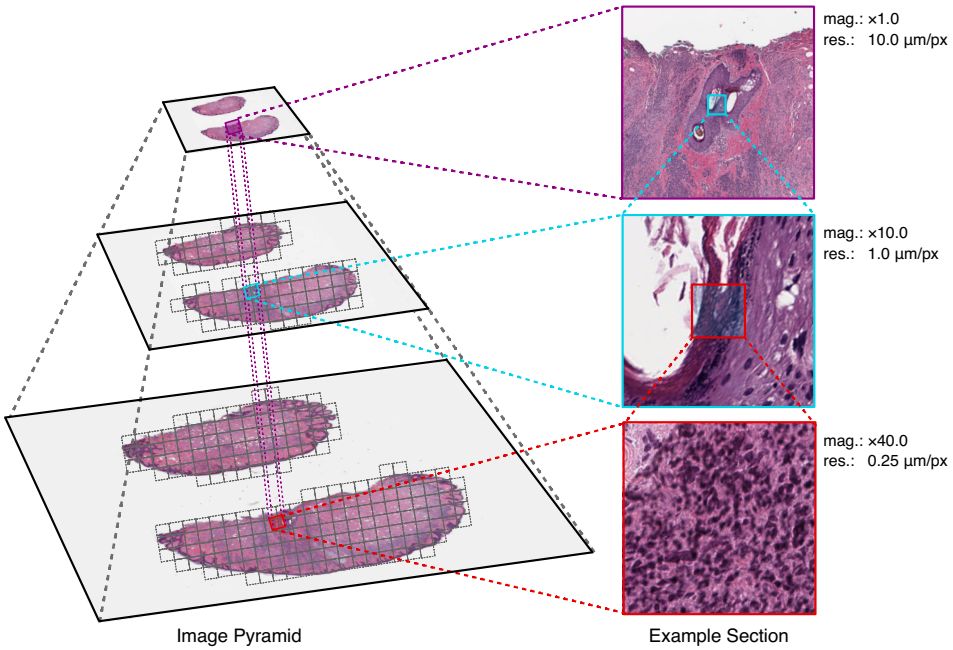


Figure 2.2: WSI pyramid consisting of three layers with increasing size. Each layer is tiled into smaller patches. Adapted from [22].

standard encompasses the definition of an image file format, including metadata, and the specification of a network protocol for data exchange [114, 193]. Almost all radiology vendors support it, but it has also been adopted in other medical fields [193]. In pathology, although the DICOM working group 26 has already defined a standard and provided implementation guidelines for WSI [202, 253], the discussion between pathologists and vendors about its implementation is still ongoing [114]. Moreover, the standard must be integrated into WSI scanners to support DICOM files and across all technical systems within the pathology department. Of course, this includes the pathological information system as a key component. Proprietary file formats still dominate the market, and widespread implementation of the DICOM standard has not yet been achieved [114, 198]. This situation presents challenges for WSI processing and its integration into clinical workflows.

Vendor-Formats Many WSI scanner manufacturers have developed proprietary file formats tailored to their specific systems [198]. Examples include 3DHISTECH with the `.mrxs` format, Leica with `.svs`, Olympus with `.vsi`, Ventana with `.bif`, Zeiss with `.czi`, and Hamamatsu using multiple formats such as `.vms`, `.vmu`, and `.ndpi` [90, 198]. Some of these proprietary formats, e.g., `.svs`, rely on the TIFF (`.tiff`) structure, extended by vendor-specific metadata fields and file ex-

tensions [210]. However, exact specifications must be requested directly from the respective manufacturer, as for Leica defined in their standard [158]. As explained in Section 2.1.2, these formats generally organize image data as a pyramid within a single file [158, 210], with each layer consisting of multiple tiles [210]. However, not all WSI formats are based on TIFF (e.g., Zeiss's `.czi` format). Furthermore, manufacturers differ in whether a WSI is stored as a single file, as is typical for TIFF-based formats, or as multiple files per WSI. For example, 3DHISTECH (`.mrxs`) stores each pyramid layer in a separate file within a directory [251].

For processing these WSI in proprietary formats, OpenSlide [91] is a widely used framework. However, it does not support all file formats, and crucial metadata relevant to the clinical context may remain inaccessible [114]. In general, the number of proprietary formats poses a challenge for interoperability, carries a risk of vendor lock-ins, and hinders the integration of digital solutions into the clinical workflow [114]. Thus, a standardized data format is needed for computational pathology, such as DICOM [114].

DICOM Developed by working group 26 in 2010, the DICOM Supplement 145 defines the DICOM standardization for digital pathology. It outlines how WSI should be structured as image pyramids, defines the access patterns of tiled images as multi-frame objects, and communication protocols for storage [246, 253]. Despite its potential, the standard leaves room for interpretation, leading to variability in implementations [246]. Due to this, accompanied by technical hurdles, the adoption of the standard has been slow. The Leica GT450DX scanner is the first scanner to store images in DICOM format directly and is FDA approved for digital diagnosis since 2024 [244], 7 years after the first FDA-approved scanner by Philips [77].

Despite the development and FDA approval of the first solutions supporting DICOM, hospitals face the challenge of a heterogeneous IT landscape. As IT systems evolve over time, they require DICOM integration, and previously purchased scanners may lack DICOM support. Therefore, data harmonization is necessary. Existing WSI can be converted to DICOM using tools like BioFormats [169], WSIDICOMIZER [125], VIPS [184], and PixelMed [55]. However, these tools have several limitations, including extended processing time, limited vendor support, and loss of metadata.

2.1.4 Whole-Slide Images and Computation

But how are WSIs processed for DL applications? As we know, their large spatial size, pyramidal data structure, and non-standardized file formats pose processing challenges. On the hardware side, DL algorithms utilize Graphics Processing Units (GPUs) because they perform matrix operations more efficiently than the well-known Central Processing Units (CPUs). However, analyzing entire WSIs with GPUs is computationally infeasible. Hence, patch-based processing is commonly

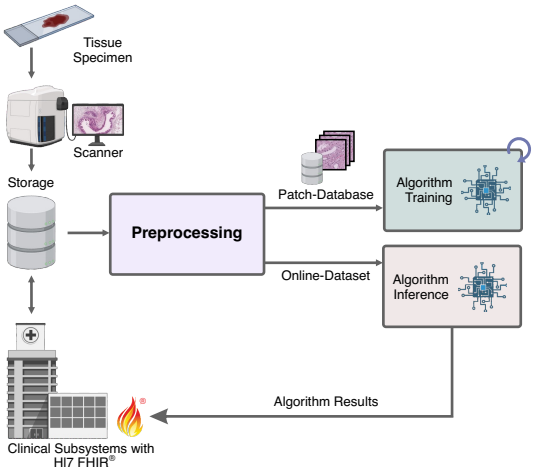


Figure 2.3: Preprocessing in CPath workflows including DL. Created with [57].

used, where WSIs are divided into smaller image sections that are the input for DL algorithms. Patches are usually extracted at different resolutions and may or may not have overlapping regions, depending on the task. At higher resolutions, it is possible to see fine-grained details like nuclear morphology, but at the cost of tissue context to recognize broader patterns [248]. As an example, for classification tasks, a patch size of $256 \times 256\,\mathrm{px}$ or $512 \times 512\,\mathrm{px}$ with no overlap and a resolution of $0.5 - 2.0\,\mu\mathrm{m/px}$ (corresponding to $\times 20 - \times 5$ magnification) is commonly used [29, 72, 73, 177, 284]. To capture broader tissue patterns such as tumor areas, more context may be required [76, 240, 265], resulting in patches acquired at resolutions $> 1\,\mu\mathrm{m/px}$ ($\times 10$) or even multiscale patches. In contrast, nuclei segmentation requires high resolutions up to $0.25\,\mu\mathrm{m/px}$ ($\times 40$).

Besides the patch-based methodologies, compression-based approaches have been proposed to process WSIs in an end-to-end fashion [65, 258, 259]. However, these methods are not yet widely used in practice.

Although WSI preprocessing is a central component of CPath pipelines, few works specifically focus on WSI preprocessing [117]. Often, it is implemented as a necessary step in the ETL (extract, transform, load) pipeline and not optimized. Figure 2.3 highlights the central role of preprocessing, which must accommodate diverse file formats from various manufacturers while enabling adaptable patch extraction tailored to the requirements of downstream DL algorithms, such as classification, tissue segmentation, and nuclei segmentation. The preprocessing requirements differ between the algorithm development and deployment phases. During development, persistent training datasets must be extracted. During inference, throughput and stability should be prioritized.

Contribution: Chapter 3 introduces a versatile preprocessing pipeline that satisfies the requirements of DL algorithms for classification and segmentation tasks. It overcomes limitations of existing solutions, such as functionality and processing speed. We evaluate the suitability of the existing DICOM conversion approaches for converting the vendor-agnostic format into DICOM. The overall objective is to design a preprocessing framework that supports the development and deployment of algorithms within CPath workflows.

2.2 A Short Primer on Deep Learning

Deep Learning is a subset of Artificial Intelligence, specifically Machine Learning (ML). In classical ML, input features are first defined by experts and then provided to an algorithm to derive predictions based on those features. An example of ML would be to selectively define laboratory values, such as blood glucose and cholesterol levels, to predict the risk of developing cardiovascular disease. In contrast, in DL, the features do not need to be selected and are learned by the algorithm itself. Thus, in our example, the laboratory values do not need to be preselected, but rather all available time-series data of all laboratory tests are used. The algorithms extract features and learn patterns from the data.

The learning principles of ML are typically divided into the three main approaches "supervised learning", "unsupervised learning", and "reinforcement learning". Due to its relevance for our work, this Section will focus on supervised learning.

Mathematically[1], the problem can be formulated as an optimization task consisting of the core components *data*, *model class*, *loss function*, and *optimization algorithm* [53]. The input *data* is defined as

$$\mathcal{X} = \{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_N\}$$

with N being the number of data samples. Each data sample $\boldsymbol{X}_i$ has an associated label $\boldsymbol{Y}_i$. The set of labels is defined as

$$\mathcal{Y} = \{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_N\}\,.$$

The goal is to find a *model class* f_θ that approximates the mapping from input data to output labels

$$\mathcal{X} \to \mathcal{Y}, \qquad f_\theta : \mathcal{X} \to \hat{\mathcal{Y}},$$

[1]The mathematical formulation is based on the lecture of Prof. Dr. Sen Cheng, Artificial Neural Networks, at the Ruhr University Bochum, winter semester 2020/21.

where θ defines the model's parameters and $\hat{\mathcal{Y}}$ denotes the model prediction set [53]. To define the optimization problem, a *loss* (or cost) function, generally defined as

$$\mathcal{L}(\theta, \mathcal{X}, \mathcal{Y}) = \sum_{i=1}^{N} l_{\text{pair}}(\theta, \boldsymbol{X}_i, \boldsymbol{Y}_i) = \sum_{i=1}^{N} l_{\text{pair}}(\hat{\boldsymbol{Y}}_i, \boldsymbol{Y}_i), \tag{2.1}$$

is necessary. It is a measurement of the discrepancy between the predicted labels $\hat{\boldsymbol{Y}}$ and the ground truth labels $\boldsymbol{Y}$. The loss function (2.1) needs be minimized to find the optimal parameter set $\hat{\theta}$ [53]:

$$\hat{\theta} = \arg\min_{\theta} \ \mathcal{L}(\theta, \mathcal{X}, \mathcal{Y}). \tag{2.2}$$

This process is called training the model. The necessary but not sufficient condition to solve eq. (2.2) is

$$\nabla_{\theta} \mathcal{L}(\theta, \mathcal{X}, \mathcal{Y}) = 0, \tag{2.3}$$

which can often just be solved numerically. The algorithms used to perform this *optimization* are called optimizers. One common example is the Gradient Descent algorithm, which iteratively updates the parameters θ to minimize the loss function.

In the early stage of ML, the models had a limited amount of parameters (e.g., Logistic Regression). Nowadays, DL networks belong to the class of artificial neural networks. They have been inspired by neurons (units) and their connectivity in the human brain [171]. These units are arranged in layers, and the information flows from layer $l-1$ to layer l. Units within the same layer are not connected in between. The activation a_i^l of a unit i in layer l is computed using a non-linear activation function

$$a_i^l = \sigma(\mathcal{W}, \mathcal{B}, \boldsymbol{X}) = \sigma\left(\sum_j w_{ij}^{l-1} a_j^{l-1} + b_i^{l-1}\right),$$

where w_{ij} are the weights connecting the previous layer's units j to unit i, and a bias b_i [53, 171]. The sets $\mathcal{W}$ and $\mathcal{B}$ represent all the weights and biases in a network and are combined to the parameter set $\theta = \{\mathcal{W}, \mathcal{B}\}$ [53, 171]. In total, the network has L layers. The activation function $\sigma(\cdot)$ is non-linear such that the network can approximate arbitrary mappings $\mathcal{X} \rightarrow \mathcal{Y}$. Optimization of the parameters θ is performed using an algorithm called "backpropagation", which propagates the loss backward through the network iteratively using the gradient [53]. An example of a neural network is given in Figure 2.4a. Based on this concept, a large number of model classes have been developed, all of which can be categorized into artificial neural networks.

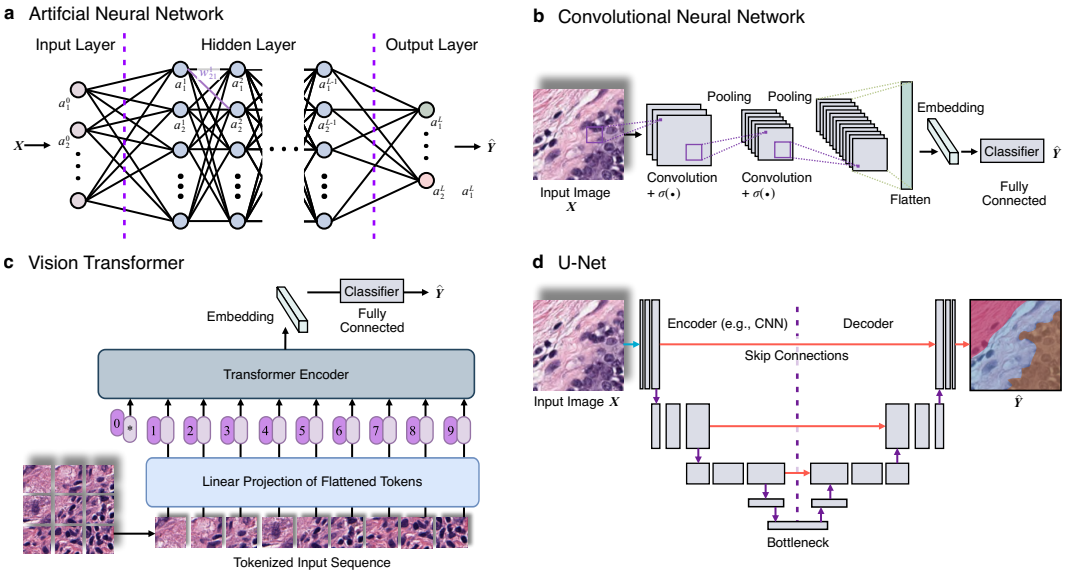


Figure 2.4: Overview of DL networks.
a) Artifical neural network with multiple hidden layers. **b**) Convolutional neural network consisting of convolutions and pooling operations. **c**) Vision Transformer networks with input tokens. Since ViTs are permutation invariant, position encodings are added to the input tokens (Adapted from [67, 119]). **d**) U-Net network structure for semantic segmentation.

The primary model classes for image processing include convolutional neural networks (CNNs), Vision Transformers (ViTs) [67], and encoder-decoder-based segmentation networks like U-Net [234] or DeepLab [41, 43] (Figure 2.4).

CNNs extract local image features through convolutional layers and pooling operations (inspired by the visual cortex). These convolutional layers learn filter operations, such as edge or texture detections, and slide over the image to extract them [171]. As multiple convolution layers are stacked hierarchically, more complex patterns are extracted and aggregated from layer to layer (Figure 2.4b). Examples of popular CNN architectures include ResNet [112] and EfficientNet [255].

A key constraint of CNNs is their limited visual context due to the size of the convolution kernels. Inspired by language processing, ViTs use self-attention mechanisms (Transformer architecture), in which small image patches (called tokens) are related to all other image patches. Given an input image $\boldsymbol{X} \in \mathbb{R}^{H \times W \times 3}$ with height H and width W, N flattened input patches $\boldsymbol{X}_{\mathrm{P}} \in \mathbb{R}^{N \times (P^2 \cdot 3)}$ are extracted and projected into a latent space $\boldsymbol{Z}_0 \in \mathbb{R}^{N \times D}$ with dimensions D [67]. The Transformer encoder consists of blocks [266], each one with multi-head self-attention as a central component. For each head h, the tokens are mapped to queries ($\boldsymbol{Q}_h$), keys ($\boldsymbol{K}_h$) and

values ($\boldsymbol{V}_h$) and the attention score are calculated by

$$\text{Attention}_h(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = \text{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{D}}\right)\boldsymbol{V}_h, \ \text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_j}} \tag{2.4}$$

followed by normalization layers and linear layers with a ReLU activation function [266]. The softmax function in eq. (2.4) normalizes the attention values to be in the range $[0, 1]$ and sum up to 1 [266]. Simply said, the attention mechanism allows ViTs to compute pairwise interactions between image patches (tokens) to capture global context dependencies, as each token attends to all other tokens in an image. Section 5.2 provides a detailed derivation. An illustration of the ViT architecture is given in Figure 2.4c. Various ViT models exist with differing parameter counts, typically denoted as small (ViT-S), base (ViT-B), large (ViT-L), and huge (ViT-H) in increasing order. CNNs and ViTs serve as image encoder networks because they extract feature (embedding) vectors from input images used for tasks like classification.

For segmentation tasks with the aim to assign each pixel to its corresponding class, encoder-decoder networks, such as the U-Net are commonly used (Figure 2.4d). U-Net uses skip connections to link the encoder to a decoder. This allows for precise semantic segmentation at high resolution. The bottleneck layer of the encoder enables the network to learn an abstract representation of the image in order to use global information for segmentation [171]. Another popular approach is the DeepLab architecture [41, 42, 43]. For other computer vision tasks like object detection, additional network architectures exist, e.g., R-CNN [89], YOLO [230], RetinaNet [168], DETR [32], but these are not relevant for this work.

This is only a brief introduction to the topic. For more information, see the article from LeCun et al. [156] and the textbook from Goodfellow et al. [92].

2.3 Multiscale Segmentation Approaches in Digital Pathology

The following Section 2.3.1 is based on the related works presented in our *Computerized Medical Imaging and Graphics* publication from 2023 [76]. Section 2.3.2 is based on our CellViT/CellViT^{++} publications from 2024 [119] and 2025 [116].

2.3.1 Macro-level Analysis: Whole Tissue and Region Segmentation

Semantic segmentation in medical imaging allows physicians to precisely identify and delineate distinct structures or disease regions within an image. In histopathology, several works have shown that quantitative tissue analysis provides insight into the characteristics and internal mechanisms of tissue and uncovers biomarkers [271]. Grünwald et al. [99] demonstrated that intratumoral heterogeneity in the stroma

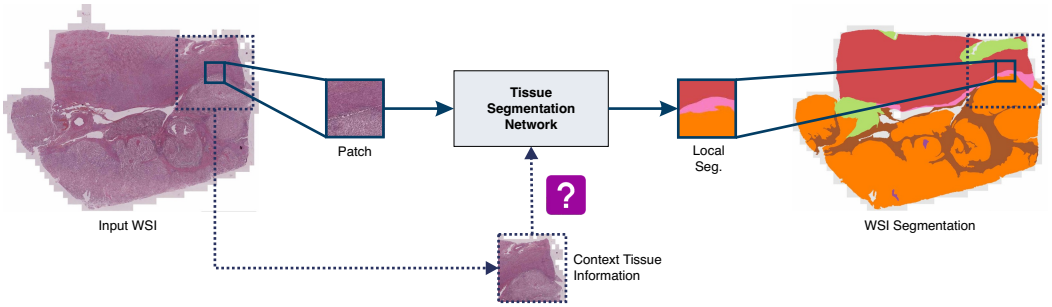


Figure 2.5: Overview of patch-based WSI segmentation. Given a central patch (solid line), context information about the surrounding tissue (dashed line) needs to be integrated into the segmentation algorithm.

in pancreatic cancer influences tumor immunity, differentiation, and treatment response. Junttila et al. [132] already described in 2013 that the composition of the tumor microenvironment is a predictive marker and should be quantified. Many more works reveal prognostic and predictive markers based on fine-grained tissue segmentation [13, 151, 219, 237, 271]. However, manual annotation is tedious and cannot be scaled to a clinical routine setting, underscoring the need for automatic segmentation methods.

As mentioned in Section 2.2, segmentation networks with encoder-decoder structures are used to predict dense segmentation maps. The encoder creates an internal representation of the image that the decoder upsamples to predict a segmentation mask. In addition, skip connections are employed to enhance the precision of the segmentation outputs. The CNN-based encoder-decoder networks include SOTA solutions like U-Net [234] and DeepLabV3 [42]. Variants of both networks have been found to be successful for medical imaging segmentation of different modalities. For instance, nnU-Net [128] is a popular framework that can be used for a variety of medical image segmentation problems, particularly in radiology. Published in 2020, nnU-Net has accumulated more than 4,500 citations in five years, and the open-source code received more than 6,000 stars on GitHub and over 1,800 forks, underlining the influence on the field [93, 190].

Due to the high resolution of WSIs, the semantic segmentation of an entire WSI poses a considerable challenge. Applying SOTA segmentation algorithms directly is impeded by GPU hardware limitations. Naively, one could segment the WSI by downscaling it to a low resolution. However, this leads to a loss of fine-grained tissue morphology and is not a viable approach. Therefore, patch-wise segmentation algorithms are applied [273]. Jin et al. [130] showed that the patch-wise segmentation quality depends on two factors: Patch resolution and field of view (spatial extend). They developed a module that dynamically learns to select the best trade-off between both settings. In total, the following question arises:

How can we ensure that the information from surrounding tissue is taken into account when segmenting small patches of a WSI such that the global information about the tissue composition can be integrated into the local segmentation?

Figure 2.5 shows a WSI with a patch (solid line) for segmentation. The tissue context (dashed line) should be considered during the segmentation process, as it contains crucial information about the surrounding tissue composition. Several works integrate contextual information to expand the field of view (FOV), following the same scheme [100, 162, 165, 240, 262, 265]. Given a central patch for segmentation, a concentric context patch with lower physical resolution is integrated into the network. One of the first methods used patch classification but produced only coarse segmentation maps [165]. Li et al. [162] proposed a multiscale U-Net model that fused three FOV images at the input layer, resulting in a threefold increase in computational time and limiting fusion to the first network layers. Following methods improved on this idea by fusing context information at the bottleneck layer or decoder of the U-Net, but this required the integration of multiple encoders for each FOV [240, 265]. Van Rijthoven et al. [265] proposed a new approach (HookNet) that employs a dual encoder-decoder architecture for both the patch and the context image. The information is fused between the decoders by spatially aligning the feature maps of the context branch with those of the patch segmentation branch. In the same manner, Schmitz et al. [240] suggested the msY-Net, but they employed a single decoder branch to enhance the computational efficiency. By 2021, both models outperformed previous methods, however, with some drawbacks. Specifically, the computational cost of these models is much higher than that of single encoder-decoder networks. Compared to the baseline U-Net with $17.80\,\mathrm{M}$ parameters, the msY-Net has $40.76\,\mathrm{M}$ parameters [240], which increases the training and inference time. Furthermore, these network architectures are highly specialized (e.g., msY-Net is based on a ResNet-U-Net combination and HookNet is an adapted U-Net). Transferring them to possible upcoming architectures and encoder networks from the CV domain might not be applicable.

Thus, the initial issue remains partially unsolved: There is a need for a CPath segmentation framework that efficiently integrates contextual information, has low computational overhead, and allows adaptive segmentation architectures (e.g., U-Net [234], DeepLabV [41], U-Net++[302], DeepLabV3Plus [43], SegFormer [289]) and image encoders (e.g., ResNet [112], EfficientNet [255], Transformer [67]). One promising development is the idea of adapting the attention mechanism to the CV domain [40, 67, 123, 266]. In the past few years, several papers proposed a combination of CNNs and the attention mechanism [40, 102, 272]. These approaches are not targeting to increase the FOV for local images but increasing the receptive field of CNNs. For instance, Chen et al. [40] incorporated a Transformer in the bottleneck layer of the U-Net (TransUNet) to enable feature maps to attend to

all image regions. Guo et al. [102] integrated an external attention mechanism combined with a dataset memory into the U-Net such that the essential information of the entire dataset could be integrated when segmenting one sample. Even further, Wang et al. [272] combined the external memory by Guo et al. [102] and the internal attention mechanism of the TransUNet [40] with a local attention mechanism to capture local (small region), global (entire image), and external (entire dataset) context. While these works aim at capturing long-range and spatial context information, they still assume that an entire image can be processed in a single processing step.

Contribution: Many approaches have been proposed to address the problem of incorporating contextual information during patch-wise WSI segmentation. However, the computational effort is high, and the improvements are marginal. We propose a lightweight tissue context approach based on self-attention and tissue embeddings. Unlike previous methods that rely on large FOV context patches, we introduce a novel external context memory framework that stores compressed representations of the entire tissue in Chapter 4.1. Our method integrates into existing encoder-decoder frameworks such as U-Net or DeepLabV3, enhancing their segmentation performance for WSI. Evaluation is performed on an internal dataset for the segmentation of kidney cancer and two public datasets for tumor segmentation in breast and liver cancer. Additionally, in a clinical context, we trained a network for pancreatic cancer and correlated the histological patterns with patient survival.

2.3.2 Micro-level Analysis: Cellular Segmentation

Cells are the building blocks of the human body [61, p. 18]. Each human cell (eukaryotes) contains a nucleus, surrounded by organelles such as mitochondria and the endoplasmic reticulum within the cytoplasm [61, p. 19]. All these components are enclosed by a plasma membrane, which defines the cell boundary [61, p. 19]. The membrane remains unstained by H&E, such that segmenting entire cells, including their cytoplasm, is challenging [39, 228]. In contrast, nuclei are stained blue-purple by the hematoxylin and are clearly distinguishable. The nucleus contains all the genomic information of the organism, and its morphology reflect the type of cell and the state of the cell cycle (mitosis) [61, p. 35]. Segmentation of nuclei enables the extraction of quantifiable cellular morphologies, such as shape, size, and texture, which can be correlated with clinical outcomes, used for diagnosis, or applied to study tumor-immune interactions [68, 291]. Examples include detecting tumor-infiltrating lymphocytes or inflammatory cells in the tumor microenvironment [99, 236, 252]. However, manual large-scale nuclear analysis is

time-consuming and prone to high intra- and inter-observer variability such that automated approaches are needed [291]. For the purpose of this work and in line with the current literature, cell segmentation specifically refers to the segmentation of nuclei.

In contrast to tissue segmentation, nuclei segmentation involves not only semantic segmentation into different cell classes but also instance-level semantic segmentation in which individual nuclei must be distinguished. This task is commonly referred to as panoptic segmentation [141]. Nuclei segmentation methods can be divided into conventional feature-based and emerging DL-based approaches.

Conventional Feature-based Methods Feature-based approaches rely on classical image processing algorithms and features such as intensity, texture, shape, and the morphological properties of nuclei. A significant challenge in these methods is the separation of overlapping nuclei, for which various techniques have been developed [3, 52, 167, 181, 256, 267, 282, 294]. These techniques range from predefined nuclear geometries combined with watershed algorithms [3, 52, 267], to approaches involving morphological operations [282] or ellipse fitting [167]. A major drawback of these techniques is their reliance on expert knowledge, sensitivity to hyperparameters, and lower performance compared to recent DL approaches [46, 95].

CNN-based DL Methods As demonstrated in Section 2.3.1, the U-Net architecture is the primary architecture for semantic segmentation. However, for the panoptic segmentation of cell nuclei, the original U-Net implementation cannot separate clustered nuclei [95]. In the current literature, DL algorithms for nuclei instance segmentation are further divided into two-stage and one-stage methods [124]. Two-stage methods incorporate a cell detection network in the first stage to localize cell nuclei within an image, generating bounding box predictions of nuclei [147, 172, 249]. These detected nuclei are then passed on to a subsequent segmentation stage to retrieve a fine-grained nucleus segmentation. Mask-RCNN [111] is one of the leading two-stage models built on top of the object detection model Fast-RCNN [88]. Koohbanani et al. [147] utilized Mask-RCNN networks for nuclei instance segmentation. Another two-stage method for nuclei segmentation is BRP-Net [249], which creates nuclei proposals in the first place, then refines the boundary, and finally creates a segmentation out of this. This process is computationally intensive and takes 12 minutes to segment a $1{,}360 \times 1{,}024$ px image and is therefore not scalable to WSI with gigapixel resolutions [249].

In contrast, one-stage methods such as HoVer-Net [95], Stardist [277], CPP-net [46], DCAN [38], TSFD-Net [124], and more [204, 229] adopt a different approach by directly predicting the nuclear contour or using additional prediction maps to separate nuclei. HoVer-Net, for example, uses a decoder that is supposed to predict not only the semantic mask but also the horizontal and vertical dimensions

of a nucleus in order to separate overlapping nuclei. In particular, the StarDist and HoVer-Net approaches show promising results in the panoptic segmentation of cell nuclei and outperform two-stage detection models in terms of accuracy and runtime.

ViT-based DL Methods One of the main reasons ViTs have not been adopted for medical image processing at a large scale is their reliance on large training datasets, as they lack inductive biases for image processing [67, 254]. Unlike CNNs, which are inherently translation and rotation invariant and extract image features hierarchically based on stacked convolutions, ViTs must learn these properties during training [186, 254]. On the other hand, when provided with sufficient large datasets, the patterns learned by ViTs are more meaningful [225]. In the medical context, annotated image data for training is rare. To overcome this limitation, a typical approach is to pretrain (supervised) a network on a large image dataset, typically natural images like ImageNet [59], and fine-tune the network on the target medical data. This method, consisting of pretraining and fine-tuning, has also been proven to be successful for CNNs and is a standard procedure in computer vision (Figure 2.7). Other approaches try to circumvent the domain shift between natural and medical images by directly pretraining on the target domain, such as pretraining a network for classifying tissue patches directly on tissue data. Since there are no or only a few labeled medical imaging datasets available at a large scale [163], self-supervised learning methods (SSL, Section 2.4) emerged.

While searching for a suitable model for cell segmentation, we came across the work of Chen et al. [44]. In their paper, they introduced a hierarchical ViT-based model pretrained via SSL on $10{,}678$ WSI, equivalent to $104\ \mathrm{M}\ 256 \times 256$ tissue patches [44]. The model, named Hierarchical Image Pyramid Transformer (HIPT), was developed as a feature encoder for WSIs to learn a meaningful internal representation of tissue and morphology.

When analyzing their first extraction stage (called HIPT-256) for embedding patches of size 256×256 at $0.50\,\mu\mathrm{m/px}$, the authors claimed that some of their attention blocks attend to cell nuclei and the network learned cell-tissue interactions. In one of our works published in 2023 about the classification of WSI, we were able to confirm this behavior [118]. An illustration is given in Figure 2.6.

Initially, we planned to integrate their HIPT-256 model directly into an existing framework to test its performance for cell segmentation. Although the ViT architecture is generally well suited for segmentation tasks, exemplified by models such as TransUNet [40], UNETR [107], Swin-UNETR [106], and SegFormer [289], none of these networks currently supports panoptic segmentation. Nonetheless, we see potential of using in-domain pretrained ViTs such as the HIPT-256 for cell segmentation.

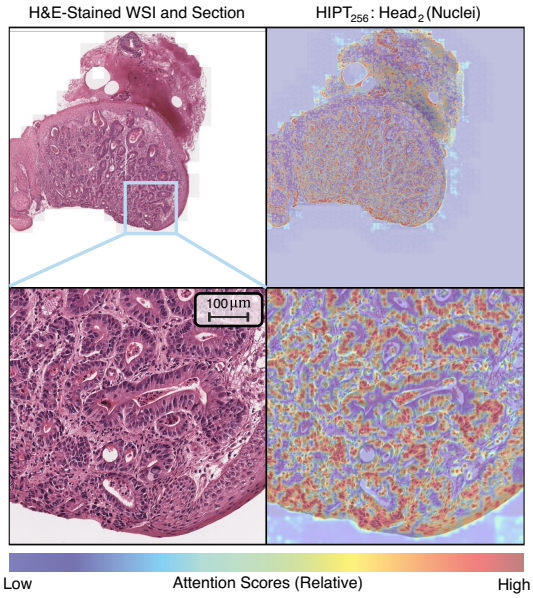


Figure 2.6: HIPT-256 attention visualization on our internal MEMORI cohort for the second Transformer head of the last block. To derive attention heatmaps, a forward pass of the patches through the network is performed, and the attention scores are stored for visualization. Adapted from [118]

Contribution: While all the recent methods for panoptic nuclei segmentation employ CNN-based networks, we explore the capability of pretrained Vision Transformers as an alternative for the task of nuclei segmentation. ViTs are especially effective in encoding contextual information and fine-grained structures within complex tissue images, which are important for the accurate identification of individual nuclei. To the best of our knowledge, we are the first to use ViTs as the encoder networks for panoptic nuclei segmentation of H&E stained WSIs (Section 5).

2.4 Self-Supervised Learning and the Advent of Foundation Models

The conventional pretraining and fine-tuning strategy using task-specific CNN models (Figure 2.7) has reached saturated performance. This saturation arises primarily from the lack of large annotated datasets for pretraining, domain shifts between pretraining and target domain, and the limited receptive fields of CNNs, which restricts their ability to capture global patterns. In contrast, ViTs have demonstrated the capability to learn more meaningful, and in particular, global patterns compared to CNNs when trained on massive datasets [225].

Thus, the following question arises:

If ViTs learn more meaningful representations when trained on sufficiently large datasets, how can we address the shortage of large-scale annotated training data?

Since 2020, approaches have emerged that leverage unlabeled datasets for pretraining [8]. These approaches generate labels directly from the data and train models on these data-label pairs, a paradigm known as self-supervised learning (SSL). The automatic label generation during training removes the requirement for manually curated pretraining datasets, allowing entire image archives to be used [56, 243]. SSL methods have matched, and in some cases surpassed, the performance of models trained on labeled images, even on benchmarks such as ImageNet [8]. Various SSL strategies have been proposed.

Contrastive learning approaches like SimCLR [48] learn visual features by maximizing similarity between two augmented views of the same image (positive pair) and minimizing similarity between augmented views of different images (negative pairs) [8]. A drawback of SimCLR is that the algorithm requires large batch-sizes ($>$ 4,096) during training and relies on negative samples [8]. Especially in the medical domain, the definition of negative pairs can be ambiguous (e.g., are two views of different chest CT scans positive or negative pairs?). Despite this, some studies have successfully employed SimCLR as a pretraining strategy in the medical domain [120], including CPath [54]. Momentum Contrast (MoCo) [110] and its variants [49, 51] improve SimCLR by incorporating a momentum encoder and a negative sample dictionary queue, allowing the use of fewer negative pairs [8].

Self-distillation approaches, such as BYOL [98], SimSiam [50], DINO [34], DINOv2 [211], and iBOT [301], also learn representations through augmented views. Unlike contrastive methods, they do not rely on negative pairs such that they are more practical for medical domains and computationally more efficient due to reduced batch-sizes. Especially DINO and DINOv2 have been proven to be an effective method for pretraining ViTs. Caron et al. [34] showed that the self-supervised learned ViTs features not only learn meaningful representations useful for zero-shot classification but also contain explicit information about semantic segmentation of the image.

Other SSL approaches in computer vision include correlation-based methods like DeepCluster [33], VICReg [12], Barlow Twins [298], and W-MSE [74]. Additionally, Masked Image Modeling strategies, such as MAE [109], Context Encoders [215], BEiT [11], and SimMIM [290], are frequently used. A comprehensive review of all mentioned strategies is given by Balestriero et al. [8].

While interest in SSL methods has grown, the term *foundation model* was coined. As the Stanford Institute for Human-Centered Artificial Intelligence states, "A foundation model is any model that is trained on broad data (generally using

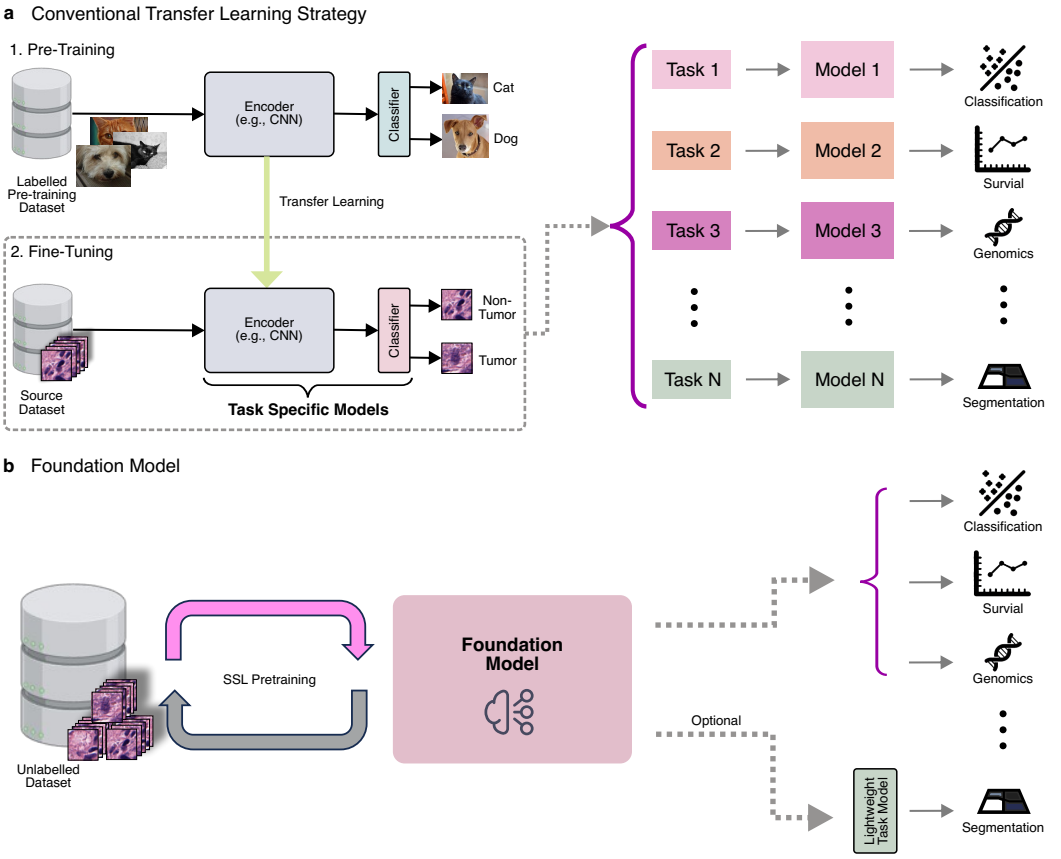


Figure 2.7: Comparison between classical DL and foundation models.
a) Conventional transfer learning strategy for DL. An encoder model is first trained on a labeled pretraining dataset, for example ImageNet. Then, the encoder is fine-tuned for each target task using annotated in-domain data. **b**) Foundation models are trained on a large unlabeled dataset and subsequently applied to multiple tasks with minimal adaptation. Created with [57].

self-supervision at scale) that can be adapted, e.g. fine-tuned, to a wide range of downstream tasks" [18]. Technically, this is not an innovation, as the networks are still based on existing architectures (mainly ViT in the image domain). The paradigm shift primarily concerns the application of these models. Instead of pretraining and then fine-tuning a model for a specific task, foundation models provide a common basis for feature extraction for various downstream tasks, with only minor task-specific modifications being made [18, 196]. They aim to serve as generalist models capable of addressing a wide range of tasks based on learned representations. A comparison between the classic DL approach and the foundation models is given in Figure 2.7. Beyond well-known natural language processing models like GPT [23] or LLaMA [263], there are also foundation models emerging in the areas of CV and CPath.

As for the "natural image"-domain", Kirillov et al. [142] recently published a promptable segmentation model as a foundation model [18] for semantic segmentation, also known as "Segment Anything" (SAM). The SAM model comprises an image encoder (ViT) and a lightweight mask decoder network. The final backbone (ViT-H) of SAM was trained supervised on 1.1 billion segmentation masks from 11 million images. A three-stage data engine consisting of assisted manual, semi-automatic, and automatic mask generation acquired this extensively annotated dataset. Next to segmenting natural images, the SAM model has also shown remarkable performance in medical image segmentation [157].

Within histopathology, the HIPT model [44], along with earlier CNN-based models by Ciga et al. [54] and Kang et al. [136], represents the first histopathological foundation models. All of these models have been published in 2022. Subsequently, an increasing number of models have been proposed, as shown in Table 2.1 (sorted by year of publication). Two key trends can be observed: While the earliest models were based on CNNs, all models since 2023 used the Transformer architecture, with a tendency toward larger models. Second, while the initial models were pretrained on public datasets, training on internal datasets comprising millions of slides now dominates. The computational resources and scanning effort required for such datasets can only be managed by a few institutions. However, many models, such as UNI [45], Virchow [270], Virchow-2 [303], and Prov-Gigapath [292], are publicly available and can be used for research. The models mentioned in Table 2.1 were evaluated on various benchmarks and showed excellent performance for tasks such as tumor detection, overall survival prediction, or detection of genetic alterations. Campanella et al. [30] recently published a benchmark using WSI acquired from routine clinical settings to externally validate CPath foundation models in tasks such as disease detection and biomarker detection. For disease detection, the bench-

Table 2.1: Overview of image only foundation models for CPath.
List is sorted by publication date. Adapted from [30]. ND: Not disclosed. *The best model in their study averaged across all tasks. **Semantically-relevant contrastive learning.

Model	Architecture	Param. (M)	Algorithm	Training Data Source	Tiles (M)	Slides (K)	Organs/ Cancer Types	Year	Public Weights	Cite
Lunit*	ViT-S	22	DINO	TCGA	19	21	ND	2022	Yes	[136]
Ciga	ResNet18	12	SimCLR	TCGA, Public data	ND	25	>19	2022	Yes	[54]
	ResNet34	22								
	ResNet50	26								
	ResNet101	45								
HIPT	Stacked ViT	24	DINO	TCGA	104	11	33	2022	Yes	[44]
CTransPath	Custom CNN	28	SRCL**	TCGA, Public data	16	32	25	2022	Yes	[274]
Phikon	ViT-B	86	iBOT	TCGA	43	6	13	2023	Yes	[81]
Virchow	ViT-H	632	DINOv2	Internal	2,000	1,488	17	2023	Yes	[270]
Campanella	ViT-S	22	DINO	Internal	1,600	423	42	2023	No	[28]
	ViT-L	303	MAE		3,200					
Rudolf-V	ViT-L	304	DINOv2	TCGA, Internal	1,200	134	14	2024	No	[64]
UNI	ViT-L	303	DINOv2	Internal	100	100	20	2024	Yes	[45]
Prov-GigaPath	LongNet	1,135	DINOv2	Internal	1,300	171	31	2024	Yes	[292]
Virchow-2	ViT-H	632	DINOv2	Internal	1,700	3,100	~200	2024	Yes	[303]

mark includes the identification of nine types of cancer, with models (Table 2.1) trained using DINO/DINOv2 demonstrating comparable high performance in all types of cancer [30, 31]. A larger performance gap was observed in the more challenging biomarker detection task, which requires identifying 11 biomarkers across breast (4) and lung (7) cancers. In breast tissue, all DINO/DINOv2 pretrained models perform similarly, while in lung tissue, UNI and Prov-GigaPath achieved consistently superior results [30, 31].

Contribution: Foundation models are characterized by their versatility. The current developments of foundation models are likely to have an enormous impact on digital pathology and clinical workflows. Based on our CellViT cell segmentation model in Chapter 5.2, we explore the application possibilities of foundation models for cell segmentation in Chapter 5.3. Inspired by the idea that foundation models should offer a wide range of applications, we propose to expand the conceptual scope of the term "foundation model". In our perspective, foundation models are also models that provide excellent results and generalization capacity in a specific domain and not only in a specific task. An example is the "Segment Anything Model", which can be considered as a foundation model for semantic image segmentation. We want to extend CellViT such that it becomes a foundation model for arbitrary cell detection and segmentation for digital pathology titled CellViT^{++}.

Part II

From Slide to Insight - Methods

3

A Comprehensive Framework for Whole-Slide Image Preprocessing

A fast and adaptable preprocessing pipeline is an essential part of digital pathology workflows and computational analysis. It is necessary for developing DL algorithms and significantly influences end-to-end processing time in clinical pipelines. Although real-time applications are not feasible at this time, we show that the processing time can be reduced by choosing image-loading back-ends and parallelized processing. We first define the software requirements for the preprocessing pipeline in Section 3.2. Sections 3.3 and 3.4 then explore the impact of parallel processing and the choice of image-loading back-end on the processing time

The following Sections are built on our BVM 2024 publication [117]. A transparency statement is attached at the end of this dissertation.

As the field of digital pathology continues to advance and algorithms are being developed, a versatile and fast preprocessing framework for WSI becomes increasingly important. To recap, WSIs are stored as pyramidal images in a tiled representation. Due to their size, they cannot be processed entirely as one image, such that most DL algorithms employ patch-wise processing. The example introduced in Section 2.1 illustrates the scale of patches per WSI that must be processed. Consider a WSI with a spatial resolution of $80{,}000 \times 60{,}000\,\mathrm{px}$. Using a patch size of $256 \times 256\,\mathrm{px}$ on the highest resolution would result in $73{,}555$ patches that must

be extracted and processed. Each millisecond spent per patch processing extends the overall processing time, in our example, by $1\,\mathrm{min}\,13.5\,\mathrm{s}$. Besides processing throughput, functionality is also crucial, e.g., patch extraction for segmentation and classification algorithms, automatic training mask generation from digital annotations, or compatibility with popular DL frameworks like PyTorch [17].

However, few works solely focus on WSI preprocessing. Often, preprocessing is implemented as a necessary step in the ETL algorithm pipeline without optimization. For instance, the work of Lu et al. [177] (CLAM) introduces preprocessing in the context of algorithm development. Their algorithm to classify whole WSI based on patches involves preprocessing using OpenSlide [91], but does not include multiprocessing and has interoperability limitations. An alternative is Deep-Histopath (D-Histo) [69], initially developed for a conference challenge and later modified for general use. This framwork, however, does not support annotations and is inappropriate for segmentation algorithms. PathFlowAI (PathAI) is an algorithm-agnostic solution that includes DL model integration, but not a standalone preprocessing tool [159]. In addition, further research has been proposed in the form of Python frameworks, including SliDL [17], FAU-DLM [205], histolab [182], and TIAToolBox (TIA) [222]. SliDL is a Python package to directly build data pipelines within PyTorch, but not a standalone solution (programming expertise required). Similarly, HistoLab adds features like tissue detection and pen marker filtering. TIAToolBox is another commonly used framework with a broad range of functionality but comes with the disadvantage of a relatively high runtime.

In this Chapter, we introduce a flexible Python-based WSI preprocessing pipeline that has been optimized for speed and functionality, named PathoPatcher. We compare PathoPatcher with other solutions like the popular TIAToolbox. Inspired by SliDL, we include an interface to the DL framework PyTorch. Our experiments using slides with varying tissue sample sizes demonstrate that our framework reduces processing runtime by up to 78%. Due to OpenSlide support, our framework is designed to handle WSIs from many scanning manufacturers with different (proprietary) file formats. As introduced in Section 2.1.3, a standardized file format for WSI is still missing, and the adaption of DICOM remains incomplete. Therefore, we evaluate whether vendor-specific formats can be converted into DICOM using open-source conversion tools and test the interoperability of the resulting DICOM files.

3.1 Integrating Preprocessing into Hospital IT Systems

A vision of the central role of preprocessing in the hospital IT landscape is visualized in Figure 3.1. Initially, slides are scanned during routine pathological procedures or digitized for retrospective cohorts. The files are either stored in the manufacturer's proprietary format or standardized through an intermediate step and saved as

DICOM files. When using proprietary file formats, downstream compatibility must be ensured. It is crucial to confirm that the files can be processed within Python as the go-to programming language for DL algorithms [75, 134]. This also applies to imported external WSIs. All WSIs are then stored on a long-term storage system. From there, WSIs can be queried and preprocessed as needed.

Two use cases should be considered. First, training DL algorithms requires that preprocessed datasets (consisting of hundreds to thousands of WSI) are stored on the training machine to reduce processing times for training models. For the development of segmentation models, it is also crucial to integrate annotations from widely used annotation tools, such as QuPath [10], into the preprocessing framework. Second, in-memory dataloader are essential for fast inference times when using trained algorithms in a clinical setting.

The algorithms' results, along with existing metadata, are stored in a database and can be retrieved and exchanged via an FHIR-compliant interface, such as the pathology information system. FHIR, which stands for Fast Healthcare Interoperability Resources, is a widley used standard for exchanging healthcare information electronically [14].

3.2 Methods

3.2.1 Software requirements

Efficient WSI preprocessing requires software that satisfies essential functional requirements and technical demands. Here, the requirements for preparing training datasets must be distinguished from those for performing inferences with DL

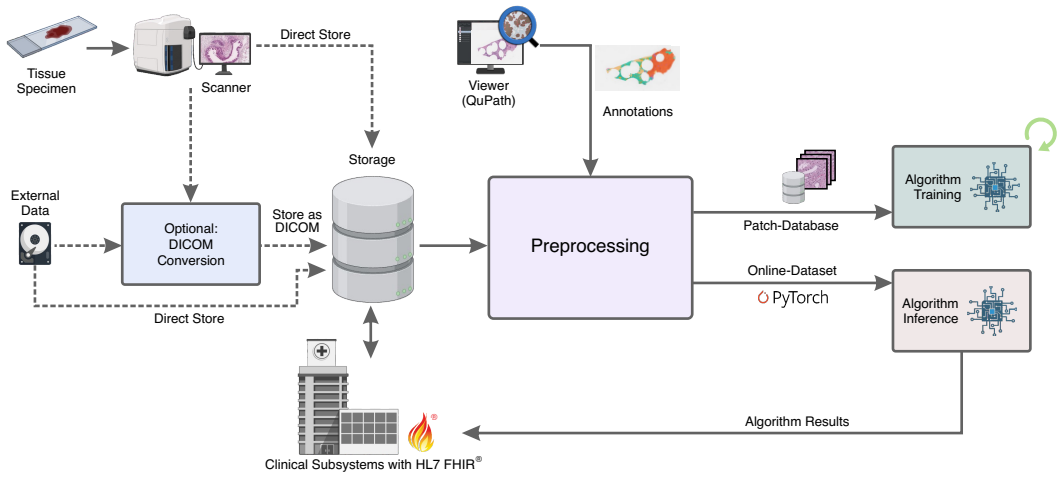


Figure 3.1: General workflow illustrating the processing of WSIs and integrating algorithmic results within a hospital setting. Created with [57].

algorithms. However, both variants demand the support of arbitrary patch sizes, overlap settings (sliding window), and allow the specification of magnification or pixel resolution to accommodate varying data extraction needs. In order to optimize computational resources by limiting processing to diagnostically relevant patches, tissue detection capabilities are necessary to exclude non-tissue areas. To enable applications with multiscale segmentation networks, there is a need for multi-resolution extraction of registered patches from consecutive image pyramid layers [76, 100, 162, 240, 262, 265]. Stain normalization can also be essential to standardize the appearance of WSI, minimizing the variability of differing staining protocols between laboratories that could otherwise introduce inconsistencies in computational analyses and could affect algorithm generalization [115, 191]. To ensure patch quality, it is important for the software to include features that at least detect or even remove manual markers on the slide.

When creating training datasets, particularly those intended for use in segmentation algorithms, annotations must be loaded and processed. In this regard, it is essential that annotations are compatible with the GeoJSON format used by QuPath [10], a widely used software for displaying and annotating WSIs.

From a technical perspective, the preprocessing speed is crucial given the size of WSIs, which may contain more than $100{,}000$ patches depending on the scanning resolution and preprocessing parameters. Efficient frameworks should ensure fast data access and support parallel processing to maximize performance.

With the widespread adoption of Python, specifically PyTorch, for DL and WSI processing, a Python-based framework would facilitate integration with established DL workflows.

3.2.2 Technical Implementation

Considering the previously outlined software requirements, we developed a preprocessing tool named PathoPatcher, using Python and widely used image processing libraries [20, 91, 264]. A detailed illustration of the PathoPatcher components, based on the workflow proposed in Figure 3.1, is given in Figure 3.2.

The software consists of four sequential components: Tissue detection, parallelized patch extraction, postprocessing including quality control and stain normalization, and data provisioning. Different image loading back-ends from multiple manufacturers have been integrated. In addition to OpenSlide, we integrated NVIDIA's CuCIM (Compute Unified Device Architecture Clara Image) [207] library and WSIDICOM [126] from the IMI-Bigpicture initiative [200]. All image loading back-ends are unified through a standardized API oriented on the OpenSlide framework, thus enabling flexible usage of all back-ends. OpenSlide can support multiple manufacturer formats, including WSI in DICOM format, as introduced in version 4.0.0. However, our experience revealed that this applies primarily to Leica's DICOM implementation. In contrast, CuCIM is only compatible with TIFF-like

formats, e.g., `.tiff`, and also Leica's `.svs`, but leverages GPUs for image processing. It employs an adaptive caching mechanism, which becomes critical if the extracted patches do not align with the internal tiling scheme of the WSI pyramid levels. In such cases, tiles may need to be reloaded multiple times, resulting in increased input/output (I/O) operations. The caching mechanism addresses this by storing previously accessed WSI tiles in the host system's random access memory (RAM), thereby reducing redundant I/O operations. As previously outlined in Section 2.1, DICOM is targeted as the reference standard for medical images. Thus, it is even more important that PathoPatcher supports DICOM. However, since numerous manufacturers have yet to implement an adaptation of the standard, a considerable number of images remain available in the manufacturers' proprietary file formats (Figure 3.2). We have incorporated an optional step into our pipeline to address this issue. This stage includes the WSIDICOMIZER [125] converter developed by IMI-Bigpicture to convert existing WSI into a DICOM-compliant WSI format. The current limitation is that the resulting DICOM WSIs are not yet compatible with the DICOM interface of OpenSlide due to modified metadata and non-standardized DICOM tag settings. To overcome this limitation, WSIDICOM from IMI-Bigpicture is used as an alternative back-end, when DICOM files cannot be processed by OpenSlide. This enables us to support a wide range of manufacturer formats and even standardize them to the DICOM format before processing.

After importing the slide metadata and selecting the appropriate image loading back-end, the first processing step is tissue detection, which utilizes Otsu's thresholding method [212]. In this step, markers on the digitized slide are checked. In particular, green and black markers can be removed from the tissue mask using classic color filters, and such patches can be filtered out. Subsequently, the patches

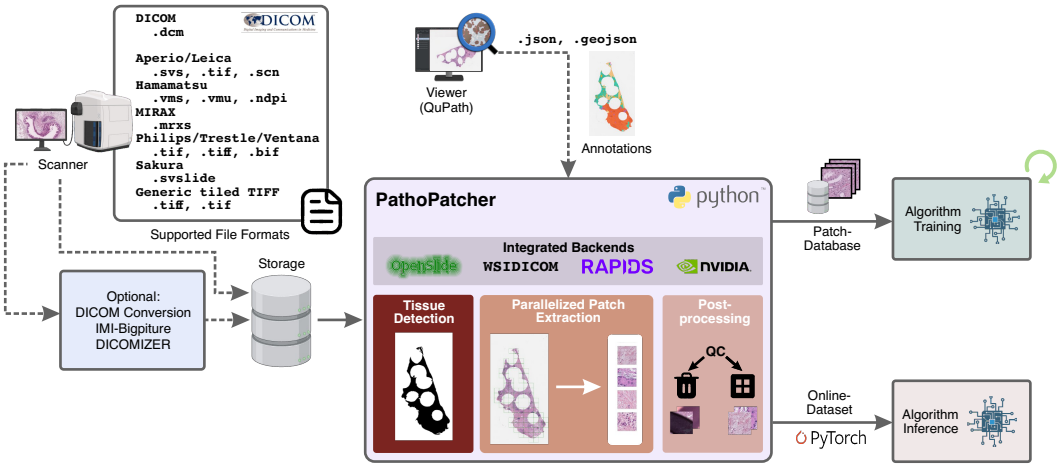


Figure 3.2: Illustration of the core components of PathoPatcher. Created with [57].

are extracted using the aforementioned image loading back-end. To ensure a fast runtime, this is done parallelized. The preprocessing parameters, including patch size, overlap, and context-patches, can be either set by a configuration file or directly via a command line interface (CLI). To support annotations, we implemented an interface to import annotations made in QuPath. Optional postprocessing includes stain normalization using the Macenko method [179] and quality control (e.g., filtering out patches with marker or blurriness).

The software design of PathoPatcher was guided by two primary use case scenarios for the application of algorithms: (1) development and training and (2) inference in clinical settings. During algorithm development, preprocessing WSI datasets can be computationally demanding, particularly due to iterative DL hyperparameter optimization. We implemented a temporary patch database with a standardized file and metadata structure to optimize this process. This design facilitates efficient data re-use for multiple training runs. However, such intermediate data storage would introduce unnecessary I/O operations on the storage device during inference. To mitigate this, we included custom PyTorch dataset classes, allowing direct data integration for DL algorithms based on PyTorch. These two options ensure that PathoPatcher is suitable for development and inference workflows.

To ensure accessibility for the broader scientific community, the code has been released under a Creative Commons license. It is available as a Python package on PyPI[1].

3.3 Experimental Setup

3.3.1 Quantitative Evaluation of Preprocessing Methods for Whole-Slide Imaging

In order to evaluate the functionality of PathoPatcher compared to existing tools, we categorize the software requirements defined in Section 3.2.1 into four key areas: (1) patching capabilities, (2) pipeline properties, (3) API design, and (4) implementation. We compare our preprocessing pipeline against algorithm-specific solutions, such as those developed for the CLAM algorithm by Lu et al. [176] and DeepHistopath [69] for the TUPAC16 challenge [268], as well as standalone frameworks including PathFlowAI [159], SliDL [17], FAU-DLM [205], and TIAToolbox [222]. As many of these tools are still under development, an overview of all compared software versions is provided in the Appendix in Table A.4.

[1]Code: github.com/TIO-IKIM/PathoPatcher, installation command: `pip install pathopatch`

3.3.2 Qualitative Runtime Analysis

In addition to comparing PathoPatcher with existing frameworks in terms of functionality, we conducted experiments to measure processing throughput. Specifically, we examined the impact of parallel processes, the choice of slide loading back-end (OpenSlide, CuCIM, WSIDICOM), and the file format (TIFF-like formats versus DICOM format).

We collected an initial cohort encompassing 50 WSI scanned with an Aperio AT2 scanner (File format: `.svs`, $240\,\mathrm{px}$ internal tiling) and processed these WSIs using a standardized configuration for DL algorithms. We selected a patch size of $256\,\mathrm{px}$, an overlap of $0\,\mathrm{px}$ and extracted patches at the highest possible resolution with $0.25\,\mu\mathrm{m/px}$ (commonly referred to as $\times 40$ magnification). Based on the resulting amount of patches per WSI, we categorized them into five groups to compare if the processing times vary between tissue amount per slide: fewer than $3{,}000$ patches (tiny), $3{,}000$ - $5{,}000$ patches (small), $5{,}000$ - $20{,}000$ patches (medium), $20{,}000$ - $50{,}000$ patches (large), and more than $50{,}000$ patches (huge), with 10 WSIs per group. Since the `.svs` file format is based on the vendor-agnostic `.tiff` format, we consider it a representative example of TIFF-like file formats [158].

Additionally, we further collected and processed 10 DICOM WSIs using an Aperio GT450DX scanner (File format: `.dcm`, $256\,\mathrm{px}$ internal tiling) with the same setup, using 10 carefully selected WSI to match the patch distribution of the medium WSI group ($5{,}000$ - $20{,}000$ patches) scanned with the Aperio AT2 scanner. The GT450DX scanner is one of the first scanners that stores files directly in DICOM file format and the first scanner that the FDA approved for digital diagnosis [244]. In doing so, we want to investigate whether DICOM and its implementation in digital pathology in its current state is a suitable file format and to identify potential compatibility or performance issues.

The runtime experiments are divided into three main categories. All runtime experiments have been conducted on a single virtual machine equipped with 24 CPU cores, 32 GB RAM, an NVIDIA A100 GPU with 80 GB RAM, and 200 GB SSD storage.

Table 3.1: **Experiment 1**: Impact of parallel processing and WSI back-end.

Back-end	OpenSlide, CuCIM
Cohort	Initial Cohort
Scanner	Aperio AT2
Dataformat	svs
Conversion	x
WSI-Groups	Tiny, Small, Medium, Large, Huge
Slides per Group	10
Processes	1, 2, 4, 8, 16
Preprocessing setting	Patch size: 256, Overlap: 0, Target-MPP: 0.25

Experiment 1: Impact of Parallel Processing and WSI Back-end
First, we evaluate the performance of PathoPatcher on the `.svs` cohort, focusing on the impact of parallel processing and comparing the OpenSlide WSI back-end with CuCIM. We varied the number of processes (1, 2, 4, 8, 16) and averaged the runtime over five runs for each configuration. Due to high processing times for large WSIs ($\geq$ 20,000 patches), we restricted the experiments to 8 and 16 parallel processes. Runtime performance is reported as image throughput ($\mathrm{s}/100\,\mathrm{patches}$). The setting is defined in Table 3.1.

Experiment 2: Processing DICOM files
Second, we compare the impact of file format on processing time, comparing `.svs` against two DICOM implementations. This experiment is conducted on the medium-sized WSI. The Aperio AT2 scanned WSIs were converted to DICOM using WSIDICOMIZER. Since these DICOM files are not readable with OpenSlide, we used WSIDICOM as back-end. Also, we compared OpenSlide and CuCIM on `.svs` files with the DICOM back-end of OpenSlide, leveraging an internal DICOM cohort scanned with the Aperio GT450DX scanner. The number of processes for each configuration was increased systematically from 1 to 16 (Table 3.2a). Since the internal tiling of the Aperio DICOM files matches our standard patch size ($256\,\mathrm{px}$), we performed an additional experiment with 8 processes using a patch size of $520\,\mathrm{px}$ at a target resolution of $0.50\,\mu\mathrm{m/px}$ and $0\,\mathrm{px}$ overlap (Table 3.2b).

Experiment 3: Comparison to TIAToolBox
Finally, we compare the performance of our preprocessing pipeline with the TIAToolbox framework. We compare our performance to the original single-threaded implementation of TIAToolbox and a multi-threaded extension we integrated. As the runtime was very high for TIAToolbox, we limited the experiments to three WSIs for each patch number class (5 runs). The setting is defined in Table 3.3.

Table 3.2: **Experiment 2:** Evaluating DICOM processing throughput.

Back-end	OpenSlide, CuCIM	WSIDICOM	OpenSlide
Cohort	Initial Cohort	Initial Cohort	Additional DICOM Cohort
Scanner	Aperio AT2	Aperio AT2	Aperio GT450DX
File format	svs	dcm	dcm
Conversion	x	WSIDICOMIZER	x
WSI-Groups	Medium	Medium	Medium
Slides per Group	10	10	10
Processes	1, 2, 4, 8, 16	1,2,4,8,16	1,2,4,8,16
Preprocessing setting	Patch size: 256, Overlap: 0, Target-MPP: 0.25	Patch size: 256, Overlap: 0, Target-MPP: 0.25	Patch size: 256, Overlap: 0, Target-MPP: 0.25

(a) **Experiment 2.1**: Processing DICOM files on highest magnification with $256\,\mathrm{px}$ patch size.

Back-end	OpenSlide, CuCIM	WSIDICOM	OpenSlide
Cohort	Initial Cohort	Initial Cohort	Additional DICOM Cohort
Scanner	Aperio AT2	Aperio AT2	Aperio GT450DX
File format	svs	dcm	dcm
Conversion	x	WSIDICOMIZER	x
WSI-Groups	Medium	Medium	Medium
Slides per Group	10	10	10
Processes	8	8	8
Preprocessing setting	Patch size: 520, Overlap: 0, Target-MPP: 0.50	Patch size: 520, Overlap: 0, Target-MPP: 0.50	Patch size: 520, Overlap: 0, Target-MPP: 0.50

(b) **Experiment 2.2**: Processing DICOM files on $0.50\,\mu\mathrm{m/px}$ with $520\,\mathrm{px}$.

Table 3.3: Experiment 3: Comparison of PathoPatcher to TIAToolBox.

Toolbox	PathoPatcher(OpenSlide, CuCIM), TIAToolbox
Cohort	Initial Cohort
Scanner	Aperio AT2
Dataformat	svs
Conversion	x
WSI-Groups	Tiny, Small, Medium, Large, Huge
Slides per Group	3
Processes	8
Preprocessing setting	Patch size: 256, Overlap: 0, Target-MPP: 0.25

3.4 Results and Analysis

3.4.1 Functionality

A comparative analysis of the leading frameworks for WSI preprocessing is provided in Table 3.4. In particular, PathoPatcher and PathAI are the only two frameworks that meet all patching requirements. None of the remaining frameworks supports multi-resolution extraction, which is particularly relevant for multi-scale models. These models incorporate images at multiple spatial resolutions, which is particularly useful in CPATH image segmentation. Most frameworks, except PathAI, SliDL, and PathoPatch, lack support for loading annotations. All frameworks include at least OpenSlide as the back-end, with TIAToolbox offering the most file format flexibility as it includes multiple loading back-ends. Marker removal is supported only by Deep-Histopath. Our marker removal method combines color filtering and postprocessing, but we cannot ensure consistent removal for all colors (e.g. red marker cannot be consistently distinguished from tissue by thresholding).

In summary, PathFlowAI is similar to PathoPatcher, with a focus on direct model integration. PathoPatcher, on the other hand, is meant to be used as a standalone solution. Histolab is a versatile and extensible Python framework, but it is important to recognize that it was not considered a standalone solution in our evaluation. Histolab offers a range of features that can be integrated into specific workflows. However, for our specific use case and the need for a standalone solution, we found that TIAToolbox is the only competitive framework that can automatically select the appropriate loading back-end and includes essential features.

3.4.2 Runtime

Figure 3.3a illustrates the results of the comparative analysis carried out in Experiment 1 between OpenSlide and CuCIM within PathoPatcher. In fact, the CuCIM implementation exhibited a notable increase in speed compared to OpenSlide. With regard to the number of processes employed, the optimal number appears to be eight. Although the runtime was higher and exhibited a higher variance with fewer processes, a nearly constant throughput was achieved with eight processes. Further doubling the number of parallel processes did not increase the runtime. Com-

Table 3.4: Comparison of preprocessing frameworks. Requirements are separated into image options for selecting patching parameters, pipeline components, interfaces to interact with the framework (API), and implementation (Impl.) properties. Adapted from [117]. Abbr.: OS (OpenSlide).

	Requirements	CLAM	Deep-Histopath*	PathFlowAI	FAU-DLM	SliDL*	TIAToolbox	PathoPatcher
Patching	Select Patch Size	✓	✓	✓	✓	✓	✓	✓
	Select Patch Overlap	✓	✓	✓	-	✓	✓	✓
	Select Resolution	✓	✗	✓	✓	✓	✓	✓
	Multi-Resolution Extraction	✗	✗	✓	✗	✗	✗	✓
Pipeline	Tissue Detection	✓	✓	✓	✓	✓	✓	✓
	Stain Normalization	✗	✓	✓	✗	✗	✓	✓
	Load Annotations	✗	✗	✓	Just ROI	✓	✗	✓
	Exclude Segmentation Classes	✗	✗	✓	✗	✓	✗	✓
	Marker Removal	✗	✓	✗	✗	✗	✗	Annotation/DL
API	CLI	✓	✗	✓	✗	✗	✓	✓
	File-Selection	✗	✗	✗	✓	✗	✓	✓
	ML-Framework Integration	PyTorch	TensorFlow	PyTorch	PyTorch	PyTorch	PyTorch	PyTorch
Impl.	Multi-Processing	✗	✓	✓	✓	✓	✗	✓
	Back-end	OS	OS	OS	OS	pyvips	OS/DICOM Tifffile/Own	OS/CuCIM WSIDICOM

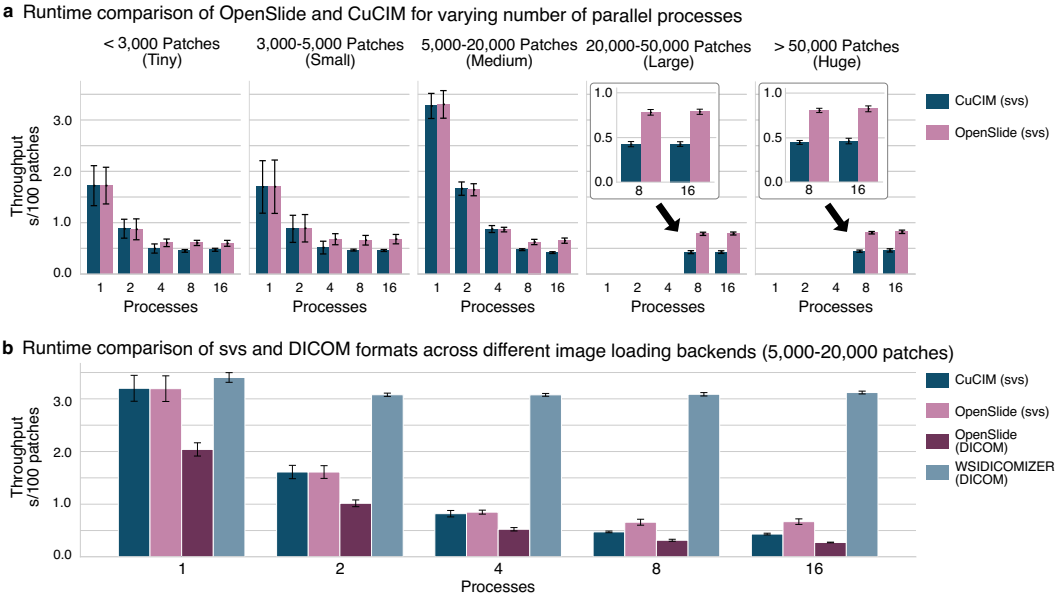


Figure 3.3: Runtime comparison of WSI loading back-ends.
The plot shows the average processing throughput in $s/100\,\text{patches}$, with error bars representing the standard deviation across five independent experiments. **a**) Comparison of OpenSlide and CuCIM for different numbers of parallel processes, grouped by patch count. **b**) Performance comparison between the proprietary Leica .svs file format and the DICOM format, evaluated using two DICOM implementations (Leica vs. IMIBigPicture WSIDICOMIZER). This experiment was conducted on 10 WSIs from the medium group (5,000 - 20,000 patches). Adapted from [117].

paratively, increasing the number of processes from one to eight with OpenSlide back-end resulted in a runtime decrease of $69.08 \pm 10.57\%$ on average, which was further reduced by approximately $26.25 \pm 2.62\%$ on average using CuCIM. The runtime analysis of DICOM files compared to TIFF-like files, as shown in Figure 3.3b, yielded ambiguous results.

Next to runtimes for the two DICOM implementations in combination with the OpenSlide and WSIDICOM back-end, we included the runtime of OpenSlide and CuCIM using the original `.svs` files for reference. As expected, the processing time per 100 patches decreased with an increasing process number. However, when using WSIDICOMIZER, the runtime remained consistently high regardless of the number of parallel processes. In contrast, processing DICOM files (acquired directly by the Aperio Scanner) with OpenSlide reduced runtime. Based on the results and in line with the results in Figure 3.3a, eight parallel processes emerged as the best trade-off between throughput and system load. With this configuration, the processing time per 100 patches decreased from $0.474 \pm 0.015\,\mathrm{s}/100\,\mathrm{patches}$ seconds (CuCIM) by $33.49 \pm 3.71\%$ to $0.315 \pm 0.019\,\mathrm{s}/100\,\mathrm{patches}$ by the combination of DICOM and OpenSlide. Conversely, when WSIDICOM is used with converted DICOM files, the processing time increased by $551.49 \pm 20.71\%$ compared to CuCIM, with a runtime of $3.087 \pm 0.032\,\mathrm{s}/100\,\mathrm{patches}$. To investigate whether the speedup of DICOM file processing was caused by the internal tiling matching the acquisition patch size of $256\,\mathrm{px}$, we conducted an additional experiment with a patch size of $520\,\mathrm{px}$ pixels. The results with this setting verified the superior performance of the OpenSlide DICOM back-end over CuCIM (Table A.2 in the Appendix).

The last experiment compares the runtime of PathoPatcher with the TIAToolbox framework (Figure 3.4). PathoPatcher demonstrated a significant increase in runtime efficiency and was faster than TIAToolbox in all the configurations independent of the back-end. This speed up was further enhanced with the increasing number of extracted patches. When compared to the previously published single process implementation of TIAToolbox, PathoPatcher achieved a runtime reduction of $78.36 \pm 6.63\%$. With our multiprocessing adaption (eight processes), we still achieved a runtime reduction of $45.60 \pm 8.00\%$ when using CuCIM and $34.48 \pm 10.13\%$ when using OpenSlide.

3.5 Chapter Conclusion

Developing algorithms for digital pathology and integrating them into clinical routine requires a robust and fast preprocessing. In this Chapter, we presented PathoPatcher, a preprocessing framework specifically for WSI manipulation. Our pipeline is constructed to meet the requirements for both classification and segmentation algorithms and enables fast inference runtimes. Unlike existing frameworks, our solution provides a broad range of functionality. Many competitive solutions

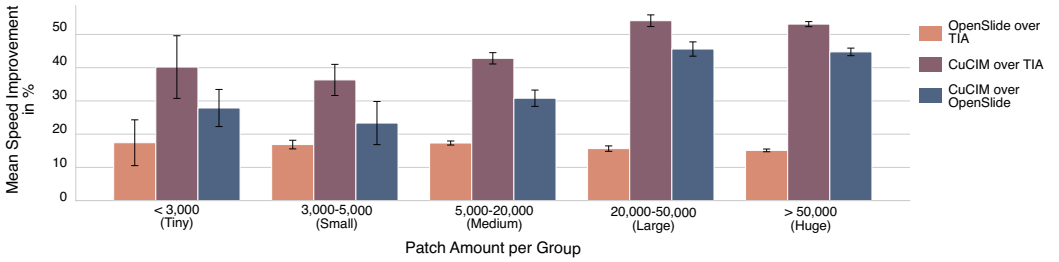


Figure 3.4: Comparison of PathoPatcher (using OpenSlide and CuCIM back-ends) and TIAToolbox (TIA) using eight parallel processes, grouped by patch quantity. The plot shows the average percentage speed improvement, with error bars representing the standard deviation across five independent experiments. Adapted from [117].

are centered around specific algorithms, such as CLAM or DeepHistopath, or are implemented as Python libraries to be integrated into existing codebases, e.g., PathFlowAI, histolab, without providing end-to-end solutions. Among the existing frameworks, TIAToolbox is the only one that fulfills our requirements. Nonetheless, our experiments demonstrate that PathoPatcher reduced the processing time by 78.36 ± 6.63 %. We benchmarked three slide-loading back-ends and highlighted the impact of file format and caching strategy on processing speed. Interestingly, although the CuCIM library yielded the best performance for TIFF-like formats, Leica's DICOM implementation in combination with OpenSlide outperformed even the CuCIM back-end. Based on the experimental results, CuCIM remains the preferred choice for TIFF-like formats, while Leica's DICOM implementation is recommended for DICOM data. Converting TIFF-like formats to DICOM with WSIDICOMIZER and the WSIDICOM back-end significantly slows down processing. Consequently, such a conversion is only recommended if harmonization of the data into DICOM is necessary and processing speed is secondary.

Although our work is an important contribution to the clinical use of CPath, some limitations remain. The experiments are restricted to two file formats, limiting the generalizability of our findings to TIFF-Like formats and DICOM. Additionally, as our experiments on the converted DICOM images reveal, there is a considerable performance gap between proprietary DICOM implementations (Leica) and open-source conversions (WSIDICOM), which requires further investigation. Thus, vendor solutions are still necessary to implement the DICOM standard in clinical workflows. Even though we achieved substantial reductions in processing time, real-time performance has not yet been achieved. Moreover, the quality control (QC) mechanism is based solely on filtering techniques. Future improvements should expand and validate these QC features as well. A feedback mechanism for pathologists and laboratory staff could also be improved, for example by generating a detailed QC report per slide. This further would improve translation into clinical systems.

In summary, the preprocessing pipeline introduced in this Chapter lays the processing groundwork for the algorithms presented in the following Chapters. It overcomes key limitations of existing solutions and improves runtime by using multiple parallel processes to access tiles. Our main contributions include:

Contribution 1: We provide a fast and flexible preprocessing pipeline that is suitable for a variety of computational pathology tasks, such as training of classification and segmentation algorithms, as well as inference pipelines. Through the use of multiprocessing, we decreased the processing time by up to approx. 78%.

Contribution 2: By including multiple image loading backe-ends such as OpenSlide, CuCIM and WSIDICOMIZER, we support a wide range of vendor-specific and vendor-agnostic WSI formats. Our tool is able to deal with most of the available WSI data formats.

Contribution 3: We designed the pipeline as a Python package to enable seamless integration into existing Python-based projects and ensure ease of use. We provide a CLI that can be used directly without the need for coding expertise.

4

Whole Tissue Segmentation

In pathology, the examination of tissue sections under a microscope involves interaction with the slide, as pathologists navigate through the slide by panning and zooming in and out. They access structures not only by their local morphology but also by their spatial relations. Previous segmentation algorithms for identifying macroscopic tissue structures did not sufficiently take this into account. Instead, segmentation was performed on small image patches with a limited field of view and the results stitched together. This patch-wise approach is rather prone to segmentation artifacts and incorrect tissue class assignments. To this end, we introduce a method that incorporates context tissue information into the local patch segmentation by mimicking how pathologists consider spatial relationships. Our experiments on three datasets highlighted that this method is effective in mitigating segmentation artifacts and reducing false positive tumor segmentations.

The content of the Chapter is based on our publication in Computerized Medical Imaging and Graphics [76]. I co-authored the work and contributed significantly to the final published version (total contribution 30%). The initial concept of the Memory Attention Framework (Method Section), along with most of the software implementation, was developed by Oliver Ester, with the mathematical concept outlined by me. These Sections are included in this thesis to improve the understanding of the method. The experimental design for the RCC and CY16 dataset has been jointly developed by Oliver Ester and myself (30% contribution). I was entirely

responsible for conducting the experiments on the Paip 2019 dataset. I conducted all subsequent evaluations using an internal pancreatic cancer dataset, including the inference pipeline implementation and clinical assessment, and supervised the annotation procedure. Detailed contributions to the work are outlined in the Personal Contribution Statement and the Transparency Statement in the Appendix.

4.1 Context and Objectives

How can we ensure that the information from surrounding tissue is taken into account when segmenting small patches of a WSI such that the global information about the tissue composition can be integrated into the local segmentation?

Remembering the central question raised in the context of macro-level tissue analysis (Section 2.3)? Pathologists, when identifying a specific tissue type, typically examine a slide section at high magnifications and incorporate contextual information from neighboring slide regions. Some works have been published trying to imitate this behavior but with limited success [100, 162, 240, 262, 265]. Either the performance increase was not sufficient [100, 162], or the method was computationally complex and hard to adapt [240, 262, 265].

In contrast to related works, we propose a new way of incorporating contextual tissue information. Instead of relying on multiple image encoders [100, 162, 240, 262, 265], our method makes use of the attention mechanism [266]. Basically, we transform a patched version of the WSI into an embedding space and query over this space to retain context information. The attention mechanism helps the network to adaptively determine the importance of each surrounding tissue patch to incorporate this information during the segmentation of a central patch. This procedure offers two significant advantages. First, it requires substantially fewer parameters compared to a network with multiple encoders. Second, it can be integrated as an optional extension to existing encoder-decoder architectures. We call this approach "**M**emory **A**ttention **F**ramework" (**MAF**).

To demonstrate the applicability of our method, we evaluate the MAF on two public datasets (Camelyon16 (CY16) for breast cancer and PAIP 2019 for liver cancer) and an internal renal cell carcinoma (RCC) dataset using conventional segmentation models like U-Net and DeepLabV3 in combination with the MAF. Our results demonstrate the superiority of MAF over baseline architectures and competitive context integration algorithms with multiple encoders. In addition to the experimental evaluation, we demonstrate the downstream application of the segmentation network using an additional pancreatic cancer cohort, on which we perform survival analysis using tissue segmentation results.

4.2 Methods

4.2.1 Preliminary Mathematical Definitions

For the mathematical formulation, it is assumed that a WSI comprises a single plane and does not have a pyramid-like structure. Let

$$\boldsymbol{W} \in \mathbb{R}^{H_{\text{wsi}} \times W_{\text{wsi}} \times 3}$$

be a WSI with height H_{wsi} and width W_{wsi}. The WSI is divided into non-overlapping patches $\mathcal{P} = \{\boldsymbol{P}_{r,c}\}$, $\boldsymbol{P}_{r,c} \in \mathbb{R}^{H_{\text{p}} \times W_{\text{p}} \times 3}$ with height H_{p} and width W_{p}. In this definition, $r \in \mathbb{Z}$ is the row and $c \in \mathbb{Z}$ column position in a uniform, two-dimensional spatial grid with the dimensions N_{r} and N_{c}. For simplicity, we assume that the patches are squared, i.e. $H_{\text{p}} = W_{\text{p}} = S_{\text{p}}$. The objective of image segmentation is to predict a segmentation mask $\hat{\boldsymbol{Y}}_{r,c} \in \{0,1\}^{S_{\text{p}} \times S_{\text{p}} \times C}$ with C classes for each corresponding patch $\boldsymbol{P}_{r,c}$, with $\boldsymbol{Y}_{r,c} \in \{0,1\}^{S_{\text{p}} \times S_{\text{p}} \times C}$ denoting the ground truth.

4.2.2 Network Architecture

Our starting point for the design of a CPath tissue segmentation model is a standard encoder-decoder segmentation architecture, such as the U-Net and its variants [107, 234, 302], or the DeepLab series [41, 42, 43]. The decoder f_{enc} maps an input patch $\boldsymbol{P}_{r,c}$ subsequently to a set of feature maps $\mathcal{Z} = \{\boldsymbol{Z}_0, \cdots, \boldsymbol{Z}_L\}$, where L denotes the number of depth layers. Given this, the decoder f_{dec} up-samples the feature maps and predicts the segmentation mask $\hat{\boldsymbol{Y}}_{r,c}$. However, this approach does not consider

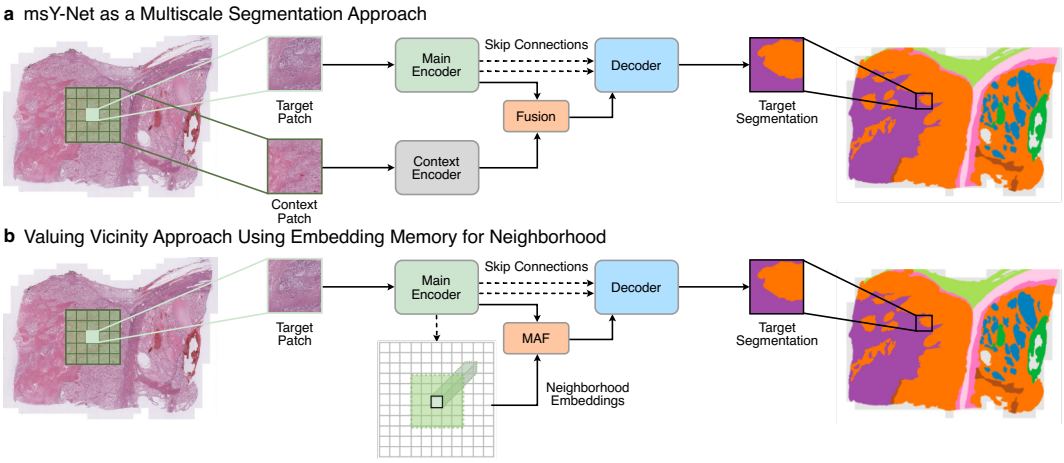


Figure 4.1: Comparison between context integration methods for image segmentation. **a**) msY-Net with two encoder as an example for a context integration method. **b**) Our proposed MAF integrates the context information using only one encoder.

the spatial relationship between the patches of a WSI, resulting in segmentation artifacts and wrong class assignments due to context information missing. In other words, patch segmentation is solely based on local information and does not incorporate tissue context.

Most solutions for integrating tissue context are based on a multiscale approach where additional context patches with a larger FOV are integrated into the network. Examples include the msY-Net by Schmitz et al. [240] and the HookNet by Van Rijthoven et al. [265]. However, these approaches have some disadvantages. The most promising approaches utilize one encoder for each input patch (including the context), resulting in networks with substantially more parameters and increased computational complexity (Figure 4.1a). Additionally, these methods are challenging to adapt to new network architectures, limiting the translation of research findings from the field of CV to CPath.

Due to the disadvantages mentioned above, we use an attention-based approach to consider the tissue context, which we call the Memory Attention Framework (MAF). This framework builds upon an arbitrary encoder-decoder architecture, extending it by an attention module at the bottleneck layer L (transition from encoder to decoder network), as shown in Figure 4.1b. During the segmentation of a patch, the information of the neighboring patches is taken into account by calculation and querying an embedding memory. Two concepts are important:

1. Embedding memory
2. Neighborhood attention

Our method can be integrated on top of existing encoder-decoder architectures. A high-level comparison between the msY-Net and the MAF is shown in Figure 4.1, illustrating the difference between feeding a context patch into an additional encoder versus using an embedding memory.

Embedding Memory The tissue context is incorporated through a fusion mechanism in the bottleneck layer L. To achieve this, a compressed representation of the patches $\mathcal{P}$ is computed by mapping each patch $\boldsymbol{P}_{r,c}$ to an embedding vector using the encoder f_{enc} and the linear transformation f_{emb}. The set of embeddings for a WSI is then defined as the memory

$$\mathcal{M} = \{\boldsymbol{e}_{r,c} \mid r \in [1-k, N_{\text{r}}+k],\ c \in [1-k, N_{\text{c}}+k]\}\,, \quad \boldsymbol{e}_{r,c} \in \mathbb{R}^{D_{\text{MAF}}}. \tag{4.1}$$

The memory is expanded in each spatial direction by the neighborhood size k. For a visual explanation, see Figure 4.2. At the beginning of each training epoch, the memory is calculated by a forward pass through the encoder without gradient computation. Due to the high computational cost, an online memory update is not performed. If a patch is absent in the 2D grid, e.g., excluded background patches, no embedding is calculated, and the memory entry is filled with zeros and excluded for subsequent computations.

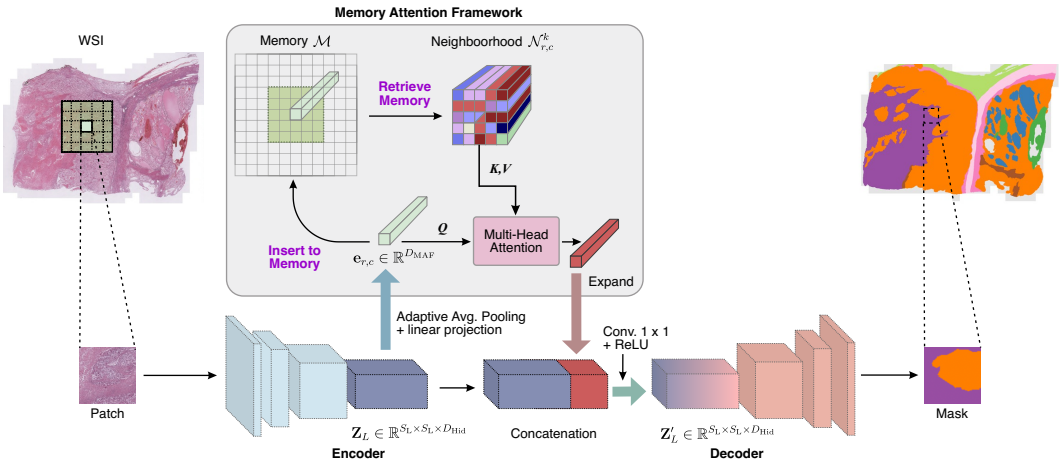


Figure 4.2: Overview of the memory attention framework within an encoder-decoder architecture.
The context information is integrated into the bottleneck layer L of the encoder-decoder network. First, each patch needs to be embedded to build up $\mathcal{M}$. Then the neighborhood attention mechanism integrates the context information of the neighboring patches by calculating a weighted mean of each embedding in the neighborhood $\mathcal{N}$. Adapted from [76].
Abbr.: Conv. (Convolution)

Neighborhood Attention Integrating the context for a patch with the coordinates r, c is achieved by utilizing a rectangular neighborhood

$$\begin{aligned} \mathcal{N}^k_{r,c} &= \{\boldsymbol{e}_{r',c'} \mid r' \in [r-k, \ldots, r+k],\ c' \in [c-k, \ldots, c+k],\ r' \neq r,\ c' \neq c\}\,, \\ \mathcal{N}^k_{r,c} &\subseteq \mathcal{M} \end{aligned} \tag{4.2}$$

with the radius k. The integration of the neighborhood $\mathcal{N}$ is performed using the attention mechanism, already defined in eq. (2.4). This approach enables the network to determine to what extent the information from neighboring patches should be incorporated. For each $\boldsymbol{e}_{r',c'} \in \mathcal{N}^k_{r,c}$, the attention mechanisms calculates a normalized weight

$$a_{r',c'} \geq 0, \quad \left(\sum_{r'=r-k, r' \neq r}^{r+k} \sum_{c'=c-k, c' \neq c}^{c+k} a_{r',c'} \right) = 1. \tag{4.3}$$

Importantly, the attention mechanisms contain learnable parameters, such that it is guided to learn tissue interactions. The weighted sum of the embeddings

$$\sum_{r'=r-k, r' \neq r}^{r+k} \sum_{c'=c-k, c' \neq c}^{c+k} a_{r',c'} \boldsymbol{e}_{r',c'} \tag{4.4}$$

is expanded to match the first two dimensions $S_{\mathrm{L}} \times S_{\mathrm{L}}$ of the feature map $\boldsymbol{Z}_L$, subsequently concatenated to $\boldsymbol{Z}_L$, and, finally, transformed back into the dimension $S_{\mathrm{L}} \times S_{\mathrm{L}} \times D_{\mathrm{Hid}}$. Therefore, the method can be incorporated as an additional module into any encoder-decoder structure without alternating the network architecture.

4.3 Experimental Setup

We evaluated the MAF using three histopathological datasets: an internal RCC dataset and the publicly available CY16 and Paip 2019 datasets. The MAF was integrated into two baseline architectures, U-Net and DeepLabV3. For comparison with context-based approaches, we benchmarked against msY-Net by Schmitz et al. [240].

4.3.1 Baseline and Methods

Determining the Optimal MAF Setup The first task is to determine the proper setting for the MAF. Starting from a baseline setup without MAF, various configurations have been tested. For this purpose, the two most important parameters of the MAF, namely the embedding dimension D_{MAF} and the neighborhood radius k have been varied. In addition, we investigated whether a 2D positional encoding instead of the standard 1D encoding [67, 266] and an additional classifier loss [189] (prediction of the class distribution of a patch) improved the performance of the MAF. The baseline setup consisted of a DeepLabV3 network with a ResNet50 backbone, pretrained on the ImageNet dataset. This evaluation was performed on our internal RCC dataset.

Comparison with Baselines and Context-based Approaches Based on the best configuration for the MAF, we performed further experiments using the CY16 and the Paip 2019 dataset. We assessed how the U-Net and DeepLabV3 models' performance is affected by the MAF integration and compared the performance to the msY-Net by Schmitz et al. [240]. For a fair comparison with the msY-Net, a ResNet18 model (ImageNet pretrained [59]) was used for these experiments.

4.3.2 Evaluation Datasets

Renal Cell Carcinoma Dataset This internal dataset was created in collaboration with the MHH Hannover. It consists of 175 fully annotated WSI of patients with metastatic renal cell carcinoma. The WSIs were acquired at a resolution of $0.1729\,\mu\mathrm{m/px}$. A board-certified pathologist created expert-level annotations for seven tumor categories (tumor vitality, tumor regression, tumor necrosis, tumor bleeding, angioinvasion, capsule, and cyst) along with three non-tumor categories (extrarenal, cortex, and mark).

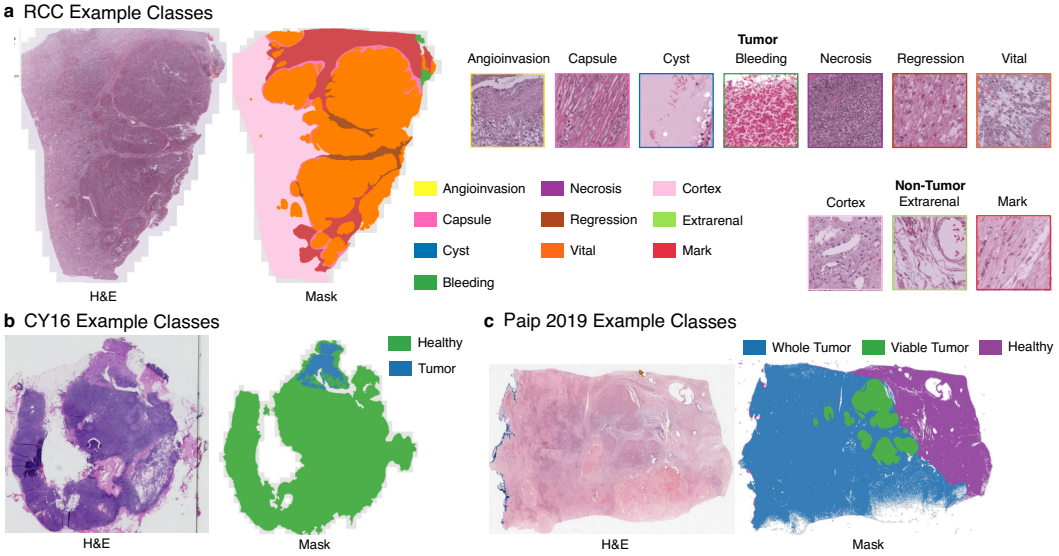


Figure 4.3: Example WSI from each tissue dataset used for evaluating the MAF, accompanied by segmentation masks.

Camelyon 16 As the images of the RCC dataset cannot be shared, the CY16 dataset was employed as a public benchmark dataset. This dataset was introduced as part of the Camelyon16 challenge [71] and contains 270 WSI ($0.243\,\mu m/px$) of sentinel lymph nodes from breast cancer patients with or without metastases [71]. Following the experimental setup of Schmitz et al. [240], we used the identical 20 WSI containing at least one macroscopic metastasis (diameter $\geq 2\,mm$) [71, 240]. Metastatic tissue and healthy tissue annotations are provided by the authors.

Paip 2019 In addition to the CY16 dataset, we also used the Paip 2019 dataset as a public benchmark dataset. This dataset consists of 50 public WSI[1] ($0.502\,\mu m/px$) of liver cancer patients with annotations for healthy tissue, vital tumor, and non-vital tumor.

Example WSIs from all datasets and segmentation masks manually annotated by experts are presented in Figure 4.3.

Dataset Preprocessing For preprocessing, we used the PathoPatcher framework presented in Chapter 3. A patch size of $S_P = 256$ without overlap was employed for all datasets. Additionally, msY-Net requires squared context patches $S_C = 256$ extracted at a resolution four times lower than local patches. To align color spaces, we have used Macenko normalization.

[1]The 40 test WSI are not publicly available

Table 4.1: Overview of the dataset splits used for training, validation and testing. We conducted 5-fold cross-validation on the WSI level. The amounts given are the respective WSI count per fold.

Dataset	Train	Validation	Test
RCC	112	28	35
CY16	12	4	4
Paip 2019	32	8	10

4.3.3 Training

The baseline segmentation architectures compared are the U-Net and DeepLabV3 models, in which we incorporated ResNet50 and ResNet18 encoders (pretrained on ImageNet) for both architectures. To deal with significant class imbalances in the datasets, only 100 patches per WSI were sampled for training. Color jitter was the only data augmentation performed, as spatial transformations may affect the MAF's applicability. The training was conducted for 100 epochs with a batch size of 32 using the SGD optimizer with early stopping of 10 epochs. A detailed training configuration along with software versions is provided in Tables A.8 and A.9 in the Appendix.

4.3.4 Evaluation Metrics and Strategy

The performance is measured using the DICE score. Consider a binary segmentation task with predictions $\hat{\boldsymbol{Y}} \in \{0,1\}^{H \times W}$ and ground truth labels $\boldsymbol{Y} \in \{0,1\}^{H \times W}$. Then the $DICE$-score is defined as

$$DICE(\hat{\boldsymbol{Y}}, \boldsymbol{Y}) = \frac{2 \cdot \left| \hat{\boldsymbol{Y}} \cap \boldsymbol{Y} \right|}{\left| \hat{\boldsymbol{Y}} \right| + \left| \boldsymbol{Y} \right|}. \tag{4.5}$$

This can be interpreted as twice the overlap between the two masks, normalized by the total number of pixels across both masks. If the task is a multi-class segmentation task with C classes, the $DICE$-score is calculated for each class independently, denoted as $DICE_c$, and the mean value over all classes is denoted by $DICE_{\text{Total}}$.

To assess the performance, we performed a 5-fold cross-validation for each experiment and dataset. To avoid data leakage, 20% of each training fold data is used for epoch-wise validation to enable early stopping (Table 4.1). We report the average $\overline{DICE}$-score along with standard deviation across the test results of all folds. Importantly, the metric was first calculated across all patches of a WSI before averaging.

4.4 Results and Analysis

4.4.1 Preliminary Input Resolution Evaluation

Before determining the best MAF setting (neighborhood radius k and embedding dimension D_{MAF}), we performed preliminary experiments to determine the input resolution (μm/px) for each dataset. We varied the input resolution of the patches from 1.383 μm/px to 11.066 μm/px for the RCC dataset, from 0.243 μm/px to 3.888 μm/px for the CY16 dataset and from 0.502 μm/px to 8.048 μm/px for the Paip 2019 dataset (reducing by a factor of 2 each experiment). We then trained a DeepLabV3 network and a U-Net network for each resolution. The following resolutions achieved the best segmentation performance measured by the $\overline{DICE}$-score:

- RCC: 2.77 μm/px (16-fold downsampling of the base-resolution)
- CY16: 0.49 μm/px (2-fold downsampling of the base-resolution)
- Paip 2019: 1.00 μm/px (2-fold downsampling of the base-resolution)

The experimental results are provided in the Tables A.5-A.7 in the Appendix.

4.4.2 Determining the MAF Setup

To determine the MAF configuration, we first varied D_{MAF} (using k=8) and evaluated the performance on the RCC dataset, with results reported in Table 4.2. The $\overline{DICE}_{\text{Total}}$-scores indicate that the best performance was achieved with $D_{\text{MAF}} = 1024$, reaching $\overline{DICE}_{\text{Total}} = 0.547 \pm 0.019$.

Architecture	MAF	D_{MAF}	Adaption	$\overline{DICE}_{\text{Total}}$ ± SD
DeepLabV3	✗			0.502 ± 0.011
	✓	512		0.532 ± 0.016
		1024		0.547 ± 0.019
		2048		0.540 ± 0.012
		1024	2D pos. enc.	0.552 ± 0.017
		1024	2D pos. enc. + $\mathcal{L}_{\text{cls}}$	0.573 ± 0.013
U-Net	✗			0.481 ± 0.013
	✓	1024	2D pos. enc. + $\mathcal{L}_{\text{cls}}$	0.500 ± 0.015

Table 4.2: Variations of the MAF setup on the RCC dataset, using a ResNet50 backbone, varying D_{MAF}. As adaptions, we added 2D positional encoding and the classification loss $\mathcal{L}_{\text{cls}}$. Reported are the averaged 5-Fold results. Neighborhood radius fixed at k=8.

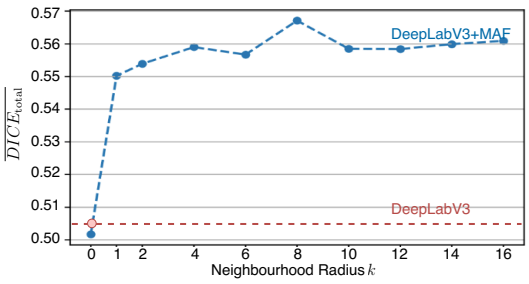


Figure 4.4: Influence of the neighborhood size k on the segmentation performance on the RCC dataset (average of 5 folds)

Table 4.3: Context integration contribution of MAF compared with msY-Net and baseline U-Net and DeepLabV3 (all based on ResNet-18). Difference Δ refers to the corresponding baseline structure without MAF on each dataset. Second best underlined. Adapted from [76].

Dataset	RCC		CY16		PAIP 2019	
Score	$\overline{DICE}_{\text{Total}} \pm \text{SD}$	Δ	$\overline{DICE}_{\text{Tumor}} \pm \text{SD}$	Δ	$\overline{DICE}_{\text{Total}} \pm \text{SD}$	Δ
U-Net	0.45 ± 0.02		0.73 ± 0.06		0.70 ± 0.01	
U-Net + MAF	0.47 ± 0.01	+0.02	0.75 ± 0.08	+0.02	$\underline{0.73 \pm 0.01}$	+0.03
DeepLabV3	$\underline{0.49 \pm 0.02}$		0.76 ± 0.07		0.73 ± 0.02	
DeepLabV3 + MAF	**0.54 ± 0.01**	**+0.05**	**0.80 ± 0.09**	**+0.04**	**0.77 ± 0.01**	**+0.04**
msY-Net	0.46 ± 0.01		$\underline{0.79 \pm 0.08}$		0.73 ± 0.02	

Subsequently, we replaced the 1D positional encoding by 2D positional encoding, as we hypothesize that this more effectively aligns with our concentric neighborhood definition. This adjustment led to improved performance, with $\overline{DICE}_{\text{Total}} = 0.552 \pm 0.017$. As a last refinement, we followed the work of Mehta et al. [189], and introduced an additional classification loss $\mathcal{L}_{\text{cls}}$ using an auxiliary task. Given the neighborhood embeddings $\mathcal{N}^k_{r,c}$, the class distribution within the central patch $\boldsymbol{P}_{r,c}$ should be predicted and the cross-entropy loss is calculated. We then adjusted the training loss to incorporate the classification loss ($\mathcal{L} = 0.8 \cdot \mathcal{L}_{\text{seg}} + 0.2 \cdot \mathcal{L}_{\text{cls}}$). This further enhanced performance to $\overline{DICE}_{\text{Total}} = 0.573 \pm 0.013$.

Using this setting, we explored the effect of the neighborhood size k by varying the size between $k = 1$ and $k = 16$. A graph visualizing the average performance is illustrated in Figure 4.4. Performance improved as the size of the neighborhood increased to $k = 8$, followed by a saturation for larger neighborhood sizes. Therefore, we chose $k = 8$ for all subsequent experiments.

4.4.3 Comparison with Baselines and Context-based Approaches

The comparison between MAF and msY-Net is presented in Table 4.3. For the MAF, we used the setting derived on the RCC dataset, using $k = 8$ and $D_{\text{MAF}} = 1024$. Compared to the baseline DeepLabV3 and U-Net, the integration of spatial context through the MAF consistently improved segmentation performance. Among these, DeepLabV3 with MAF achieved superior results on all three datasets ($+0.05$ For RCC, $+0.04$ CY16, $+0.04$ Paip 2019). In contrast, msY-Net surpassed the baseline U-Net, but not DeepLabV3, and achieved lower results compared to the DeepLabV3+MAF architecture.

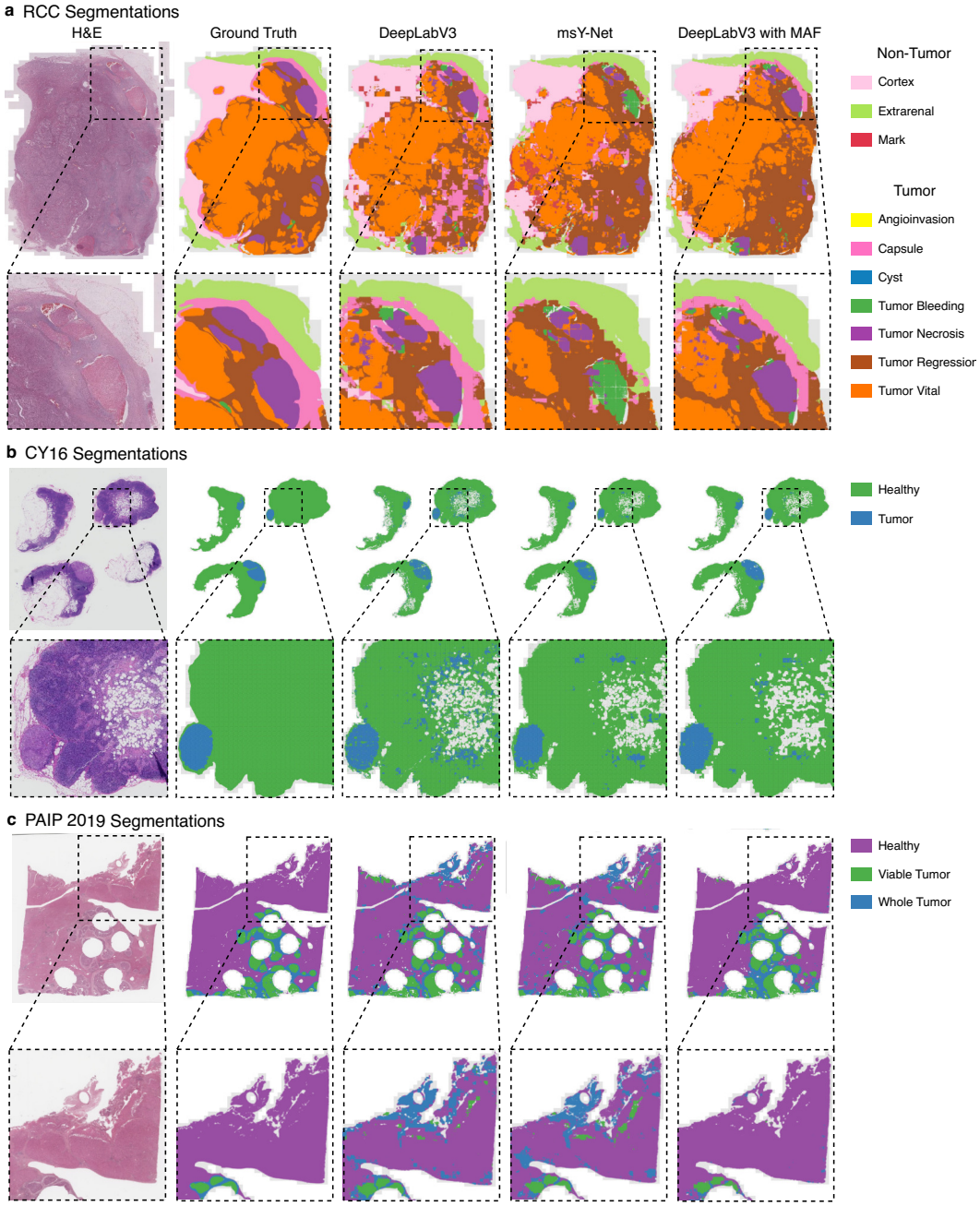


Figure 4.5: Qualitative comparison between the baseline DeepLabV3 and the context integration approaches msY-Net and DeepLabV3 with MAF.
a) Comparison on the RCC dataset, using a ResNet18 backbone for msY-Net and a ResNet50 backbone for DeepLabV3 and MAF. **b**) Comparison on the CY16 dataset with identical network settings as in (a). **c**) Comparison on the Paip 2019 dataset, but with a ResNet18 backbone for all networks. Adapted from [76].

4.4.4 Qualitative Results

For a deeper understanding of the effect of the MAF, we visualized the segmentation results on the RCC, CY16, and Paip 2019 dataset in Figure 4.5. Depicted are ground truth segmentation masks along with segmentation results obtained by a DeepLabV3 network without and with MAF (k=8), as well as the results from the msY-Net network. In general, both the MAF and msY-Net approaches effectively reduce the number of false-positive tumor regions. Without context integration, isolated patches can be completely misclassified as an incorrect tissue class, visually apparent as artifacts in the segmentation masks. Both context approaches mitigate these artifacts due to their spatial context information. However, this improvement is more pronounced for the MAF approach, clearly visible in the example CY16 and Paip 2019 images.

4.5 Clinical Application

Moving beyond algorithm optimization, we explored the clinical relevance of segmentation networks using the Selocan pancreatic cancer dataset and the corresponding TCGA-PAAD (The Cancer Genome Atlas) cohort.

Based on the results of the previous experiments, we propose the following workflow for training new models, exemplified using the pancreatic cancer cohorts:

1. Determine the appropriate resolution: Depending on the complexity of the task and the properties of the tissue, the optimal resolution for the segmentation network varies between $0.25\,\mu\mathrm{m/px}$ and $2.00\,\mu\mathrm{m/px}$. To determine the optimal input resolution, first train a baseline DeepLabV3 model without MAF for each resolution. We recommend using $S_\mathrm{p} = 256\,\mathrm{px}$ or $S_\mathrm{p} = 512\,\mathrm{px}$ as patch size.
2. Following this procedure, train the model with the MAF ($D_\mathrm{MAF} = 1024$, $k = 8$). Evaluate whether the MAF overhead results in a performance improvement. If the improvement is only marginal, consider removing it to increase runtime efficiency.

Dataset The Selocan cohort consists of patients with stage I-IV pancreatic cancer (400 samples). Initial research based on this cohort explored the analysis of tumors as multicellular ecosystems and their relationship with the extracellular matrix, connective tissue, and vascularization (tumor microenvironment, TME) [99]. This study demonstrated that heterogeneity in pancreatic cancer is a function of regional tissue properties within the TME [99]. Based on the observations, the three functional stromal subtypes (subTMEs) reactive, intermediate, and deserted TME were identified [99]. However, because processing each sample takes a considerable amount of time, it has not been possible to thoroughly extract these subtypes

Table 4.4: Segmentation algorithm performance comparison for Selocan cohort. Presented are the class-wise Dice ($DICE_c$) scores for the fully annotated test set. The highest score for each tissue type is shown in bold, and the second-highest is underlined. We used the DeepLabV3 network with ResNet50 backbone for all experiments.

Resolution	Model	MAF	Tumor	Deserted	Intermediate	Reactive	Adipose	Other	Background	Total
1.00 µm/px (×10)	RN50-ImageNet	✗	0.44	0.53	0.54	0.13	0.20	0.09	**0.92**	0.41
	RN50-Lunit	✗	0.56	0.51	0.52	0.11	0.18	0.14	<u>0.90</u>	0.49
	RN50-CCL	✗	0.66	0.52	0.59	0.13	<u>0.28</u>	0.28	**0.92**	0.43
0.50 µm/px (×20)	RN50-ImageNet	✗	0.61	<u>0.56</u>	0.58	<u>0.22</u>	0.26	0.19	0.88	0.47
	RN50-Lunit	✗	0.68	**0.57**	**0.64**	**0.23**	0.27	0.25	0.85	0.50
	RN50-CCL	✗	0.66	**0.57**	0.61	0.21	0.18	0.21	0.89	0.47
0.25 µm/px (×40)	RN50-ImageNet	✗	0.73	0.52	0.60	0.16	0.22	0.34	0.84	0.49
	RN50-Lunit	✗	**0.76**	0.52	0.62	0.15	0.24	<u>0.38</u>	0.87	<u>0.51</u>
	RN50-CCL	✗	<u>0.75</u>	<u>0.56</u>	0.60	0.20	0.24	**0.39**	0.86	<u>0.51</u>
MAF-Setting										
0.50 µm/px (×20)	RN50-Lunit	✓	0.69	0.53	<u>0.63</u>	0.13	**0.40**	**0.39**	0.88	**0.52**

from WSIs for the entire cohort. Manually identifying these regions is too time-consuming. The lead pathologists of the project stated that annotating a single WSI would take about 3-4 hours[2], depending on the complexity of the contours and the amount of tissue on the slide. Given the cohort size of 400 internal samples and the additional 182 public TCGA-PAAD validation samples, this would result in an estimated total workload of approximately $1{,}746\,\mathrm{h}$, assuming an average of $3\,\mathrm{h}$ per slide. This, by far, exceeds the annual working time of approximately $1{,}592\,\mathrm{h}$ hours for a full-time employee in Germany (as of 2023) [127].

Model Training As part of the Selocan project, a pathologist annotated between 1 and 5 ROIs per sample across 62 WSIs ($0.25\,\mu\mathrm{m/px}$). Additionally, a test cohort consisting of 9 fully annotated samples was created to assess WSI-level performance. Tissue was segmented into the classes "tumor", the three stromal subtypes "reactive", "intermediate", and "deserted", "adipose tissue", and "other non-tumoral tissue". Non-tissue areas were additionally defined as "background". Examples of the subTME types are shown in Figure 4.6a.

However, given the increasing availability of in-domain pretrained encoder models for CPath, we also evaluated whether these models could further improve performance compared to ImageNet pretrained networks. For this purpose, we included two in-domain models based on the ResNet50 architecture in the comparison. The first, referred to as RN50-CCL, was pretrained by Wang et al. [275] on $15\,\mathrm{M}$ patches cropped from $32{,}000$ publicly available WSIs including TCGA and Paip. The second, referred to as RN50-Lunit, was pretrained by Kang et al. [137] on $19\,\mathrm{M}$ million patches solely extracted from TCGA. A train-val split of 53/9 WSI was used for training, with the final evaluation conducted on the 9 fully annotated test WSIs (53/9/9 in total).

[2]Subjective estimation by the pathologists during project planning

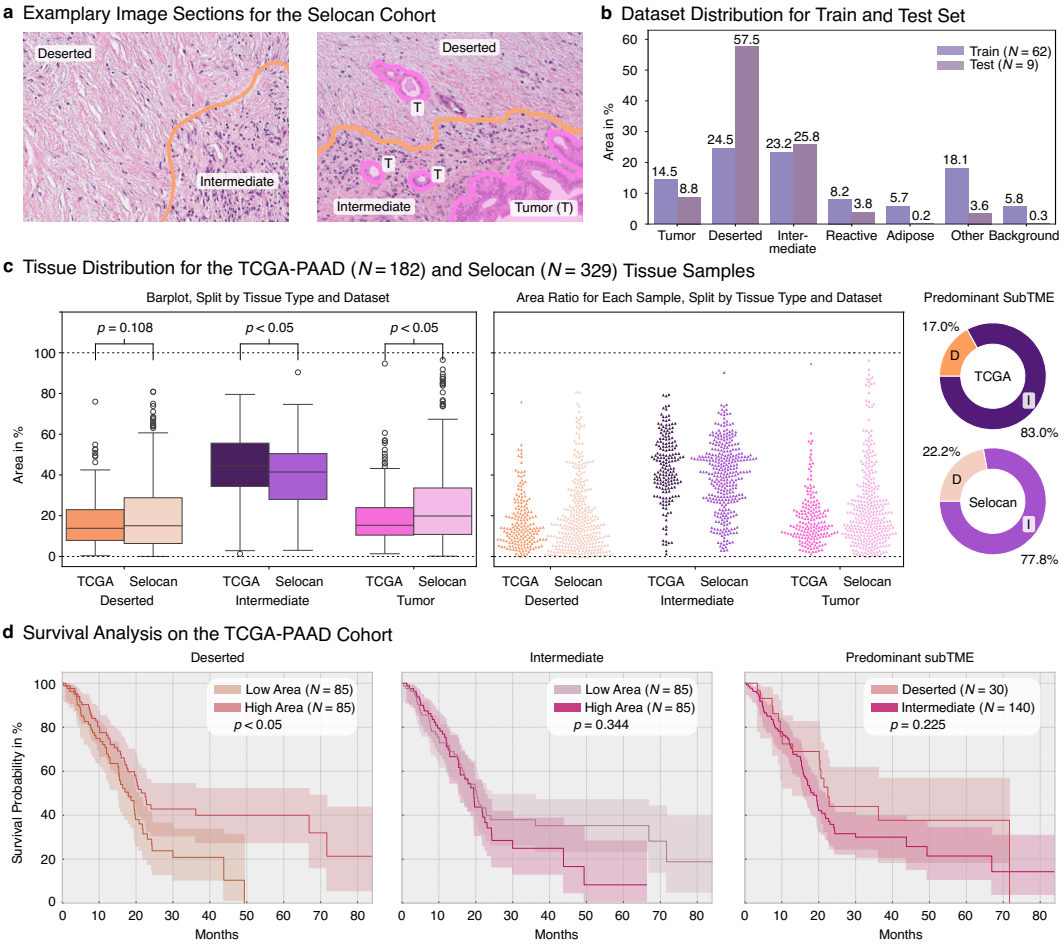


Figure 4.6: Pancreatic Cancer Cohort Evaluation
a) Exemplary image sections with ground truth tissue class overlay. **b**) Distribution of the train and test dataset. **c**) Post-inference results after applying the segmentation network on both the Selocan and TCGA cohorts. Two-sample Kolmogorov-Smirnov test was applied to check if the distributions differ (H_0: The distributions of the two datasets (Selocan and TCGA) are identical.). **d**) Kaplan-Meier analysis of subTME features and their association with overall survival. Log-rank tests were used to compare survival distributions.

Results The results of the test set are presented in Table 4.4. The in-domain pretrained models showed superior average performance compared to the ImageNet variants. Since deserted, intermediate, and reactive substroma classes are of particular importance for downstream evaluation, we selected $0.50\,\mu\mathrm{m/px}$ input resolution and the RN50-Lunit model for the MAF experiment, despite a modest lower $DICE_{\mathrm{Total}}$. Including the MAF further improved the average performance, i.e., for the non-tumor tissue regions. However, the performance for the stromal subtypes deteriorated and we decided not to include the MAF for inference. These findings

underscore the importance of task-specific evaluation as the model with the best overall performance is not always the best for the task. Some of the performance differences between individual tissue classes can be explained by the imbalance in the tissue annotations, especially for the reactive, adipose, and background classes (Figure 4.6b). The class definitions of the stromal subtypes also influenced the segmentation scores. Since borders between the subTMEs are ill-defined, the annotations suffer from intra-observer variability, impacting the scores. After consultation with the pathologist and medical advisory partners, we finally decided to use this model, as the performance measures and tissue assignments were satisfactory; however, we decided to merge the reactive subTME with the intermediate subTME. Further strategies using consensus annotations and fuzzy learning would have been too expensive at this stage but remain promising approaches for additional research projects.

Inference and Clinical Evaluation Inference was performed on the remaining 329 samples of the Selocan cohort and 182 samples of the TCGA-PAAD cohort. Figure 4.6c illustrates a comparison between the two cohorts by tumor composition and subtumoral microenvironment (subTME). Distribution of the area characterized as deserted stroma (as percentages) presents an identical pattern in both cohorts, having a similar quantitative distribution. Nonetheless, slight differences are observed in the distributions of intermediate stromal and tumor volume. These differences were statistically confirmed by applying a two-sided Kolmogorov-Smirnov test ($p < 0.05$). We measured minor differences in the predominant subTME of each cohort. In general, the intermediate stroma was dominant in approx. $80\,\%$ of the samples. To further investigate the clinical relevance of subTMEs, we next performed Kaplan-Meier survival analyses to assess the potential impact of the subTME and tumor composition on patient prognosis (Figure 4.6d). We quantified the percentage of subTME within an area of $40\,\mu\mathrm{m}$ around the tumor borders for each isolated tumor area and calculated sample-wise averages. The patients were then divided into high and low-risk groups based on the median of the distribution. Our analysis revealed that a low spread of deserted subTME areas around tumors was associated with poor survival (log-rank test: $p < 0.05$). In contrast, no significant effect was found for the intermediate and dominant subTME patterns. However, a trend towards decreased survival was observed in the highly expressed intermediate subTME group, though this finding was not statistically significant (log-rank test: $p = 0.344$). The findings indicate that tissue composition, more specifically the existence of residual subTME regions, may be a possible marker for survival. Further research is necessary to expand on these preliminary results. We hypothesize that the spatial relationship between the identified regions and their topologies will reveal more meaningful patterns. We further aim to incorporate cellular data because both the deserted and intermediary subTME differ in cell composition and spatial distribution.

4.6 Chapter Conclusion

In this Chapter, we introduced an extension to semantic segmentation networks that incorporates the domain characteristics of CPath. Using a patch-neighbor attention mechanism, which queries the neighboring tissue context, we improved conventional encoder-decoder segmentation networks with context information. Our approach outperforms both patch-based segmentation algorithms and competitive context integration methods that utilize context FOVs across three datasets. In particular, the performance gains of the proposed MAF were mainly prominent in the challenging RCC dataset. Based on the comparison with the CY16 and Paip 2019 datasets with fewer classes, we conclude that the MAF is more beneficial for complex segmentation tasks. Regarding the resolution of the input patches, no definitive pattern emerged, but the reasonable range could be constrained to $(0.25\,\mu\mathrm{m/px})$ to $(2.00\,\mu\mathrm{m/px})$. Across all datasets, the DeepLabV3 model consistently outperformed the U-Net model, which we attribute to the atrous (dilated) convolutions and atrous spatial pyramid pooling mechanisms, which should improve the acquisition of multiscale contextual information within an image [42].

Beyond the experimental evaluation, we showed that the framework can be used to analyze clinical cohorts. Specifically, we found that histological tissue patterns in the stroma of pancreatic cancer patients correlate with patient survival.

In summary, the contributions are as follows:

Contribution 1: We introduced a new method for incorporating tissue context information into existing encoder-decoder network architectures, without additional context patches.

Contribution 2: We demonstrated that our method can be beneficial for tissue segmentation for multiple tissue types, outperforming existing competing solutions

Contribution 3: Using pancreatic cancer cohorts (Selocan, TCGA-PAAD), we evaluated the clinical applicability of segmentation networks and demonstrated that histological patterns correlate with patient survival.

5

Enhancing Cell-level Analysis: CellViT and Beyond

In this Chapter, we propose two progressive methods for panoptic nuclei segmentation. In Chapter 5.2, we introduce the CellViT model, the first to leverage the ViT architecture for panoptic nuclei segmentation, achieving remarkable performance on the challenging PanNuke dataset. Building on this, we introduce an extension (CellViT^{++}) in Chapter 5.3. This extended model uses the existing CellViT model as segmentation backbone and incorporates a classification module to extend the model for new cell taxonomies. Through this extension, the model can be fine-tuned with minimal training data.

5.1 The Importance of Cell-level Analysis

Cell-level analysis is an important element of histological tissue evaluation, complementing the study of macroscopic tissue architecture. It is vital for diagnosing and classifying diseases, which depend on the number, shapes, and sizes of cells [280], especially the nuclei. For instance, tumor-infiltrating lymphocytes and inflammatory cells within the tumor microenvironment are important markers for breast cancer [4, 252]. Therefore, cell nuclei in histological images must be accurately detected and segmented. Furthermore, cell segmentation is widely used in the emerging field of spatial transcriptomics [129, 173]. Researchers employ cell segmentation and cell-type-specific gene expression knowledge to analyze cell morphology, location, single-cell gene expression, and detect intracellular variations [173].

However, this task is not feasible for pathologists at a large scale due to its time consuming nature and the high degree of intra- and inter-observer variability [119]. Thus, the need for automated and reliable cell detection and segmentation in conjunction with cell feature extraction in WSI is evident. Deep Learning methods based on CNNs [95, 204, 250, 277] became highly influential tools in this context. Nevertheless, this task remains challenging due to inconsistencies in cell morphology, staining intensity, and the presence of overlapping or touching nuclei.

The following Section 5.2 is built on our Medical Image Analysis publication from 2024 [119]. An entire transparency statement is attached at the end of this dissertation in the Appendix.

5.2 CellViT: A Novel Approach to Cell Segmentation

Nuclei segmentation approaches can be divided into two groups, conventional feature-based approaches [3, 52, 167, 181, 256, 267, 282, 294] and DL-based methods [38, 46, 95, 124, 204, 229, 239, 249, 277]. Over the last few years, DL methods have become the standard method due to their improved performance and independence of domain expertise. These DL-based approaches mainly rely on CNN-based detection [95] or segmentation networks. Following the emergence of ViT-based foundation models and their superior performance in various clinical benchmarks [30], we developed a novel model architecture for panoptic nucleus segmentation using ViTs. A high-level overview of the architecture is shown in Figure 5.1. The input image is tokenized and embedded via a ViT encoder, linked to the decoder through skip connections for segmentation. At the same time, cell embeddings are extracted and made available for downstream tasks.

In the following Sections, we present the method and evaluate its performance on three datasets. Specifically, we compare the in-domain foundation model HIPT-256, pretrained on $104\,\mathrm{M}$ tissue patches, against the generic Segment Anything Model, pretrained on natural images.

5.2.1 Methods

5.2.1.1 Multi-Branch Network Architecture

Our architecture, inspired by the UNETR model [107] for 3D image analysis, is adapted for 2D images, as depicted in Figure 5.2. In contrast to conventional segmentation networks with a single decoder, our network incorporates three multitask output branches (NP, HV, NT), as presented by the HoVer-Net network [95]. These branches are used to predict the nuclei (NP), split them into isolated instances (HV) and assign the nuclei type (NT):

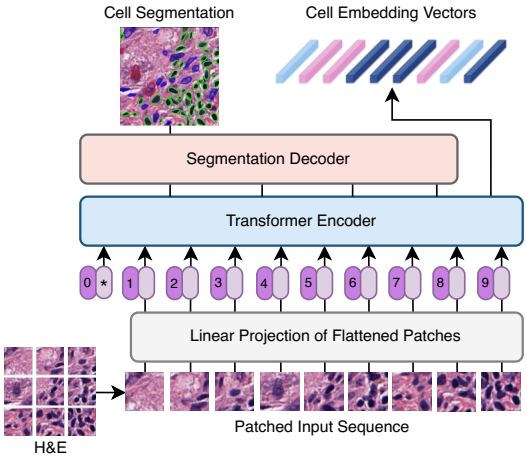


Figure 5.1: Network structure of CellViT. An input image is transformed into a sequence of tokens (flattened input sections). By using skip connections at multiple encoder depth levels and a dedicated upsampling decoder network, precise nuclei instance segmentations are derived. Nuclei embeddings are extracted from the Transformer encoder. Adapted from [119].

- NP-branch: Predicts binary nuclei map
- HV-branch: Predicts the horizontal and vertical distances of nuclear pixels to their center of mass, normalized between -1 and 1 for each nuclei
- NT-branch: Predicts the nuclei types as instance segmentation maps

To combine the output of each branch and split overlapping nuclei, postprocessing steps are required.

As an alternative approach, we also evaluate the performance of the STARDIST decoder method and its extension, CPP-Net. We integrate their techniques into the proposed UNETR-HoVer-Net architecture by modifying the NP-branch and HV-branch. Instead of the NP-branch, an object probability branch PD is used to predict whether a pixel belongs to a nucleus by predicting the Euclidean distance to the nearest background pixel. The HV-branch is replaced by an RD branch to predict the radial distances of an object pixel to the nucleus boundary (star-convex representation) [277]. The CPP-Net decoder further enhances the radial distance predictions through an additional refinement phase [46].

5.2.1.2 Encoder-Decoder Structure

The core component of our network architecture is the Vision Transformer used as an image encoder, illustrated in Figure 5.2a. Related to the U-Net architecture, the encoder is linked to an upsampling decoder network through skip connections. This architecture enables us to use ViTs as image encoders while preserving detailed information for segmentation. Although numerous modifications of the U-Net framework for Vision Transformers have been proposed, e.g., SwinUNETR [106], our approach was to choose the network design that is based on the original ViT structure by Dosovitskiy et al. [67] without any alterations. This allows us to use foundation models, such as HIPT-256 and SAM.

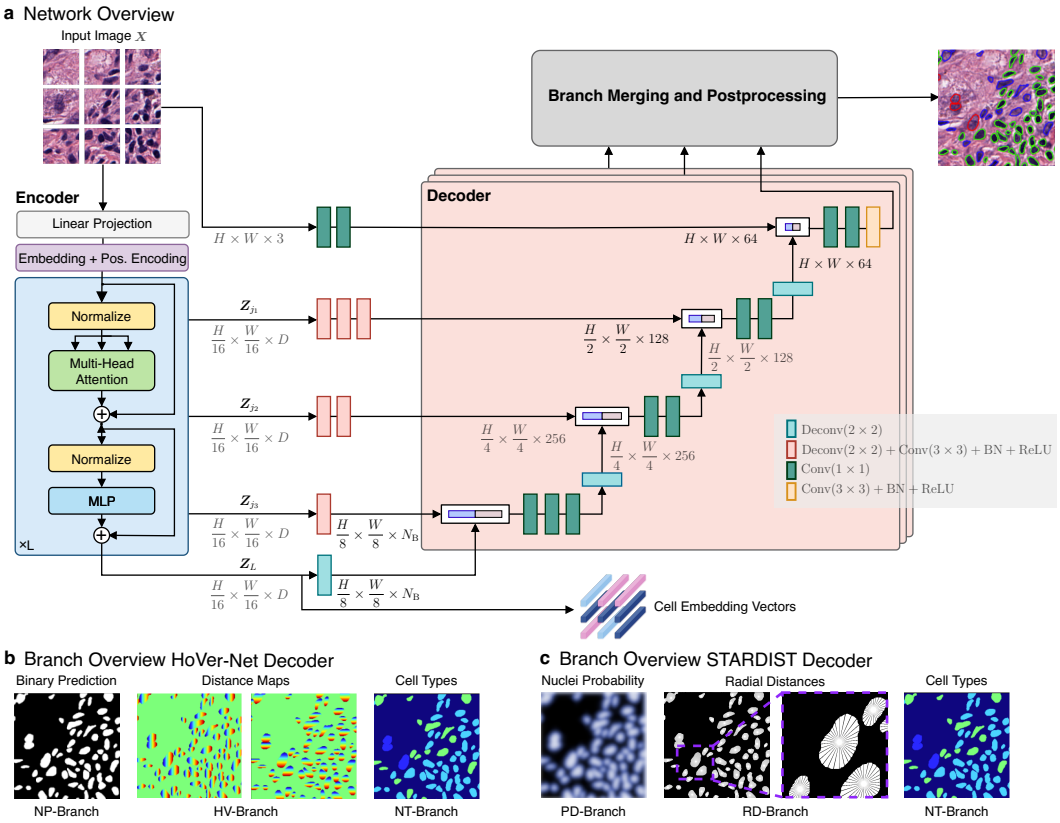


Figure 5.2: Network structure of our proposed CellViT architecture.
a) Encoder-Decoder based CellViT model. Both decoder structures, HoVer-Net and STARDIST, consist of multiple segmentation branches with identical architectures, differing only in the number of output channels. For visualization purposes, the tissue classification branch is not illustrated. **b)** HoVer-Net decoder branches: NP (binary nuclei prediction), HV (horizontal and vertical distance maps), and NT (nuclei type). **c)** STARDIST decoder branches: PD (nuclei probability), RD (radial distances), and again NT. Adapted from [119].

Encoder As introduced in Section 2.2, ViTs process a 1D sequence of token embeddings as their input [67, 266]. Therefore, an input image

$$\boldsymbol{X} \in \mathbb{R}^{H \times W \times 3}$$

with height H, width W and 3 input channels needs to be divided and transformed into a sequence of flattened tokens

$$\boldsymbol{X}_{\mathrm{P}} \in \mathbb{R}^{N \times (3 \cdot P^2)}$$
$$\boldsymbol{X}_{\mathrm{P}} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{P}}^1 & \boldsymbol{x}_{\mathrm{P}}^2 & \cdots & \boldsymbol{x}_{\mathrm{P}}^N \end{bmatrix}^{\mathsf{T}}, \ \boldsymbol{x}_{\mathrm{P}}^i \in \mathbb{R}^{3 \cdot P^2}.$$

Each token is a flattened representation of a squared image section with dimension $P \times P$. The number of tokens N can be calculated via $N = HW/P^2$, which is the effective length of the input sequence [107]. Accordingly, a linear projection layer

$$\boldsymbol{E} \in \mathbb{R}^{D \times 3 \cdot P^2}$$

is used to map the flattened tokens $\boldsymbol{X}_\mathrm{P}$ into a D-dimensional latent space. The latent vector size D remains constant through all L layers of the Transformer. In contrast to the UNETR network, we add a learnable class token $\boldsymbol{x}^\mathrm{class}$ to the token sequence for classification purposes [67].
As Transformers are permutation invariant such that they cannot inherently capture spatial relationships, a learnable positional embedding $\boldsymbol{E}_\mathrm{pos} \in \mathbb{R}^{(N+1) \times D}$ is added to the projected token embeddings to preserve spatial context [107]. In summary, the final input sequence $\boldsymbol{Z}_0$ for the Transformer encoder calculates as follows:

$$\begin{aligned}
\boldsymbol{Z}_0 &= \left[\boldsymbol{E} \cdot \boldsymbol{x}^\mathrm{class} \quad \boldsymbol{E} \cdot \boldsymbol{x}_\mathrm{P}^1 \quad \cdots \quad \boldsymbol{E} \cdot \boldsymbol{x}_\mathrm{P}^N\right]^\mathsf{T} + \boldsymbol{E}_\mathrm{pos},\ \boldsymbol{Z}_0 \in \mathbb{R}^{(N+1) \times D} \\
\boldsymbol{Z}_0 &= \left[\boldsymbol{z}_0^\mathrm{class} \quad \boldsymbol{z}_0^1 \quad \cdots \quad \boldsymbol{z}_0^N\right]^\mathsf{T},\ \boldsymbol{z}_0^i \in \mathbb{R}^D.
\end{aligned}$$

The Transformer encoder comprises alternating layers of multi-headed self-attention (MHA) [67] and multilayer perceptrons (MLP). A ViT is composed of several stacked Transformer blocks such that the latent tokens $\boldsymbol{Z}_i$ are calculated by

$$\begin{aligned}
\boldsymbol{Z}_l^{'} &= \mathrm{MHA}(\mathrm{Norm}(Z_{l-1})) + \boldsymbol{Z}_{l-1},\ l = 1 \ldots L \\
\boldsymbol{Z}_l &= \mathrm{MLP}(\mathrm{Norm}(Z_{l-1}')) + \boldsymbol{Z}_{l-1}',\ l = 1 \ldots L,
\end{aligned}$$

with L denoting the number of Transformer blocks, $\mathrm{Norm}(\cdot)$ denoting layer normalization, and l is the intermediate block identifier [107] (Figure 5.2). The MHA is identical to the mapping described in eq. (2.4).

Decoder Building on the U-Net and UNETR frameworks, we add five skip connections between the encoder and decoder to utilize information from ascending encoder depths during decoding. The first skip connection takes the input image $\boldsymbol{X}$ and processes it through two convolutional layers (3×3 kernel size) with batch-normalization and ReLU activation functions. The subsequent four skip connections propagate the intermediate and bottleneck latent tokens $\boldsymbol{Z}_j,\ j \in \left\{\frac{L}{4}, \frac{2L}{4}, \frac{3L}{4}, L\right\}$ (without the class token)

$$\boldsymbol{Z}_j \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times D},\ j \in \left\{\frac{L}{4}, \frac{2L}{4}, \frac{3L}{4}, L\right\},$$

matching the input sequence spatial ordering (x, y grid with depth D). This is only valid if $4 \mid L$ holds, which is commonly satisfied for common ViT implementations [44, 67, 142]. Each of the feature maps $\boldsymbol{Z}_j$ is transformed by a combination of

deconvolutional layers (Deconv(·)) that doubles the resolution in both dimensions and convolutions (Conv(·)) to adjust the latent dimension (number of channels). Subsequently, the transformed feature maps are successively processed in the decoder, starting from $\boldsymbol{Z}_L$, and fused with the corresponding skip connection at each stage. As illustrated in Figure 5.2, the three segmentation branches (NP, HV, NT) share the same image encoder with the same skip connections but unique upsampling pathways.

To take advantage of the additional tissue type information available in the PanNuke dataset, we introduce a tissue classification (TC) branch to guide the encoder learning process. For this purpose, we use the class token z_L^{class} of the last Transformer layer L and pass it into a linear layer followed by a softmax nonlinearity to predict the tissue class.

5.2.1.3 Target and Losses

To improve the training speed and network convergence, different loss functions are used for each network branch (based on TSFD-Net [124]). The total loss is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NP}} + \mathcal{L}_{\text{HV}} + \mathcal{L}_{\text{NT}} + \mathcal{L}_{\text{TC}} \tag{5.1}$$

where $\mathcal{L}_{\text{NP}}$ is the loss for the NP-branch, $\mathcal{L}_{\text{HV}}$ the loss for the HV-branch, $\mathcal{L}_{\text{NT}}$ the loss for the NT-branch, and $\mathcal{L}_{\text{TC}}$ the loss for the TC-branch. Overall, the individual branch losses are composed of the following weighted loss functions:

$$\begin{aligned}
\mathcal{L}_{\text{NP}} &= \lambda_{\text{NP}_{\text{FT}}}\mathcal{L}_{\text{FT}} + \lambda_{\text{NP}_{\text{DICE}}}\mathcal{L}_{\text{DICE}} \\
\mathcal{L}_{\text{HV}} &= \lambda_{\text{HV}_{\text{MSE}}}\mathcal{L}_{\text{MSE}} + \lambda_{\text{HV}_{\text{MSGE}}}\mathcal{L}_{\text{MSGE}} \\
\mathcal{L}_{\text{NT}} &= \lambda_{\text{NT}_{\text{FT}}}\mathcal{L}_{\text{FT}} + \lambda_{\text{NT}_{\text{DICE}}}\mathcal{L}_{\text{DICE}} + \lambda_{\text{NT}_{\text{BCE}}}\mathcal{L}_{\text{BCE}} \\
\mathcal{L}_{\text{TC}} &= \lambda_{\text{TC}_{\text{CE}}}\mathcal{L}_{\text{CE}}
\end{aligned} \tag{5.2}$$

The cross-entropy loss $\mathcal{L}_{\text{BCE}}$ and DICE loss $\mathcal{L}_{\text{DICE}}$ are commonly used in semantic segmentation. To overcome the problem of underrepresented instance classes, the Focal Tversky loss $\mathcal{L}_{\text{FT}}$ is used, which is a generalization of the Tversky loss [1, 168]. The Focal Tversky loss focuses more on correct instance identification of underrepresented instances by assigning more weights to these samples [1]. Due to increases loss value for underrepresented classes, it helps the model to deal with the imbalance.

Given the total number of pixels N_{px} in an image and N_{C} classes, the segmentation losses for each prediction $\hat{\boldsymbol{Y}}$ and ground truth segmentation $\boldsymbol{Y}$ are defined

by

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{n}\sum_{i=1}^{N_{\text{px}}}\sum_{c=1}^{N_{\text{C}}} y_{ic}\log\hat{y}_{ic} \tag{5.3}$$

$$\mathcal{L}_{\text{DICE}} = 1 - \frac{2\cdot\sum_{i=1}^{N_{\text{px}}} y_{ic}\hat{y}_{ic} + \varepsilon}{\sum_{i=1}^{N_{\text{px}}} y_{ic} + \sum_{i=1}^{N_{\text{px}}}\hat{y}_{ic} + \varepsilon} \tag{5.4}$$

$$\mathcal{L}_{\text{FT}} = \sum_{c=1}^{N_{\text{C}}}\left(1 - \frac{\sum_{i=1}^{N_{\text{px}}} y_{ic}\hat{y}_{ic} + \varepsilon}{\sum_{i=1}^{N_{\text{px}}} y_{ic}\hat{y}_{ic} + \alpha_{\text{FT}}\sum_{i=1}^{N_{\text{px}}} y_{ic}\hat{y}_{ic} + \beta_{\text{FT}}\sum_{i=1}^{N_{\text{px}}} y_{ic}\hat{y}_{ic}}\right)^{\frac{1}{\gamma_{\text{FT}}}} \tag{5.5}$$

where the contribution of each branch loss (5.2) to the total loss (5.1) is regulated by the i-th hyperparameters λ_i. In the segmentation losses (5.3)-(5.5), y_{ic} is the ground truth and $\hat{y}_{ic}$ the predicted probability of the i-th pixel belonging to the class c, ε a smoothness factor, and $\alpha_{\text{FT}}, \beta_{\text{FT}}$ and γ_{FT} are hyperparemters of the Focal Tversky loss $\mathcal{L}_{\text{FT}}$.

$\mathcal{L}_{\text{MSE}}$ denotes the mean squared error of the horizontal and vertical distance maps and $\mathcal{L}_{\text{MSGE}}$ the mean squared error of the gradients of the horizontal and vertical distance maps, each summarized separately for both directions.
For the tissue classification branch TC, the standard cross-entropy loss

$$\mathcal{L}_{\text{CE}} = -\sum_{c_{\text{T}}=1}^{C_{\text{T}}} y_{c_{\text{T}}}\log\hat{y}_{c_{\text{T}}},\ C_{\text{T}} = 19,$$

with a total of 19 tissue classes (C_{T}) is used. Further details on the choice of weights for eq. (5.2) can be found in the Appendix.

5.2.1.4 Postprocessing

Since the network does not directly output a panoptic segmentation with disjoint instances, postprocessing is required. Several steps are necessary to split up nuclei and assign semantic nuclei types. Because of the spatial extent of WSIs, inference must be performed on patches extracted using a sliding-window approach. The predictions must then be fused in the areas that overlap between patches.

Nuclei Separation and Classification To separate the adjacent and overlapping nuclei from each other, we employ HoVer-Net's postprocessing pipeline. First, the gradients of the horizontal and vertical distance maps are calculated to detect the changes between the nuclei boundaries and the background. The gradient is high at the transition between nuclei or between a nucleus and background. The boundaries can be derived using the Sobel operator (edge detection filter) in combination with a marker-controlled watershed algorithm. To determine the

nuclei class, the output of the separated nuclei and the type predictions are merged by using majority voting [95].

The STARDIST and CPP-Net decoder methods use non-maximum suppression (NMS) to eliminate redundant polygons that are likely to represent the same object [239, 277].

Inference The encoder ViT provides a benefit for performing inference on gigapixel WSI over CNNs based U-Nets. Its capability to process input sequences of arbitrary length, constrained only by memory consumption and positional embedding interpolation, allows for increased input image sizes during inference. It is important to note that positional embedding interpolation must be considered when scaling the input images. In preliminary experiments conducted on the MoNuSeg dataset (Section 5.2.3.3), we observed that the network performs equivalently processing a single $1{,}024 \times 1{,}024\,\mathrm{px}$ patch or dividing the same patch into $256 \times 256\,\mathrm{px}$ sub-patches with an overlap of $64\,\mathrm{px}$. Based on these findings, we perform WSI inference using $1{,}024 \times 1{,}024\,\mathrm{px}$ large patches with a $64\,\mathrm{px}$ overlap and merge the overlapping nuclei within the small $64\,\mathrm{px}$ overlap regions.

A unique feature of the ViT architecture is that a corresponding embedding vector is generated inherently for each input token. Thus, for each detected nucleus $\hat{y}$, we also store the corresponding ViT token $z_L^{\hat{y}} \in \mathbb{R}^D$, which usage is further explained in Section 5.3.

5.2.2 Experimental Setup

5.2.2.1 Datasets

A total of three datasets have been used during our experiments. The primary dataset for training and evaluation is the pan-cancer dataset PanNuke [84, 95]. For external validation, we additionally used the MoNuSeg [149, 150] and CoNSeP [95] datasets.

PanNuke This dataset comprises $189{,}744$ annotated nuclei in $7{,}904$ images with a size of $256 \times 256\,\mathrm{px}$ and encompasses 19 unique tissue types. The nuclei are separated into five distinct categories (neoplastic, connective, inflammatory, epithelial, dead), with the class distributions shown in Figure 5.3. The specimens were captured at $\times 40$ magnification with a resolution of $0.25\ \mu\mathrm{m/px}$. The dataset is highly imbalanced, especially the nuclei class of dead cells is underrepresented (see Figure 5.3). PanNuke is acknowledged as one of the most challenging datasets for performing the simultaneous nuclei instance segmentation task [124].

MoNuSeg The MoNuSeg [149, 150] dataset is used as an external validation dataset. Unlike PanNuke, it is considerably smaller and does not categorize nuclei into distinct classes. Furthermore, it just provides binary instance segmentation masks. For this work, we only used the MoNuSeg test dataset to evaluate our model. This test set consists of 14 images with a resolution of $1{,}000 \times 1{,}000\,\mathrm{px}$, acquired at $\times 40$ magnification with $0.25\ \mu\mathrm{m/px}$. In total, MoNuSeg contains more than $7{,}000$ annotated nuclei across the seven organ types kidney, lung, colon, breast, bladder, prostate, and brain in several disease states (benign and tumors at different stages). To process the dataset more effectively with our ViT-based networks with a token size of $P = 16\,\mathrm{px}$, we resized the data to a size of $1{,}024 \times 1{,}024\,\mathrm{px}$. Given the large patch size of the original dataset, we also created a $\times 20$ dataset with a resolution of $0.50\ \mu\mathrm{m/px}$ ($\times 20$ magnification), resulting in a patch size of $512 \times 512\,\mathrm{px}$ accordingly.

CoNSeP In order to assess the feature representation capabilities of the cell embeddings produced by the CellViT model, the CoNSeP (colorectal nuclear segmentation and phenotypes) dataset of Graham et al. [95] was employed. This dataset consists of 41 H&E-stained colorectal adenocarcinoma WSI at a resolution of $0.25\ \mu\mathrm{m/px}$ and an image size of $1{,}000 \times 1{,}000\,\mathrm{px}$, which we rescaled to $1{,}024 \times 1{,}024\,\mathrm{px}$. The dataset is heterogeneous, including stromal, glandular, muscular, collagen, adipose, and tumor regions, and various nuclei types originating from different cells: normal epithelial, dysplastic epithelial, inflammatory, necrotic, muscular, fibroblast, and miscellaneous nuclei, which include necrotic and mitotic cells.

5.2.2.2 Experiments

To assess the efficacy of the proposed algorithms, two experiments were conducted using the PanNuke dataset while one experiment used the MoNuSeg dataset. Furthermore, on the best-performing CellViT variants, we analyzed the cell embeddings and their representation capability on the CoNSeP dataset. We also used an internal dataset to compare the inference time of the model to competing solutions.

PanNuke Since detecting nuclei is more clinically important than refining segmentation quality, we **(1)** performed an ablation study on PanNuke to identify the most efficient network architecture for nuclei detection by assessing different network variations (Section 5.2.3.1). These include a randomly initialized network ($\text{CellViT}_{\text{Random}}$), networks with pretrained weights from the HIPT-256 network ($\text{CellViT}_{\text{HIPT-256}}$), and networks with different pretrained SAM backbones ($\text{CellViT}_{\text{SAM-B}}$, $\text{CellViT}_{\text{SAM-L}}$, $\text{CellViT}_{\text{SAM-H}}$). To ensure comparability, the $\text{CellViT}_{\text{Random}}$ network follows the same architecture (ViT-S, $D = 384$, $L = 12$) as the $\text{CellViT}_{\text{HIPT-256}}$ network. We analyzed the impact of regularization techniques such as data augmentation, loss functions, and customized oversampling and compared the HoVer-Net decoder method to the STARDIST and CPP-Net decoders.

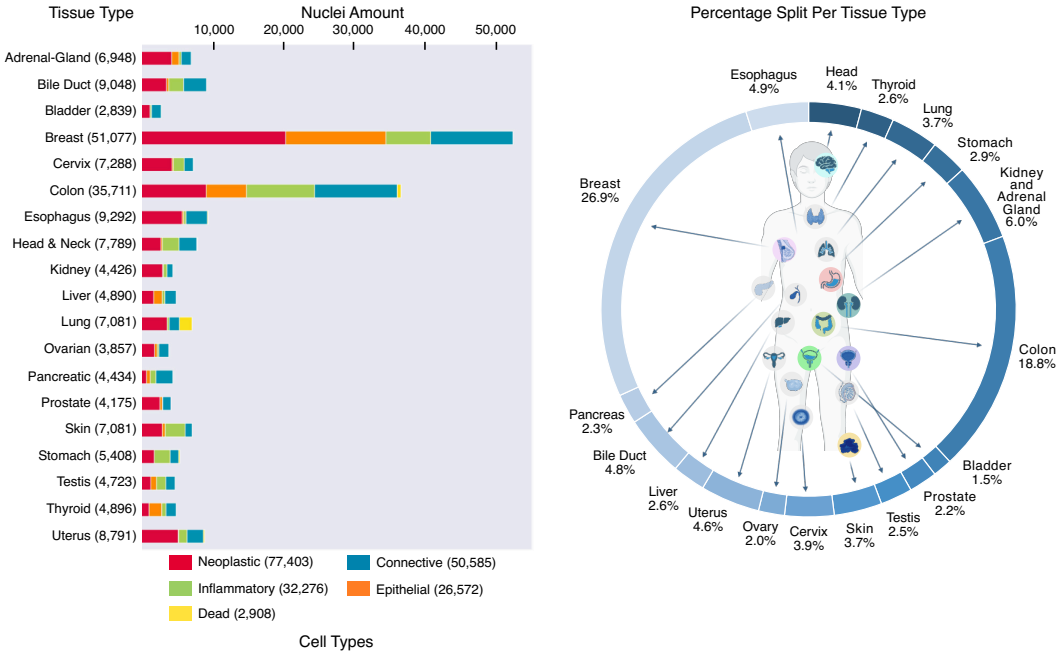


Figure 5.3: PanNuke nuclei distribution overview for each of the 19 tissue types, sorted by the total number of nuclei given in parentheses. Adapted from [84, 116, 119]. Created with [57].

Based on these investigations, the best model candidates were further (**2**) evaluated for segmentation quality (Section 5.2.3.2). To evaluate the detection and segmentation performance, we compared our models against several baseline architectures, including DIST [204], Mask-RCNN [111], Micro-Net [229], HoVer-Net [95], TSFD-Net [124], and CPP-Net [46]. Further, we retrained the STARDIST model [277] with a ResNet50 backbone and the hyperparameters of Chen et al. [46]. In the comparison, we performed our experiments using the same three-fold cross-validation splits provided by the PanNuke dataset, and report the averaged results.

MoNuSeg To assess the generalization of CellViT, we evaluated our PanNuke models on 14 publicly available test images of the MoNuSeg dataset (**3**) (Section 5.2.3.3). In this context, we compared the performance using two scenarios, one with an unpatched MoNuSeg slide and $1{,}024$ px input patch size and the other using patched 256 px input images. Additionally, we investigated the impact of our overlapping inference strategy with a 64 px overlap, focusing on the 256 px input size. Given that the dataset is available for both $0.25\,\mu\text{m/px}$ and $0.50\,\mu\text{m/px}$, we also tested the resolution invariance of all models.

CoNSeP We analyzed the cell embeddings for detected nuclei using the CellViT models with the CoNSeP dataset **(4)** (Section 5.2.3.4). Inference was performed on CoNSeP images using PanNuke pretrained CellViT models, extracting the corresponding token (cell) embeddings $\boldsymbol{z}_L^{\hat{y}} \in \mathbb{R}^D$ for each nucleus $\hat{y}$ from the final Transformer block L. The embeddings were then reduced to two dimensions using the Uniform Manifold Approximation and Projection (UMAP) method [188], applied to the 27 training images. A linear classifier was trained on the extracted embeddings from these training images to classify the detected cells into the CoNSeP nuclei classes. The classifier was then tested on the cell embeddings of the cells from the 14 test images.

Internal Dataset Finally, to compare the inference runtime **(5)**, we collected a diverse dataset of 10 esophageal WSIs with tissue areas ranging from $2.79\ \mathrm{mm}^2$ to $74.07\ \mathrm{mm}^2$ (Section 5.2.3.5). We measured the inference runtime for the HoVer-Net model, as well as for the CellViT$_{\text{HIPT-256}}$ and CellViT$_{\text{SAM-H}}$ models with patch input size $256\,\mathrm{px}$ and $1{,}024\,\mathrm{px}$ patch input size and overlap of $64\,\mathrm{px}$. For each WSI, we repeated the process three times and report the averaged runtime results.

5.2.2.3 Evaluation Metrics

Nuclear Classification Evaluation To evaluate the detection quality, we employed commonly used detection metrics. Given is a ground truth (GT) segment y and a predicted segment $\hat{y}$, with the pair $(y, \hat{y})$ being a unique matching set of a GT segment and one predicted segment. For each class $c \in \{1, .., N_{\mathrm{C}}\}$, the unique matching of $(y, \hat{y})$ divides the predicted and GT segments into three sets:

- True Positives (TP): Matched pairs of segments, i.e., correctly detected instances
- False Positives (FP): Unmatched predicted segments, i.e., predicted instances without matching GT instance
- False negatives (FN): Unmatched GT segments, i.e., GT instances without matching predicted instance.

We used the methodology of Sirinukunwattana et al. [247] and define a match $(y, \hat{y})$ if both centers of mass are within a radius of $6\,\mathrm{px}$ $(0.50\ \mu\mathrm{m/px})$ and $12\,\mathrm{px}$ $(0.25\ \mu\mathrm{m/px})$, respectively.

As metrics, we then used the conventional detection metrics precision (P_{d}), recall (P_{d}), and the $F_{1,\mathrm{d}}$-score as a harmonic mean between precision and recall. The index d indicates that these are the scores for the entire binary nuclei detection

over all classes c. Thus, the binary detection scores are defined as follows:

$$F_{1,\mathrm{d}} = \frac{2TP_{\mathrm{d}}}{2TP_{\mathrm{d}} + FP_{\mathrm{d}} + FN_{\mathrm{d}}} \tag{5.6}$$

$$P_{\mathrm{d}} = \frac{TP_{\mathrm{d}}}{TP_{\mathrm{d}} + FP_{\mathrm{d}}} \tag{5.7}$$

$$R_{\mathrm{d}} = \frac{TP_{\mathrm{d}}}{TP_{\mathrm{d}} + FN_{\mathrm{d}}} \tag{5.8}$$

We further break down TP_d into correctly classified instances of class c (TP_c), false positives of class c (FP_c), and false negatives of class c (FN_c) to derive cell-type specific scores. We then define the $F_{1,c}$-score, precision (P_c) and recall (R_c) of each nuclei class c as

$$F_{1,c} = \frac{2(TP_c + TN_c)}{2(TP_c + TN_c) + 2FP_c + 2FN_c + FP_{\mathrm{d}} + FN_{\mathrm{d}}}, \tag{5.9}$$

$$P_c = \frac{TP_c + TN_c}{TP_c + TN_c + 2FP_c + FP_{\mathrm{d}}}, \tag{5.10}$$

$$R_c = \frac{TP_c + TN_c}{TP_c + TN_c + 2FN_c + FN_{\mathrm{d}}}. \tag{5.11}$$

In order to prioritize the classification of different nuclear types, we incorporated an additional weighting factor for the nuclei classes, as suggested in the official PanNuke evaluation metrics [84, 95].

In the case of cell detection, the metrics can be explained as follows. Precision is the accuracy of the detected cells with respect to the ground truth. A high precision means that a small number of FP cells are present while most of the detected cells are correct. In the case of low precision, many FP cells are detected that are actually not present in the specimen. On the other hand, recall is the ability of the model to detect cells in the specimen. Models with high recall detect most of the present cells, and models with low recall rate are likely to miss many cells. The F_1-Score is the harmonic mean of precision and recall, so it considers both precision and recall.

Nuclear Instance Segmentation Evaluation Usually, the $DICE$ coefficient (see eq. 4.5) or the Jaccard index (AJI) are used as evaluation metrics for semantic segmentation. However, as Graham et al. [95] have already shown, these two metrics are insufficient for evaluating nuclear instance segmentation (panoptic segmentation) as they did not account for the detection quality of the nuclei. Therefore, a metric is needed that takes the following three requirements into account (compare Graham et al. [95]):

1. Separate the nuclei from the background
2. Detect individual nuclei instances and separate overlapping nuclei
3. Segment each instance

These three requirements cannot be evaluated with the Jaccard index and the DICE score, as they only satisfy requirement (1). In line with [95] and the PanNuke dataset evaluation recommendations [84], we use the panoptic quality (PQ) [141] to quantify the panoptic segmentation performance. The PQ is defined as

$$PQ = \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Detection Quality}(DQ)} \times \underbrace{\frac{\sum_{(y,\hat{y})\in TP} IoU(y,\hat{y})}{|TP|}}_{\text{Segmentation Quality}(SQ)}, \tag{5.12}$$

with $IoU(y, \hat{y})$ denoting the intersection-over-union [141] and TP, FP, and FN defined the same way as for the detection metrics. As Kirillov et al. [141] proved, each pair of segments $(y, \hat{y})$, i.e., each pair of true and predicted nuclei, in an image is unique if $IoU(y, \hat{y}) > 0.5$ is satisfied. The PQ score can be intuitively decomposed into two parts, the detection quality similar to the F_1-score commonly used in classification and detection scenarios, and the segmentation quality as the average IoU of matched segments [95, 141]. To ensure a fair comparison, we use binary PQ (bPQ), pretending that all nuclei belong to one class (nuclei vs. background) and the more challenging multi-class PQ (mPQ), taking the nuclei class into account. In doing so for mPQ, we calculated the PQ independently for each nuclei class and subsequently averaged the results over all classes [84].

5.2.2.4 Model Training

Oversampling Although the PanNuke dataset includes around 200,000 annotated nuclei, they are distributed across a limited number of 7,904 patches with 256×256 px patch size. Furthermore, there is a substantial class imbalance among tissue types and nuclei classes (Figure 5.3). Thus, we developed a new oversampling strategy based on class weightings to balance both tissue classes and nuclei classes. For each patch i in the training dataset with N_{Train} training samples, we calculated the sampling weights for the tissue class and the cell class with

$$p_i(\gamma_s) = \frac{w_{\text{Tissue}}(i, \gamma_s)}{\max\limits_{j\in[1,N_{\text{Train}}]} w_{\text{Tissue}}(j, \gamma_s)} + \frac{w_{\text{Cell}}(i, \gamma_s)}{\max\limits_{j\in[1,N_{\text{Train}}]} w_{\text{Cell}}(j, \gamma_s)}, \tag{5.13}$$

where $w_{\text{Tissue}}(i, \gamma_s)$ is a weight factor for the tissue class and $w_{\text{Cell}}(i, \gamma_s)$ for the nuclei class. The parameter $\gamma_s \in [0, 1]$ is a weighting factor that determines the strength of the oversampling. A γ_s value of 0 indicates no oversampling, while $\gamma_s = 1$ corresponds to maximum balancing. To ensure neither $w_{\text{Tissue}}(i, \gamma_s)$ nor $w_{\text{Cell}}(i, \gamma_s)$ dominates the sampling, normalization is applied to both summands in eq. (5.13). The calculation of the weighting factors for each tissue and cell class is explained in the Appendix.

Data Augmentation To increase the variety of the data and to avoid overfitting, we applied data augmentation techniques. We employed the following augmentation techniques: 90-degree rotation, horizontal flipping, vertical flipping, downscaling, blurring, gaussian noise, color jittering, superpixel representation of image sections (SLIC), zoom blur, random cropping with resizing and elastic transformations. The details of the augmentation methods, including the probabilities and hyperparameters, can be found in the Appendix (Table A.14).

Optimization and Training Strategy We trained all our models for 130 epochs and incorporated exponential learning rate scheduling with a scheduling factor of 0.85 to gradually reduce the learning rate during training (called the CellViT hyperparameters). To balance our training, we used the modified oversampling strategy (eq. (5.13)) with $\gamma_s = 0.85$. For the STARDIST and CPP-Net models, we also conducted experiments using the hyperparameters proposed by CPP-Net Chen et al. [46]. A complete overview of all hyperparameters, including optimizer, data augmentation, and weighting factors of the loss functions in equations (5.2) is provided in the Appendix (Table A.15-A.16).

We leveraged the HIPT-256 model (ViT-S, $D = 384$, $L = 12$) as the encoder, which was pretrained on histological data (Section 2.4). Additionally, we compared the performance with the three pretrained SAM checkpoints: SAM-B (ViT-B, $D = 768$, $L = 12$), SAM-L (ViT-L, $D = 1{,}024$, $L = 24$) and SAM-H (ViT-H, $D = 1{,}280$, $L = 32$). During training, we initially froze the encoder weights for the first 25 epochs. After this initial warm-up phase, we trained the entire model, including the image encoder.

Implementation All models have been implemented in PyTorch 1.13.1. To augment images and masks, we used the Albumentations library [24]. Other used libraries include the official STARDIST [239], CPP-Net [46] and CellSeg-models implementations [208]. For the pretrained HIPT-256-model, we utilized the ViT-S checkpoint[1] provided by Chen et al. [44]. As for the SAM-B, SAM-L, and SAM-H models, we use the encoder backbones of each final training stage of SAM [142], published on GitHub[2]. All experiments were conducted on an 80 GB NVIDIA A100 GPU with automatic mixed precision. However, it is worth noting that a 48 GB NVIDIA RTX A6000 is also sufficient for the HIPT-256 and SAM-H model training. An overview of all compared software versions is provided in the Appendix in Table A.17.

[1] https://github.com/mahmoodlab/HIPT

[2] https://github.com/facebookresearch/segment-anything

5.2.3 Results and Analysis

In the section below, the results for the experiments (1) nuclei detection quality and (2) segmentation quality on PanNuke, (3) generalization performance on the independent MoNuSeg cohort, (4) cell-embedding analysis, and (5) inference speed comparisons are given. If not stated otherwise, all models were trained on the PanNuke dataset with a resolution of $0.25\,\mu m/px$ and the HoVer-Net decoding strategy.

5.2.3.1 Detection Quality on PanNuke

Considering the clinical importance of nuclei detection and classification over achieving the best possible segmentation quality, we first want to determine the

Table 5.1: $\overline{F_1}$-score for detection and classification across the three PanNuke splits for each nuclei type. The centroid of each nucleus was used for computing detection metrics for segmentation networks. Adapted from [119].
*TSFD-Net was not evaluated on the official three-fold splits of the PanNuke dataset and left out by the comparison **Model retrained by ourselves ***Models trained on downscaled $0.50\,\mu m/px$ PanNuke samples.

Model	Decoder	Hyperparameter	Detection-Score					
			$\overline{F_{1,d}}$	$\overline{F_{1,Neo}}$	$\overline{F_{1,Epi}}$	$\overline{F_{1,Inf}}$	$\overline{F_{1,Con}}$	$\overline{F_{1,Dead}}$
DIST			0.73	0.50	0.35	0.42	0.39	0.00
Mask-RCNN			0.72	0.59	0.52	0.50	0.42	0.22
Micro-Net			0.80	0.62	0.58	0.52	0.47	0.19
HoVer-Net			0.80	0.62	0.56	0.54	0.49	0.31
TSFD-Net*			0.85	0.65	0.57	0.57	0.53	0.43
STARDIST (ResNet50) **	STARDIST	CPP-Net	0.82	0.69	0.70	0.57	0.51	0.10
STARDIST (ResNet50) **	STARDIST	CellViT	0.82	0.68	0.68	0.58	0.51	0.36
$CellViT_{HIPT\text{-}256}$ – Raw	HoVer-Net	CellViT	0.78	0.63	0.61	0.50	0.44	0.23
$CellViT_{HIPT\text{-}256}$ – Over	HoVer-Net	CellViT	0.78	0.62	0.62	0.50	0.44	0.24
$CellViT_{HIPT\text{-}256}$ – Aug	HoVer-Net	CellViT	0.82	0.69	0.69	0.58	0.52	0.36
$CellViT_{HIPT\text{-}256}$ – No-FC	HoVer-Net	CellViT	0.82	0.69	0.70	0.58	0.52	0.36
$CellViT_{Random}$ (no pretrain)	HoVer-Net	CellViT	0.80	0.64	0.72	0.55	0.48	0.31
$CellViT_{HIPT\text{-}256}$	HoVer-Net	CellViT	0.82	0.69	0.70	0.58	0.52	0.37
$CellViT_{SAM\text{-}B}$	HoVer-Net	CellViT	**0.83**	0.70	0.71	**0.59**	**0.53**	0.36
$CellViT_{SAM\text{-}L}$	HoVer-Net	CellViT	**0.83**	0.70	0.72	0.58	**0.53**	**0.39**
$CellViT_{SAM\text{-}H}$	HoVer-Net	CellViT	**0.83**	**0.71**	**0.73**	0.58	**0.53**	0.36
$CellViT_{HIPT\text{-}256}$	STARDIST	CPP-Net	0.79	0.62	0.60	0.52	0.47	0.28
$CellViT_{HIPT\text{-}256}$	STARDIST	CellViT	0.81	0.68	0.68	0.58	0.50	0.37
$CellViT_{SAM\text{-}H}$	STARDIST	CPP-Net	0.81	0.67	0.66	0.57	0.49	0.32
$CellViT_{SAM\text{-}H}$	STARDIST	CellViT	0.82	0.70	0.72	0.58	0.52	0.38
$CellViT_{HIPT\text{-}256}$	CPP-Net	CPP-Net	0.80	0.65	0.65	0.56	0.49	0.33
$CellViT_{HIPT\text{-}256}$	CPP-Net	CellViT	0.81	0.68	0.68	0.58	0.51	0.37
$CellViT_{SAM\text{-}H}$	CPP-Net	CPP-Net	0.82	0.70	0.70	0.58	0.52	0.18
$CellViT_{SAM\text{-}H}$	CPP-Net	CellViT	0.82	0.70	0.72	0.58	**0.53**	0.38
$CellViT_{HIPT\text{-}256}$ $(0.50\,\mu m/px)$***	HoVer-Net	CellViT	0.71	0.65	0.64	0.47	0.40	0.07
$CellViT_{SAM\text{-}H}$ $(0.50\,\mu m/px)$***	HoVer-Net	CellViT	0.73	0.67	0.67	0.49	0.42	0.08

best model based on the detection results using the PanNuke dataset. Table 5.1 presents the F_1-Score for the detection of each cell type, including the binary case ($F_{1,\text{d}}$). All mentioned CellViT variants were trained using data augmentation and our customized sampling strategy as a regularization method.

First, we analyze the CellViT models using the HoVer-Net decoder. Compared to the baseline models, the randomly initialized CellViT$_{\text{Random}}$ network achieved detection results comparable to the HoVer-Net CNN network. However, when using pretrained encoder networks, we observed a significant performance increase, reaching state-of-the-art performance. Both the CellViT$_{\text{HIPT-256}}$ and the three different SAM encoders exhibited superior performance (marked gray in Table 5.1), all at a similar level, with the CellViT$_{\text{SAM-H}}$ reaching the highest average scores. Notably, we even outperformed purely detection-based methods like Mask-RCNN and all current SOTA approaches by a large margin with up to a $26\ \%$ increase in the $F_{1,\text{Epi}}$-score of epithelial nuclei. Since TSFD was not evaluated using the official 3-fold cross-validation split with a 67/33 train-validation ratio and instead trained on a single 80/20 split, it has been excluded from the comparison.

To demonstrate the effect of extensive data augmentation, data sampling strategy, and the Focal Tversky loss, we additionally report the results for a CellViT$_{\text{HIPT-256}}$ model without regularization (CellViT$_{\text{HIPT-256}}$-Raw), with oversampling only (CellViT$_{\text{HIPT-256}}$-Over), with data augmentation only (CellViT$_{\text{HIPT-256}}$-Aug), and a model trained with oversampling and all augmentations, but without Focal Tversky loss (CellViT$_{\text{HIPT-256}}$-No-FC) in Table 5.1. Our experiments reveal that data augmentation, in particular, is a crucial regularization method that enhanced performance. Specifically, the addition of data augmentation resulted in a 0.13 increase in the $F_{1,\text{Dead}}$ score for the dead nuclei class compared to the CellViT$_{\text{HIPT-256}}$-Raw model. Oversampling and Focal Tversky loss just led to minimal improvements in detection scores. We also tested the STARDIST and CPP-Net decoder structures with the CellViT$_{\text{HIPT-256}}$ and CellViT$_{\text{SAM-H}}$ models with our hyperparameters and the CPP-Net hyperparameters suggested by Chen et al. [46]. In general, the models achieved better detection results than comparable CNN-based SOTA networks. They outperformed the ResNet50-based STARDIST model but were inferior to our suggested models with HoVer-Net decoder architecture.

In addition to the resolution of the provided dataset of $0.25\ \mu\text{m/px}$, we performed training and evaluation for the two best model variants, CellViT$_{\text{HIPT-256}}$ and CellViT$_{\text{SAM-H}}$, on downscaled PanNuke data (from 256×256 to $128 \times 128\,\text{px}$ patch size), resulting in $0.50\ \mu\text{m/px}$ resolution. The results are presented in the last two rows of Table 5.1. The downsizing led to a substantial performance drop compared to the $0.25\ \mu\text{m/px}$ networks. A deeper analysis using precision and recall (Table A.10 in the Appendix) reveals that the recall of individual classes decreased (by an average of -0.20). In particular, the recall for the dead nuclei class dropped to 0.04, indicating that this class is almost not detected at all. Interestingly, the precision increased slightly or remained almost the same compared to our best

$0.25\ \mu m/px$ models. We conclude that despite detecting significantly fewer nuclei when a nucleus is identified and classified correctly, it corresponds to the actual nucleus class with high accuracy for most classes.

For subsequent investigations, we decided to consider the CellViT$_{\text{HIPT-256}}$ and CellViT$_{\text{SAM-H}}$ models to allow a comparison between in-domain (CellViT$_{\text{HIPT-256}}$) and out-of-domain pretraining (CellViT$_{\text{SAM-H}}$).

5.2.3.2 Segmentation Quality on PanNuke

To assess segmentation quality, the panoptic quality PQ is utilized. Table 5.2 presents the PQ values for each nuclei type, averaged between all tissue types. Among all settings, the CellViT$_{\text{HIPT-256}}$ and CellViT$_{\text{SAM-H}}$ networks with the HoVer-

Table 5.2: Average $\overline{PQ}$ across the three PanNuke splits for each nuclear category. Best results are marked bold, second best underlined. *TSFD-Net was not evaluated on the official three-fold splits of the PanNuke dataset and left out by the comparison. **Model retrained by ourselves ***Models trained on downscaled $0.50\ \mu m/px$ PanNuke images. Table adapted from [119].
Abbr.: Decoder (Dec.), Hyperparameter (HP.), HoVer-Net (HV), STARDIST (SD), CPP-Net (CPP).

Model	Dec.	HP.	Neoplastic	Epithelial	Inflammatory	Connective	Dead
DIST			0.439	0.290	0.343	0.275	0.000
Mask-RCNN			0.472	0.403	0.290	0.300	0.069
Micro-Net			0.504	0.442	0.333	0.334	0.051
HoVer-Net			0.551	0.491	0.417	0.388	0.139
TSFD-Net*			0.572	0.566	0.453	0.423	0.214
STARDIST (RN50)**	SD	CPP	0.564	0.543	0.398	0.388	0.024
STARDIST (RN50)**	SD	CellViT	0.547	0.532	**0.424**	0.380	0.123
CellViT$_{\text{SAM-H}}$	HV	CellViT	**0.581**	**0.583**	0.417	**0.423**	<u>0.149</u>
CellViT$_{\text{HIPT-256}}$	HV	CellViT	0.567	0.559	0.405	<u>0.405</u>	0.144
CellViT$_{\text{HIPT-256}}$ – Raw	HV	CellViT	0.495	0.465	0.344	0.335	0.067
CellViT$_{\text{HIPT-256}}$ – Over	HV	CellViT	0.494	0.467	0.349	0.339	0.071
CellViT$_{\text{HIPT-256}}$ – Aug	HV	CellViT	0.565	0.558	<u>0.419</u>	0.403	**0.156**
CellViT$_{\text{HIPT-256}}$ – No-FC	HV	CellViT	0.567	0.548	0.416	0.404	0.141
CellViT$_{\text{HIPT-256}}$	SD	CellViT	0.516	0.507	0.400	0.331	0.128
CellViT$_{\text{SAM-H}}$	SD	CellViT	0.548	0.544	0.400	0.347	0.132
CellViT$_{\text{HIPT-256}}$	CPP	CellViT	0.540	0.524	0.414	0.369	0.133
CellViT$_{\text{SAM-H}}$	CPP	CellViT	<u>0.571</u>	<u>0.565</u>	0.405	0.395	0.131
CellViT$_{\text{HIPT-256}}$*** $(0.50\ \mu m/px)$	HV	CellViT	0.497	0.467	0.292	0.285	0.021
CellViT$_{\text{SAM-H}}$*** $(0.50\ \mu m/px)$	HV	CellViT	0.528	0.502	0.315	0.311	0.031

Table 5.3: Average $\overline{mPQ}$ and $\overline{bPQ}$ across the 19 tissue types of the PanNuke dataset for three-fold cross-validation. The standard deviation (SD) of the splits is provided in the final row. For the CellViT models, just the architecture with HoVer-Net decoder (HV-Net) is given. Best results are marked bold, second best underlined. Adapted from [119].
*TSFD-Net was not evaluated on the official three-fold splits of the PanNuke dataset and left out by the comparison **STARDIST trained by Chen et al. [46].

	HoVer-Net		TSFD-Net*		STARDIST** RN50 encoder		CPP-Net RN50 encoder		$\text{CellViT}_{\text{HIPT-256}}$ HV-Net decoder CellViT-HP		$\text{CellViT}_{\text{SAM-H}}$ HV-Net decoder CellViT-HP	
Tissue	$\overline{mPQ}$	$\overline{bPQ}$	$\overline{mPQ}$	$\overline{bPQ}$	$\overline{mPQ}$	$\overline{bPQ}$	$\overline{mPQ}$	$\overline{bPQ}$	$\overline{mPQ}$	$\overline{bPQ}$	$\overline{mPQ}$	$\overline{bPQ}$
Adrenal	0.4812	0.6962	0.5223	0.6900	0.4868	0.6972	0.4922	0.7031	0.4950	0.7009	**0.5134**	**0.7086**
Bile Duct	0.4714	0.6696	0.5000	0.6284	0.4651	0.6690	0.4650	0.6739	0.4721	0.6705	**0.4887**	**0.6784**
Bladder	0.5792	0.7031	0.5738	0.6773	0.5793	0.6986	**0.5932**	0.7057	0.5756	0.7056	0.5844	**0.7068**
Breast	0.4902	0.6470	0.5106	0.6245	0.5064	0.6666	0.5066	0.6718	0.5089	0.6641	**0.5180**	**0.6748**
Cervix	0.4438	0.6652	0.5204	0.6561	0.4628	0.6690	0.4779	**0.6880**	0.4893	0.6862	**0.4984**	0.6872
Colon	0.4095	0.5575	0.4382	0.5370	0.4205	0.5779	0.4269	0.5888	0.4245	0.5700	**0.4485**	**0.5921**
Esophagus	0.5085	0.6427	0.5438	0.6306	0.5331	0.6655	0.5410	**0.6755**	0.5373	0.6619	**0.5454**	0.6682
Head & Neck	0.4530	0.6331	0.4937	0.6277	0.4768	0.6433	0.4667	0.6468	0.4901	0.6472	**0.4913**	**0.6544**
Kidney	0.4424	0.6836	0.5517	0.6824	**0.5880**	0.6998	0.5092	0.7001	0.5409	0.6993	0.5366	**0.7092**
Liver	0.4974	0.7248	0.5079	0.6675	0.5145	0.7231	0.5099	0.7271	0.5065	0.7160	**0.5224**	**0.7322**
Lung	0.4004	0.6302	0.4274	0.5941	0.4128	0.6362	0.4234	0.6364	0.4102	0.6317	**0.4314**	**0.6426**
Ovarian	0.4863	0.6309	0.5253	0.6431	0.5205	0.6668	0.5276	**0.6792**	0.5260	0.6596	**0.5390**	0.6722
Pancreatic	0.4600	0.6491	0.4893	0.6241	0.4585	0.6601	0.4680	**0.6742**	**0.4769**	0.6643	0.4719	0.6658
Prostate	0.5101	0.6615	0.5431	0.6406	0.5067	0.6748	0.5261	**0.6903**	0.5164	0.6695	**0.5321**	0.6821
Skin	0.3429	0.6234	0.4354	0.6074	0.3610	0.6289	0.3547	0.6192	0.3661	0.6400	**0.4339**	**0.6565**
Stomach	**0.4726**	0.6886	0.4871	0.6529	0.4477	0.6944	0.4553	**0.7043**	0.4475	0.6918	0.4705	0.7022
Testis	0.4754	0.6890	0.4843	0.6435	0.4942	0.6869	0.4917	**0.7006**	0.5091	0.6883	**0.5127**	0.6955
Thyroid	0.4315	0.6983	0.5154	0.6692	0.4300	0.6962	0.4344	0.7094	0.4412	0.7035	**0.4519**	**0.7151**
Uterus	0.4393	0.6393	0.5068	0.6204	0.4480	0.6599	**0.4790**	0.6622	0.4737	0.6516	0.4737	**0.6625**
Average	0.4629	0.6596	0.5040	0.6377	0.4796	0.6692	0.4815	0.6767	0.4846	0.6696	**0.4980**	**0.6793**
SD	0.0076	0.0036	-	-	-	-	-	-	0.0503	0.0340	0.0413	0.0318

Net decoder excelled in neoplastic, connective, and epithelial nuclei. However, in the case of inflammatory and connective nuclei, they were outperformed by TSFD-Net due to their larger training dataset (80/20 split vs. 33/67 split). In particular, all models consistently yielded the lowest results for dead cells, attributed to class imbalance and the small spatial extent of dead cells. To further analyze the influence of the Focal Tversky loss and our custom oversampling strategy, we included PQ values for the $\text{CellViT}_{\text{HIPT-256}}$ model (HoVer-Net decoder) with different regularization techniques. The segmentation quality was improved by oversampling ($\text{CellViT}_{\text{HIPT-256}}$-Over) for almost all nuclei classes except neoplastic nuclei. The deterioration of neoplastic nuclei is attributed to class rebalancing, as neoplastic nuclei constitute the majority class in the dataset. Removing the Focal Tversky loss ($\text{CellViT}_{\text{HIPT-256}}$-No-FC) led to a decrease in panoptic quality for all classes, except neoplastic nuclei again.

Finally, we evaluated the segmentation performance of the $\text{CellViT}_{\text{HIPT-256}}$ and $\text{CellViT}_{\text{SAM-H}}$ against the best baseline models by computing the binary PQ (bPQ) and the more challenging multi-class PQ (mPQ) for each of the 19 tissue types of PanNuke. As baselines, we include the best HoVer-Net model by Graham et al. [95], TSFD-Net, and the original STARDIST and CPP-Net models with ResNet50

encoder [46]. For our detection experiments in Section 5.2.3.1, we retrained the baseline STARDIST model with the ResNet50 encoder. Since we were unable to reproduce the segmentation results reported by Chen et al. [46], we include the best-reported results in Table 5.3, while our results are provided in the Appendix in Table A.11. Our experimental results demonstrate that CPP-Net and STARDIST (with ResNet50 encoder) exhibited comparable bPQ values, whereas our CellViT models achieved superior mPQ. This is primarily attributed to the superior detection capabilities of our models, which impacts the mPQ value. The best average model was CellViT$_{\text{SAM-H}}$. Segmentation results per tissue for $0.50\,\mu\text{m/px}$ are given in the Appendix A.12. To provide a visual representation of the segmentation results, we include tissue-wise comparisons between the ground truth and the segmentation predictions of the CellViT$_{\text{SAM-H}}$ model in Figure 5.4. As observed in the lung example, the instance segmentation of dead cells poses a considerable challenge due to their small size.

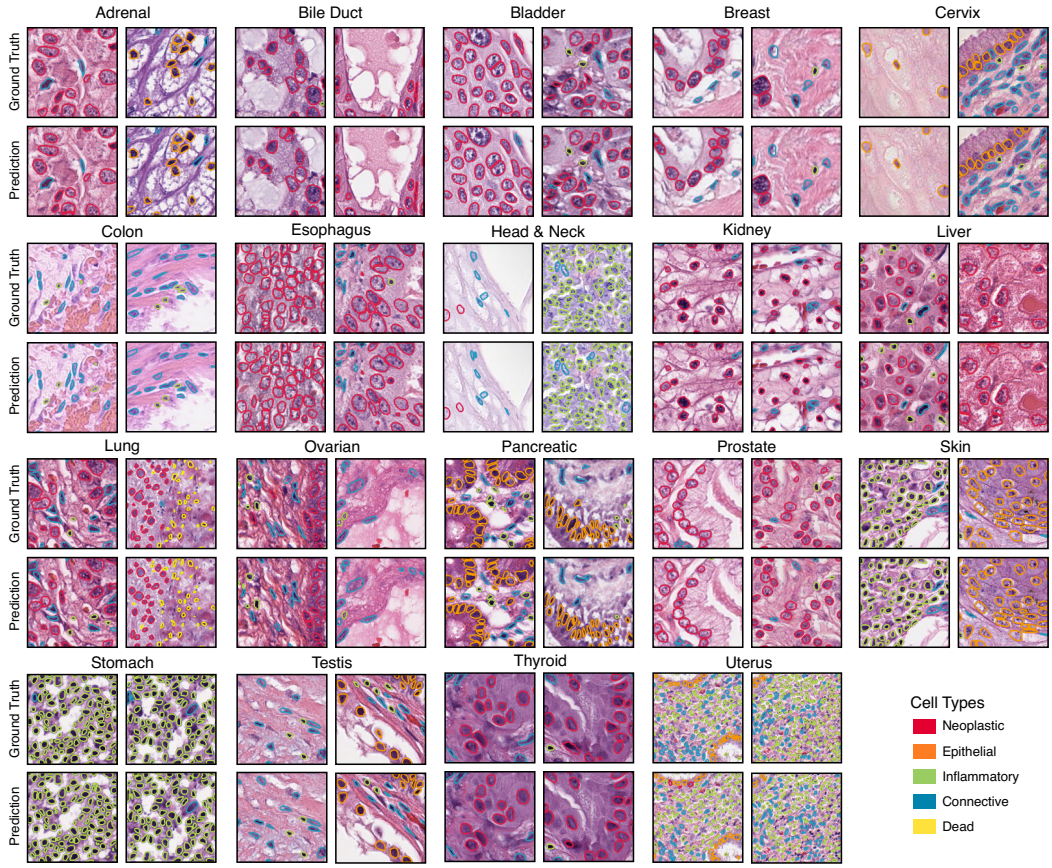


Figure 5.4: Example of PanNuke patches with ground truth annotations and CellViT$_{\text{SAM-H}}$ predictions overlaid for each tissue type. Adapted from [119].

5.2.3.3 MoNuSeg Test Performance

In this experiment, we focused on binary instance segmentation (single cell class) without retraining on the external MoNuSeg dataset to assess the generalizability of our models. Additionally, we aim to evaluate the impact of changing the input sequence size by performing inference on large-scale tiles of size $1{,}024\,\mathrm{px}$ $(0.25\,\mu\mathrm{m/px})$ and $512\,\mathrm{px}$ $(0.50\,\mu\mathrm{m/px})$, respectively, comparing the results to non-overlapping $256\,\mathrm{px}$ patches and $256\,\mathrm{px}$ patches with an overlap of $64\,\mathrm{px}$ derived by a shifting window approach. We utilized the three trained models of each PanNuke fold for every architecture and conducted inference on the MoNuSeg data without retraining.

The evaluation results are presented in Table 5.4. Consistent with previous experiments, the CellViT$_{\text{SAM-H}}$ model outperformed the CellViT$_{\text{HIPT-256}}$ model. When evaluated on $1{,}024\,\mathrm{px}$ tiles without patching, it reached a bPQ-score of 0.672, whereas for $256\,\mathrm{px}$ tiles with a $64\,\mathrm{px}$ overlap, the score was slightly below (0.671). However, when using $256\,\mathrm{px}$ patches without overlap, the bPQ-score decreased to 0.631, likely due to the absence of merging overlapping nuclei at cell borders and cells detected multiple times (higher recall). Importantly, the comparison between larger tiles and smaller tiles with overlapping indicates that inference on larger tiles did not lead to a degradation in performance. This justifies the strategy used for our inference pipeline for large-scale WSI, in which we are using $1{,}024\,\mathrm{px}$ sized patches with an overlap of $64\,\mathrm{px}$ and merging strategies.

Using the models trained with $0.50\,\mu\mathrm{m/px}$ data on the $0.25\,\mu\mathrm{m/px}$ data and vice versa, the $0.50\,\mu\mathrm{m/px}$ trained models exhibited poor performance on $0.25\,\mu\mathrm{m/px}$ data, while the $0.25\,\mu\mathrm{m/px}$ trained models experienced a less severe performance

Table 5.4: MoNuSeg validation result for CellViT$_{\text{HIPT-256}}$ and CellViT$_{\text{SAM-H}}$ models with HoVer-Net decoder and trained with CellViT hyperparameters on different dataset resolutions and inference patch sizes averaged over all three PanNuke training folds.
*Models trained on downscaled $0.50\,\mu\mathrm{m/px}$ PanNuke images. Adapted from [119].

Dataset resolution	Inference patch size	256 px with 64 px overlap				256 px without overlap				1,024 px (no patching)			
	Metric	$\overline{bPQ}$	$\overline{P}_{\mathrm{d}}$	$\overline{R}_{\mathrm{d}}$	$\overline{F_1}_{,\mathrm{d}}$	$\overline{bPQ}$	$\overline{P}_{\mathrm{d}}$	$\overline{R}_{\mathrm{d}}$	$\overline{F_1}_{,\mathrm{d}}$	bPQ	$\overline{P}_{\mathrm{d}}$	$\overline{R}_{\mathrm{d}}$	$\overline{F_1}_{,\mathrm{d}}$
0.25 µm/px (×40 mag.)	CellViT$_{\text{HIPT-256}}$	0.660	0.841	0.886	0.863	0.621	0.814	0.897	0.853	0.661	0.838	0.859	0.848
	CellViT$_{\text{SAM-H}}$	0.671	0.846	0.893	0.868	0.631	0.814	0.906	0.857	0.672	0.847	0.885	0.865
	CellViT$_{\text{HIPT-256}}$ (0.50 µm/px)*	0.509	0.748	0.893	0.804	0.491	0.728	0.895	0.792	0.515	0.759	0.905	0.813
	CellViT$_{\text{SAM-H}}$ (0.50 µm/px)*	0.524	0.746	0.963	0.840	0.514	0.729	0.963	0.829	0.540	0.749	0.966	0.842
		256 px with 64 px overlap				256 px without overlap				512 px (no patching)			
0.50 µm/px (×20 mag.)	CellViT$_{\text{HIPT-256}}$	0.588	0.918	0.766	0.834	0.586	0.902	0.759	0.824	0.593	0.919	0.771	0.837
	CellViT$_{\text{SAM-H}}$	0.627	0.922	0.791	0.851	0.620	0.908	0.784	0.841	0.627	0.909	0.792	0.846
	CellViT$_{\text{HIPT-256}}$ (0.50 µm/px)*	0.643	0.874	0.803	0.836	0.640	0.867	0.797	0.830	0.644	0.873	0.810	0.840
	CellViT$_{\text{SAM-H}}$ (0.50 µm/px)*	0.649	0.835	0.814	0.824	0.648	0.841	0.820	0.830	0.655	0.840	0.829	0.834

drop on the $0.50\,\mu m/px$ data. Nevertheless, networks trained and evaluated on the identical WSI resolution achieved the best performance. Thus, aligning the scanning resolution between different datasets and using the appropriate model is recommended. We include a visual demonstration presenting a tissue tile from the MoNuSeg test set along with binary segmentation masks generated by the CellViT$_{\text{SAM-H}}$ model in the Appendix (Figure A.2).

5.2.3.4 Embedding Analysis

Figure 5.5 shows the two-dimensional UMAP embeddings of the cells from the CoNSeP dataset. For this analysis, CellViT$_{\text{SAM-H}}$ and CellViT$_{\text{HIPT-256}}$ trained on the PanNuke dataset were used. The embeddings were obtained simultaneously during

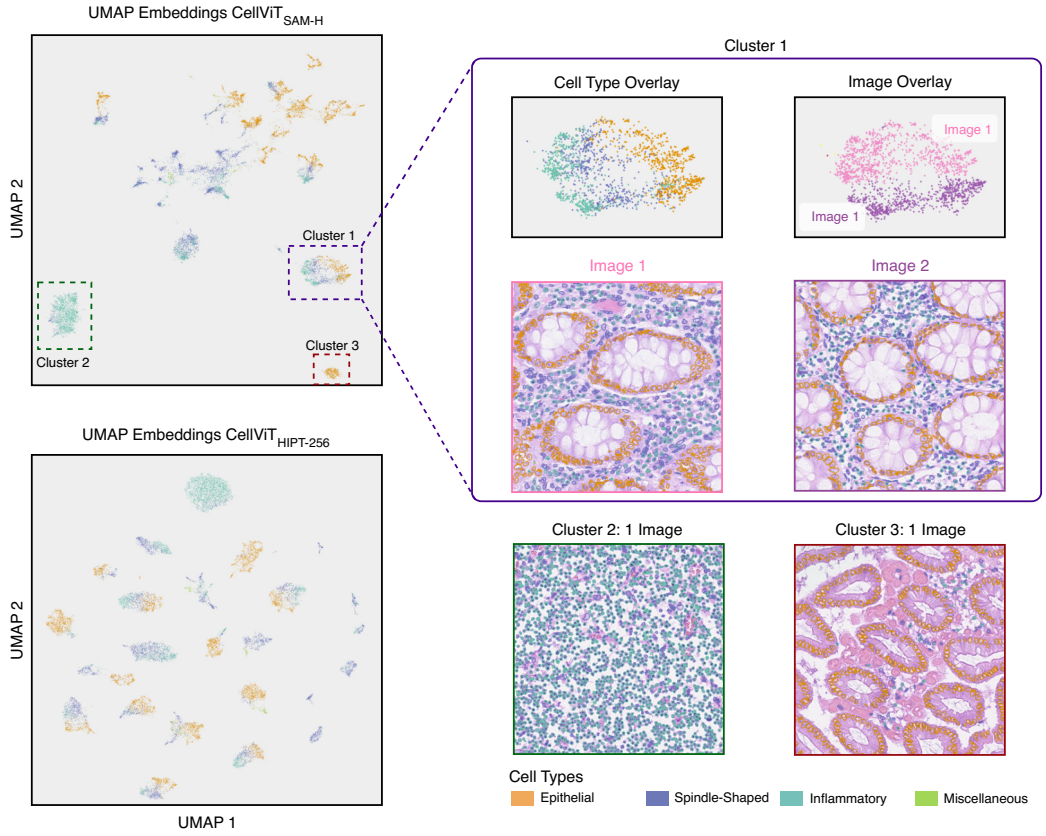


Figure 5.5: Two-dimensional UMAP embedding visualization (left) of the CoNSeP dataset with the CellViT$_{\text{SAM-H}}$ and CellViT$_{\text{HIPT-256}}$ (HoVer-Net encoder) models trained on PanNuke. We extract cell-tokens for each detected cell with our model, resulting in one embedding vector per cell. On the right side of the figure, representative clusters derived with the CellViT$_{\text{SAM-H}}$ model are displayed alongside corresponding tissue images. The color overlay illustrates the ground truth nuclei types within the dataset. Adapted from [119].

the cell detection within the inference pass. In the scatter plots (left) and tissue images (right), the color overlay corresponds to the nuclei classes. As suggested by Graham et al. [95], normal and malignant/dysplastic epithelial nuclei were grouped to epithelial nuclei, and fibroblast, muscle and endothelial nuclei to spindle-shaped nuclei. The scatter plot depicts that cells from different images and similar tissue phenotypes are within the same cluster.

Cluster 1 for the CellViT$_{\text{SAM-H}}$ model shows cell groups of two consists of cells from two images, each with several glands. Within this cluster, the local spatial distribution of cell embeddings allows for the differentiation of various nuclei types, such as epithelial, spindle-shaped, and inflammatory, even though the model was not specifically trained to differentiate them (e.g., spindle-shaped cells are not a category in the PanNuke dataset). Cluster 3, which is spatially close to cluster 1, contains even more glands, while the tissue image associated with the distant cluster 2 lacks glands, predominantly consisting of spindle-shaped and inflammatory nuclei. In conclusion, the global clusters capture mainly the differences in the nuclei's tissue environment (for example, the presence or absence of nearby glands or muscles), while the local arrangements capture the different types of nuclei.

Notably, for the CellViT$_{\text{HIPT-256}}$ model, we observed an increased emphasis on global tissue variations. We were able to effectively re-identify clusters 2 and 3 of the CellViT$_{\text{SAM-H}}$ model, but cluster 1 was split into two separate clusters.

To quantify the quality of the embeddings, we trained a linear nuclei classifier using the embeddings of the training dataset comprising $15{,}548$ nuclei to categorize the nuclei based on the CoNSeP classes. The classifier was then tested on samples from the test set ($8{,}773$ nuclei). On the test data, the classifier achieved an AUROC of 0.963 when using CellViT$_{\text{SAM-H}}$ embeddings and an AUROC of 0.960 when using CellViT$_{\text{HIPT-256}}$ embeddings. These results indicate that the ViT tokens (= nuclei embeddings) capture relevant nuclear features.

5.2.3.5 Inference Runtime

Our inference runtime benchmark demonstrated that the inference pipeline is accelerated by a factor of 2.49 (CellViT$_{\text{HIPT-256}}$) and 2.25 (CellViT$_{\text{SAM-H}}$) when using input patches of $1{,}024\,\text{px}$ as opposed to $256\,\text{px}$. The CellViT$_{\text{HIPT-256}}$ model operates 1.34 times faster than the CellViT$_{\text{SAM-H}}$ model, owing to the smaller ViT-structure of CellViT$_{\text{HIPT-256}}$ (ViT-Small with $21.7\,\text{M}$ parameters compared to ViT-Huge with $632\,\text{M}$ parameters).

Both CellViT models with their large $1{,}024\,\text{px}$ input patch size exceeded the HoVerNet model (input $256\,\text{px}$ with overlap, output $164\,\text{px}$), with speedups of 1.85 (CellViT$_{\text{HIPT-256}}$) and 1.39 (CellViT$_{\text{SAM-H}}$), respectively. Further results containing network parameter counts and number of multiply-accumulate operations (MACs) are given in Table A.13 in the Appendix. The observed speedup between HoVer-Net and our CellViT models can be attributed to the interaction of multiple compo-

nents, namely the reduced computational complexity for a $1{,}024\,\mathrm{px}$-sized FOV for the CellViT model (HoVer-Net $5{,}361.6\,\mathrm{M\,MACs}$, CellViT$_{\text{HIPT-256}}$ $2{,}127.9\,\mathrm{M\,MACs}$, CellViT$_{\text{SAM-H}}$ $3{,}413.4\,\mathrm{M\,MACs}$) and reduced preprocessing workload caused by larger patch sizes (fewer I/O operations). Simultaneously, a decrease in the number of merge operations for overlapping patches further contributed to efficiency.

5.2.4 Contribution

Based on the experimental results, we outline our contribution as follows:

Contribution 1: We present a novel U-Net-shaped encoder-decoder network for nuclei instance segmentation, leveraging Vision Transformers as encoder networks. The approach surpassed existing methods for nuclei detection by a substantial margin and achieved competitive segmentation results with other SOTA methods on the PanNuke dataset, which generalizes to other datasets (MoNuSeg, CoNSeP).

Contribution 2: We developed a framework that enables fast inference results applied on Gigapixel WSI by using an increased inference patch size of $1{,}024 \times 1{,}024\,\mathrm{px}$ in contrast to conventional $256\,\mathrm{px}$-sized patches. Compared to HoVer-Net, our inference pipeline runs 1.85 times faster.

Contribution 3: Our architecture inherently facilitates the extraction of cell embedding vectors, which correspond to the ViT token of the last Transformer layer spatially aligned to the predicted cell. Experiments on the CoNSeP dataset indicated that these embeddings capture relevant nuclear features.

The following Section 5.3 is built on the CellViT^{++} publication from 2025 [116].

5.3 CellViT++: Enhancing Cellular Analysis Capabilities

With the CellViT model, we already developed an approach with improved panoptic quality for nuclei segmentation in H&E tissue samples with reduced runtime compared to existing solutions. The model was trained on the PanNuke dataset, which includes 19 tissue types and approximately 190,000 annotated cells. However, this dataset is limited to the classes of neoplastic, inflammatory, epithelial, connective, and dead cells. Even though almost all cell types can be assigned to these taxonomic classes, this classification might be too coarse.

For example, the inflammatory cell category includes multiple immune cell types such as lymphocytes, neutrophils, eosinophils, and macrophages, with lymphocyte-ratios being a prognostic factor for breast cancer [174, 252]. Additionally, the neoplastic cells category can be partitioned further into malignant and benign cells, an important distinction for tumor detection. Hence, various cell categorization schemes may be necessary based on the type of tissue and medical question. A common method for training or adjusting models to fit a new classification system involves generating a dataset with numerous annotated cells (typically several thousand) by pathologists (e.g., see [95, 250, 277]). This method, however, is costly and time-consuming. Another method is just to rely on cell detections instead of fine-grained segmentation masks to reduce the annotation time [231]. Yet, this restricts network architectures to detection-based models, such that the widely-used HoVer-Net model could not be used in such settings.

To solve this problem, we take advantage of the CellViT model to calculate an embedding vector for each detected nuclei based on the Transformer tokens. In this Section, we explain how these embeddings can be used to adapt CellViT to new cell taxonomies. Based on the CellViT model pretrained on the PanNuke dataset, we introduce a lightweight classification head for new cell taxonomies. Remarkably, this concept does not necessitate to retrain or fine-tune the segmentation backbone of the network. The goal is to build a framework for nuclei segmentation in H&E stained tissue samples, similar to the nnU-Net for radiology [128]. Since we are extending the CellViT model introduced in the previous Section, we refer to the framework as **CellViT^{++}**. With the ongoing release of additional foundation models for CPath since the development of CellViT in 2023, we also integrate and evaluate new foundation models as part of this Section. We show the usability of our method on seven datasets, covering a broad spectrum of cell types and organs. Compared to existing nuclei segmentation approaches, CellViT^{++} achieved remarkable zero-shot segmentation performance and data-efficient cell-type classification. Furthermore,

we show that CellViT^{++} can leverage immunofluorescence staining to generate training datasets without the need for pathologist annotations. The automated dataset generation approach outperforms the performance of networks trained on manually labeled data, demonstrating its ability to create high-quality training datasets without expert annotations.

5.3.1 Enhancements and Methodological Innovations

Building on the CellViT architecture introduced in Section 5.2, our approach takes advantage of the representation learning capacity of the foundation models and the inherent structure of the Transformer architecture of the image encoder. CellViT calculates deep cell features (embeddings) and segmentation masks without adding computational load in the forward pass. Leveraging these cell embeddings to build segmentation-agnostic cell type classification modules allows the network to adapt to new cell types, bypassing the traditional requirement for separate cell cropping and feature extraction seen in two-stage models [121].

A central aspect of the CellViT^{++} framework is the robust segmentation performance achieved by the segmentation decoder, which allows us to focus solely on retraining the lightweight cell classification module to achieve precise cell classification and segmentation across different cell types. We extend the original CellViT model by:

1. Integrating recently released foundation models (2023-2024), including UNI, Virchow, and Virchow-2, alongside the previously incorporated HIPT-256 and Segment Anything models.

2. Extracting cell tokens for each detected cell.

3. Adding a lightweight cell classification module based on cell tokens to enable rapid adaptation to new classification schemes (taxonomies).

4. Optimizing the code to improve inference speed.

5.3.2 Methods

5.3.2.1 Network Architecture

An overview of the network architecture is given in Figure 5.6a. The core is the CellViT model, which provides nuclei segmentation. In contrast to the implementation suggested in Section 5.2.1.1, the nuclei type prediction branch is excluded. Instead, we make use of the binary instance segmentation. Along with the instance predictions, the CellViT model returns cell embedding vectors, which are forwarded to the newly added cell classification module.

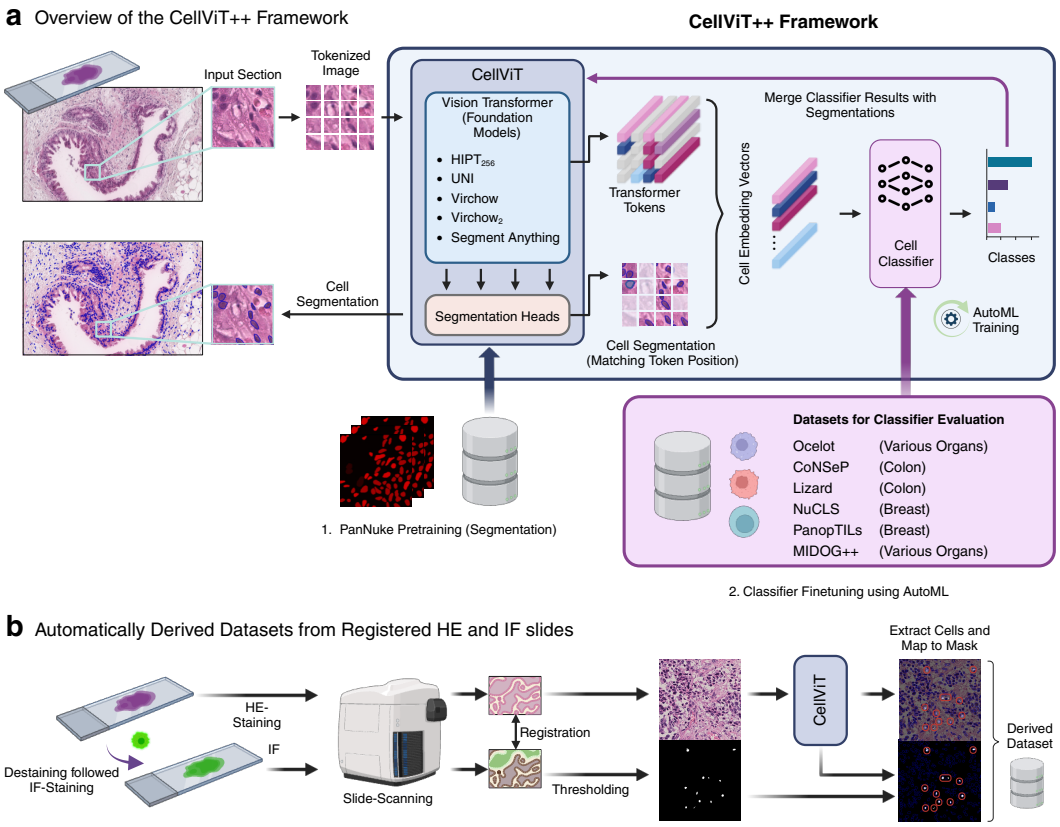


Figure 5.6: Overview of the CellViT^{++} Framework.
a) Network architecture with the new cell classification module utilizing cell embeddings derived from the Transformer tokens. Embeddings are extracted during segmentation and used to train the classification module for new cell types **b**) Pipeline to automatically derive labels from registered H&E and IF scans using CellViT^{++}, exemplified on the SegPath dataset. Adapted from [116]. Created with [57].

Token-Based Cell Classification Module Given that WSIs adhere to a fixed physical optical scale, a consistent mapping between the objects within the image and the corresponding ViT tokens can be established. Given an input resolution of $0.25\,\mu\text{m/px}$ of the images for the CellViT segmentation algorithm and a token size of $P = 14\,\text{px}$ or $P = 16\,\text{px}$, each token $\boldsymbol{x}_{\text{p}}^{i} \in \mathbb{R}^{3 \cdot P^2}$ corresponds to a squared section with approximately $4\,\mu\text{m}$ side length, which is in the range of the size of human cell nuclei [44]. Excluding the class token $\boldsymbol{x}^{\text{class}}$ and any potential register tokens, the token matrices $\boldsymbol{Z}_l$ of the respective Transformer layers l can be rearranged into a three-dimensional tensor $\boldsymbol{Z}_l \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times D}$, $l = 1 \ldots L$, with the first two dimensions corresponding to the spatial arrangement analogous to the input WSI sections. This spatial arrangement, coupled with the fact that the size of each token roughly

matches that of a nucleus, allows the assignment of tokens $\boldsymbol{z}_l^{\hat{y}_j} \in \mathbb{R}^D$ to each detected nucleus $\hat{y}_j$. Since each layer l further abstracts and enriches the tokens with additional information, we only assign the token $\boldsymbol{z}_L^{\hat{y}_j}$ of the last Transformer layer L to the prediction $\hat{y}_j$, thereby generating an embedding vector for each nucleus. Consequently, in the CellViT model, tokens from the last Transformer layer of the ViT encoder can be directly mapped to individual nuclei detected at the output of the segmentation head. This structure allows for efficient computation of deep nuclei features directly within the forward pass of the model. If a nucleus is associated with multiple tokens, we average all tokens in which the nucleus is located.

In contrast, existing two-stage models comprise a segmentation model and a separate feature extraction method. For this, visual crops of the segmented cells serve as inputs for a second model that either extracts classical image features, like by Huang et al. [121] or computes an embedding vector using another Deep Learning model. This two-stage approach introduces considerable overhead regarding computational time and memory, as the number of cells in a typical WSI can range from several hundred thousand to millions. By extracting cell embeddings jointly with the segmentation process, our method eliminates this overhead, as the tokens are inherently present within the ViT encoder.

We extend the model by a classifier module that takes the ViT tokens as input and predicts the nuclei classes. To train the classifier, we extract nuclei embeddings from annotated datasets and pair them with the ground truth nuclei classes as labels. For the classifier, we use a simple fully connected feedforward network with one hidden layer and a ReLu activation function (see Figure 5.6a). The classifier predictions can be merged with the binary segmentation masks. This approach enables us to adapt the CellViT network to new classification schemes. We denote this combination of segmentation models and the cell classification module as **CellViT^{++}**. To avoid negatively impacting downstream segmentation performance and for efficiency reasons, only the classifier is retrained based on the tokens and cell labels, while the segmentation network remains unchanged.

AutoML Training Trough Caching In order to train the classifier, the nuclei predictions and embeddings must first be calculated by performing inference with the CellViT model. If no data augmentation is used, the predictions, along with the embeddings, can be cached after the first training epoch. Since the initial extraction takes significantly longer than a training epoch of the classifier (small, fully connected feedforward network), the training time can be significantly reduced, allowing the classifier to be trained within just a few minutes. This efficiency enables us to conduct automated hyperparameter tuning for the classifiers, utilizing a Bayesian hyperparameter optimization across numerous training runs [279, 287].

Implementation Improvements Although all CellViT models were initially trained using input images of $256 \times 256\,\text{px}$, WSI-wise inference is performed on image sections of $1{,}024 \times 1{,}024\,\text{px}$ size with an overlap of $64\,\text{px}$. To accelerate throughput, we optimized the HoVer-Net postprocessing strategy explained in Section 5.2.1.4 to run on CUDA-enabled GPUs by utilizing libraries such as Numba [155], CuPY [209] and Ray [199]. This optimization enables efficient, independent parallel processing of each patch within a batch on GPUs. The final step involves merging overlapping cells, a process that incurs minimal computational overhead because of the relatively small number of overlapping cells compared to the total number of cells in a WSI. By optimizing the pipeline, we achieved a $40.38 \pm 4.77\%$ reduction in runtime on the same 10 test WSI used for the inference runtime experiments explained in Section 5.2.3.5.

5.3.2.2 Automated Cell Dataset Generation from IF Stainings

Manual annotation, particularly of segmentation masks, is costly and time-consuming and is a bottleneck for translational research. This limitation makes it impractical to create datasets at the necessary scale. Automatic generation of cell datasets from IF staining provides a possible solution to reduce the reliance on pathologists for fine-grained segmentation annotations.

Inspired by the idea of Komura et al. [146], we create an automated pipeline to generate cell-type specific datasets with limited human involvement. For this pipeline, tissue samples are stained with H&E and digitized using a slide scanner. These sections were then destained by alcohol and autoclave processing, and then IF stained using 4',6-diamidino-2-phenylindole dihydrochloride (DAPI) nuclear stain [146] to target specifically the cell types of interest. The slides were once more digitized after IF staining. After scanning the H&E and IHC/IF samples, both images are registered on cell-level and a binary mask of positively stained regions is created through thresholding of the IF channel. A CellViT^{++} model processes the H&E-stained images to extract cell segmentation masks. The masks can be re-mapped onto registered IF images for the detection of positive and negative cells. The entire dataset creation pipeline is illustrated in Figure 5.6b. It reduces the requirement for manual annotation to a minimum with just a very limited pathologist-approved validation set for verification of accuracy.

5.3.3 Experimental Setup

To assess the validity of the classification strategy, we demonstrate that incorporating the classification modules at the encoder's bottleneck layer L provides an effective method for adapting the model to new classes. We evaluate CellViT^{++} using different foundation models as image encoders to assess their ability to generate discriminative cell embeddings. All CellViT^{++} variants, respectively the segmentation models, have been pretrained on the PanNuke dataset (Figure 5.6a). To

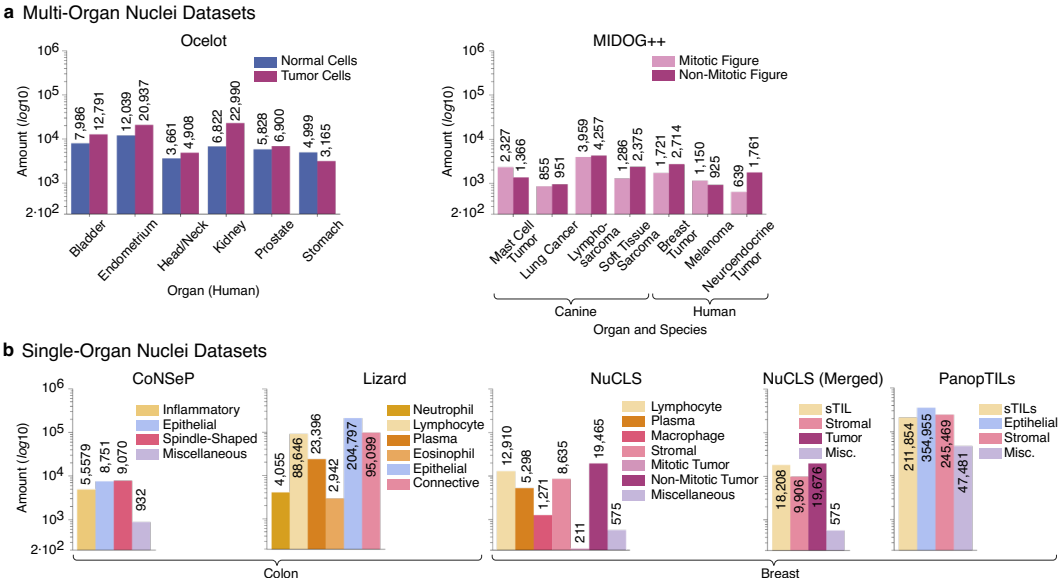


Figure 5.7: Dataset overview for the CellViT^{++} experiments.
a) Multi-organ nuclei datasets. **b**) Single-organ nuclei dataset for colon cancer and breast cancer. Adapted from [119]. Abbr.: Misc. (Miscellaneous)

test the cell classification module, we utilized multiple datasets, incorporating two multi-organ datasets (Ocelot [235], MIDOG++ [6]) that include organs absent from the PanNuke training set. Moreover, we evaluate the performance on two common cancer types: breast cancer (NuCLS [5], SegPath [146], PanopTILs [174]) and colorectal cancer (CoNSeP [95], Lizard [96]), benchmarking our method against current baseline methods on these datasets. An overview of all used datasets, along with nuclei amount per class, is given in Figure 5.7. Data efficiency and training time, including energy resources (CO_2 equivalent), are also considered for the comparison. As a key experiment, we demonstrate that combining registered H&E and IHC/IF stainings allows for the automatic generation of cell datasets, which can then be used as training datasets for the classifier module.

5.3.3.1 Datasets and Experiments

Cancer Cell Detection using Ocelot Although the PanNuke dataset encompasses a multitude of organs and includes neoplastic cells, differentiating malignant and benign cells within the neoplastic cell group is a key objective in routine histopathological diagnostics. The Ocelot dataset contains a total of 71,691 malignant tumor cells and 41,335 non-tumor cells annotated in 661 $1{,}024 \times 1{,}024\,\mathrm{px}$ ($0.20\,\mu\mathrm{m/px}$) image sections including six human organs, namely bladder, endometrium, head/neck, kidney, prostate, and stomach (see 5.7). Initially, we compare the performance of our method to baseline approaches on this dataset and evaluate the impact of data

augmentation on token classification by augmenting the input patches (Table A.37 in the Appendix). To obtain the mean and standard deviation, each experiment was repeated five times on the train and validation split (400 vs. 137 images), with evaluation on the official test set (124 images). Given the challenge of acquiring labeled WSI data and obtaining thousands of annotated cells with expert-level precision, we also investigated data efficiency. We sequentially sampled subsets of the training data, comprising 5%, 10%, 25%, 50%, and 75% and trained the models in these subsets.

Data-Efficient Learning for Colorectal Cancer As explained in Section 5.2.2.1, the CoNSeP dataset consists of 41 annotated tiles with a size of $1{,}000 \times 1{,}000\,\mathrm{px}$ from University Hospitals Coventry and Warwickshire, UK. It contains $24{,}332$ cells divided into a training ($15{,}555$ cells) and test set ($8{,}777$ cells). The cells are categorized into the four classes inflammatory cells, epithelial cells, spindleshaped cells, and miscellaneous cells. Following Graham et al. [95], we used the official split of 27 tiles for training and 14 for testing and performed a 5-fold CV on the training tiles to assess the mean model performance. We also simulated an active labeling approach to evaluate the training data efficiency by incrementally increasing the training tiles from 1 to 15, repeating each experiment five times without CV.

To ensure robustness for varying scanning resolutions, we validated CellViT^{++} using the Lizard dataset, which provides ROIs from colorectal cancer samples at $0.50\,\mu\mathrm{m/px}$.This dataset contains over $418{,}000$ cell segmentations in six categories (epithelium, lymphocytes, plasma cells, neutrophils, eosinophils, and connective tissue cells). As already shown with experiments on the MoNuSeg dataset in Section 5.2.3.3, directly processing images at $0.25\,\mu\mathrm{m/px}$ leads to distribution shifts and performance degradation. To address this, we resampled the images to 0.25 μm/px using a Lanczos filter, then processed them with CellViT^{++}, and finally downscaled the segmentation masks back to $0.50\,\mu\mathrm{m/px}$. We compare CellViT^{++} to SOTA segmentation models trained on $0.50\,\mu\mathrm{m/px}$ [95, 97, 300]. As the PanNuke samples of Lizard have been already included in the pretraining dataset, we excluded them in our analysis.

Training large models often requires substantial computational resources, resulting in considerable CO_2 emissions [217]. Our approach tackles this by leveraging pretrained domain-specific segmentation models, enabling computationally efficient fine-tuning. To substantiate this, we conducted a runtime comparison between HoVer-Net model training and CellViT^{++} fine-tuning model on the CoNSeP and Lizard datasets and compared the environmental impact using the metric from Lacoste et al. [153]. This comparison involved two setups: the baseline configurations described in the respective dataset publications (using two NVIDIA GeForce 1080 Ti GPUs) and our hardware setup (a single NVIDIA A100 80GB GPU).

Tumor Microenvironment Characterization in Breast Cancer Breast cancer is one of the most frequent cancers among females [285]. Two complementary datasets have been released for this tumor type: The NuCLS dataset (about $220{,}000$ nuclei, sample resolution $0.20\,\mu\mathrm{m/px}$) and the PanopTILs dataset ($859{,}759$ nuclei, sample resolution $0.25\,\mu\mathrm{m/px}$). The NuCLS dataset taxonomy divides the cells into tumor cells, stromal cells, and inflammatory cells. The PanopTILs dataset provides tumor-infiltrating lymphocytes (TILs), epithelial and stromal cells, and all remaining cells as categories. TILs are an important part of the immune response in the tumor microenvironment and a prognostic and predictive factor in immunotherapy in various carcinomas, including breast cancer [252]. Since these datasets are rather new, no network architectures beyond detection-based Mask R-CNN have been proposed. Therefore, we report the results of these datasets as a baseline for further research.

Automated Cell Dataset Generation from IF Stainings In this experiment, we evaluate the proposed automatic dataset generation method using two cohorts from the SegPath dataset [146], focusing on breast tissue. We employ the CD3/CD20 IF staining to identify lymphocytes and MIST1 staining to highlight plasma cells with registered H&E slides. Using different CellViT variants, we extracted more than $5{,}000$ lymphocyte cells from 220 H&E patches and $27{,}000$ plasma cells from $2{,}054$ patches within the SegPath training dataset for breast cancer, along with the tokens on which the classifiers can be trained. The performance of these classifiers was validated using the pathologist-approved subset of the NuCLS dataset in a one-vs.-all setting, distinguishing lymphocytes and plasma cells from other cell types. We also trained lymphocyte and plasma cell classifiers as baselines on the NuCLS training dataset.

Dealing with Low Prevalence when Detecting Mitotic Figures Previous experiments using the PanNuke dataset revealed that the recall for rare cell types such as dead cells is low. To further explore if this is a systematic drawback of our network or just specific for dead cells, we applied CellViT^{++} for mitosis detection using the MIDOG++ dataset [6]. Within this dataset, $11{,}937$ mitotic figures and $14{,}351$ hard negatives (non-mitotic figures that could falsely be recognized as mitotic figures) have been annotated across 503 tissue sections ($0.23 - 0.25\,\mu\mathrm{m/px}$). However, this dataset is only partially annotated such that non-mitotic cells are not annoated unless identified as hard negatives. In general, mitotic figures occur infrequently, and in this dataset account for only 0.16% of the cell population. We used CellViT$_{\text{SAM-H}}$ to extract and label all $7{,}398{,}795$ detected cells as mitotic or non-mitotic. To mitigate the problem of imbalanced classes, we trained CellViT$_{\text{SAM-H}}$ with mitotic-to-non-mitotic ratios of 1:1, 1:20, and 1:200 and performed stratified 5-fold cross-validation to report mean F_1 detection scores as suggested by the authors.

5.3.3.2 Evaluation Metrics

Nuclei Detection To assess the model's performance in cell detection, we apply the already defined metrics precision ($Prec$), recall (Rec), and F_1-score. In addition to the weighted definition ($F_{1,\text{d}}$, eq. (5.6), (5.9)) used for the PanNuke evaluation in Section 5.2.2 we add the calculation from the Ocelot benchmark (Ryu et al. [235]) as an additional score. This choice ensures consistency with published literature, as different metric definitions would otherwise hinder comparisons between studies. In line with the literature, the implementation by Ryu et al. [235] is denoted as mF_1-Score. The main difference between the two metrics is their averaging strategy. For mF_1, scores are first calculated separately for each cell class and then averaged across all classes. In contrast, the $F_{1,\text{d}}$ score treats the problem as a binary classification task and does not consider the nuclei class information. Furthermore, the mF_1 calculation does not employ the weighting used in both calculations in equations (5.6), (5.9). Another difference is the hit criterion that determines matches between ground truth nuclei and predictions($15\,\text{px}$ vs. $12\,\text{px}$-radius). More details on the algorithmic implementation can be found in the original publication [235].

Nuclei Instance Segmentation In accordance with equation eq. (5.12), the panoptic quality (bPQ, mPQ) is used to assess the panoptic segmentation quality. The mPQ score is calculated by calculating the PQ value for each image and each class calculating the average. This method may result in certain classes being omitted from the outcomes in cases where they were predicted but not present in image annotation, as described by Graham et al. [95]. To address this issue, they introduced the mPQ_+ metric, which calculates the metrics across all images first and then averages them by class [97]. This modification ensures that all classes are taken into account, regardless if the nuclei class was present in each sample. To avoid confusion, we explicitly specify whether the mPQ or mPQ_+ metrics are reported and follow the guidelines for each dataset to ensure consistency. Additional segmentation metrics, including the $DICE$-score and the average Jaccard index (AJI), as defined by Graham et al. [97], are also reported if appropriate [95, 97].

Carbon Footprint Calculation The carbon footprint and energy consumption estimations were calculated using the Machine Learning Impact calculator as presented in [153] (Machine Learning Impact calculator presented in [153]). All experiments were performed on a private infrastructure, with an estimated carbon efficiency of $0.432\,\text{kg}\,\text{CO}_2\,\text{eq}\,/\text{kWH}$ (OECD's 2014 yearly average [153]).

Table 5.5: CellViT-backbone comparison (mPQ) on the PanNuke dataset, split by cell-type. Average results of the official 3-fold CV split. Best results are marked bold, second best underlined. Adapted from [116].

Model	bPQ	mPQ	Neoplastic	Epithelial	Inflammatory	Connective	Dead
$\text{CellViT}_{\text{Random}}$	0.626	0.442	-	-	-	-	-
$\text{CellViT}_{\text{HIPT-256}}$	<u>0.670</u>	0.485	0.567	0.559	<u>0.405</u>	0.405	0.144
$\text{CellViT}_{\text{UNI}}$	0.664	0.492	0.573	0.579	0.403	0.408	<u>0.152</u>
$\text{CellViT}_{\text{Virchow}}$	0.665	0.489	0.577	<u>0.580</u>	0.393	0.409	0.147
$\text{CellViT}_{\text{Virchow-2}}$	0.665	<u>0.493</u>	<u>0.578</u>	<u>0.580</u>	0.403	<u>0.410</u>	**0.154**
$\text{CellViT}_{\text{SAM-H}}$	**0.679**	**0.498**	**0.581**	**0.583**	**0.417**	**0.423**	0.149

5.3.4 Results and Analysis

5.3.4.1 PanNuke Pretraining Results

As an extension to Section 5.2, we trained the CellViT model with the UNI, Virchow, and Virchow-2 encoders on the PanNuke dataset. The results, given in Table 5.5, show that all CellViT variants with a base model encoder exhibited superior performance compared to the base model $\text{CellViT}_{\text{Random}}$. Our subsequent analyses aim to identify the encoder that returns the most general cell representations. All subsequent CellViT segmentation models included in the CellViT^{++} framework were pretrained on 95% patches of the PanNuke dataset, with the remaining 5% used to detect overfitting.

5.3.4.2 Cancer Cell Detection Across Multiple Organs in the Ocelot Dataset

The metric utilized is the mF_1-Score. The dataset also includes area segmentation of tumor tissue, intended to serve as an additional aid for cell classification. In this thesis, only the subset that contains cell annotations was used. As demonstrated by Li et al. [166], the combination of CellViT with a tumor segmentation model (ensemble) currently yields SOTA results with a mF_1-score of 0.7243 on this dataset. Other comparative methods include a ResNet ensemble by Lafarge et al. [154] ($0.6617\,mF_1$), the FC-HarDNet model ($0.6992\,mF_1$) [175], and the model by Millward et al. [194] ($0.7221\,mF_1$). These models consist of at least two models for cell and tissue segmentation followed by a merging strategy to fuse tissue segmentations with cell segmentations. In contrast, the cell-only baseline reached $0.6444\,mF_1$ [235].

For comparison with our model, we used the CNN-based SoftCTM [241] model (tissue and cell), which has been externally validated and reported achieving a performance of 0.7172 mF_1-score. We selected this architecture because it does not rely on CellViT as a cell segmentation model. We retrained the SoftCTM model five times and report the average results along with the standard deviation (Figure 5.8a, 100% training data). On the external test set, SoftCTM achieved $0.7109 \pm 0.0069\,mF_1$, while the best CellViT^{++} variant ($\text{CellViT}^{++}_{\text{SAM-H}}$) achieved $0.6827 \pm 0.0028\,mF_1$. The

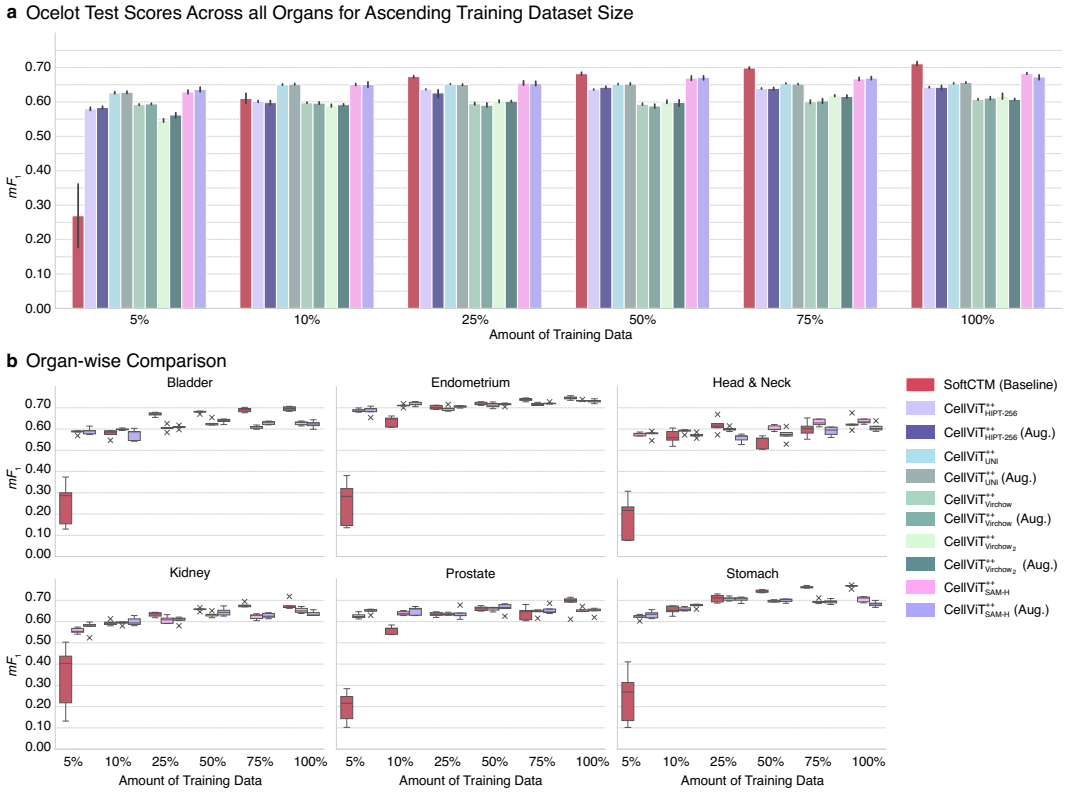


Figure 5.8: Ocelot results and comparison with state-of-the-art baseline network SoftCTM.
a) Mean F_1-Score averaged over all tissue types in the dataset on the official test set for multiple image encoders (foundation models), with models trained on limited data. Results show averages from 5 experiments with different seeds. **b**) Organ-wise detection results of the baseline SoftCTM model in comparison to the best performing CellViT^{++} model. Adapted from [116].

results are particularly noteworthy, as CellViT$^{++}_{\text{SAM-H}}$ outperformed the cell-only baseline ($0.6444\, mF_1$) by $\Delta = +0.0383\, mF_1$, even though only the cell classification module and not the segmentation model was retrained. Furthermore, as a cell-only model, CellViT$^{++}_{\text{SAM-H}}$ yielded a performance close to the supervised baseline of SoftCTM ($\Delta = -0.0282\ mF_1$), which utilized an additional tissue context image of the tumor microenvironment. In contrast, all models using histopathological foundation model encoders performed inferior to the CellViT$^{++}_{\text{SAM-H}}$ model. In addition, the results indicate that data augmentation had a negligible effect on classification performance.

The split by training data amount in Figures 5.8a and 5.8b demonstrates that all CellViT^{++} variants achieved superior performance with only 5% of the data compared to the reference method. At least 25% of the training data were necessary (equal to 18,573 cells) such that SoftCTM was on par. In contrast, the perfor-

mance of the CellViT^{++} models saturated, with a tripling of the training dataset size resulting in only a modest increase. The box plots in Figure 5.8b indicate that CellViT^{++} achieved better and more stable performance with limited training data, as the segmentation decoder serves as a robust backbone for cell detection. For underrepresented tissue types in the Ocelot dataset, such as head and neck, CellViT^{++} performed comparably or better than SoftCTM even with all available training data (e.g., Head/Neck: SoftCTM $0.6267 \pm 0.0268\, mF_1$ vs. CellViT$^{++}_{\text{SAM-H}}$ $0.6359 \pm 0.0095\, mF_1$).

Additionally, the comparison reveals that the HIPT-256, UNI, and SAM-H variants generated embeddings that allowed competitive classification performance, while the Virchow series models (Virchow, Virchow-2) consistently yielded inferior results on this dataset.

A detailed breakdown of all results is provided in the Tables A.21 and A.22 and the Appendix.

5.3.4.3 Data-Efficient Learning for Colorectal Cancer

Comparison on the CoNSeP Dataset We compare CellViT^{++} with the original HoVer-Net publication, a self-trained HoVer-Net model, and PointNu-Net [295], the current SOTA network on this dataset. Among all CellViT^{++} variants, the SAM-H model performed best, achieving an $mPQ+$ of 0.461 ± 0.014. The baseline models achieved 0.429 $mPQ+$, (HoVer-Net), and 0.446 $mPQ+$ (PointNu-Net). In terms of $mPQ+$, we set a new benchmark with CellViT$^{++}_{\text{SAM-H}}$.

In a zero-shot evaluation setting using models pretrained on the PanNuke dataset (binary setting), HoVer-Net achieved an $F_{1,\text{d}}$-score of 0.691 and a $DICE$-score of 0.802. In comparison, CellViT$^{++}_{\text{SAM-H}}$ reached an $F_{1,\text{d}}$-score of 0.772 and a $DICE$-score of 0.845, demonstrating remarkable zero-shot performance. A detailed overview of the results for all methods is provided in the Appendix (Tables A.25-A.26).

The results of the active labeling approach are presented in Figure 5.9a and 5.9b. Remarkably, using only three fully annotated tiles with approximately $2{,}500$ cells, CellViT^{++} achieved performance comparable to the PointNu-Net network (trained on 27 tiles), and with four tiles exceeded it (Figure 5.9a). The class-wise analysis presented in Figure 5.9b supports the hypothesis that the selection of training ROIs and data distribution is more important than the total amount of annotated nuclei. For example, increasing the number of annotated inflammatory cells beyond 750 showed minimal impact, and a similar plateau effect was observed for epithelial and spindle-shaped cells. In conclusion, we achieved superior performance in cell classification and noteworthy zero-shot segmentation on the CoNSeP dataset. Due to the excellent performance of CellViT$^{++}_{\text{HIPT}-256}$, CellViT$^{++}_{\text{UNI}}$, and CellViT$^{++}_{\text{SAM-H}}$, we focus our analysis on these networks.

Slide Resolution Generalizability - Lizard In this experiment, CellViT^{++} was evaluated on the Lizard dataset acquired at $0.50\,\mu m/px$ to test resolution generalizability. The best CellViT^{++} model (SAM-H) achieved a binary PQ score of 0.536 ± 0.009, which is below the performance of the comparison methods (HoVer-Net-Cerberus: 0.584 ± 0.014, HoVer-Net-Baseline: 0.624 ± 0.139, Cerberus: 0.612 ± 0.009, CGIS-CPF: 0.660 ± 0.009). Adding the token-based cell classification module yielded an mPQ score of 0.294 ± 0.002, approaching the performance of supervised models, which ranged from 0.295 ± 0.018 for HoVer-Net (Cerberus) to 0.421 ± 0.013 for CGIS-CPF.

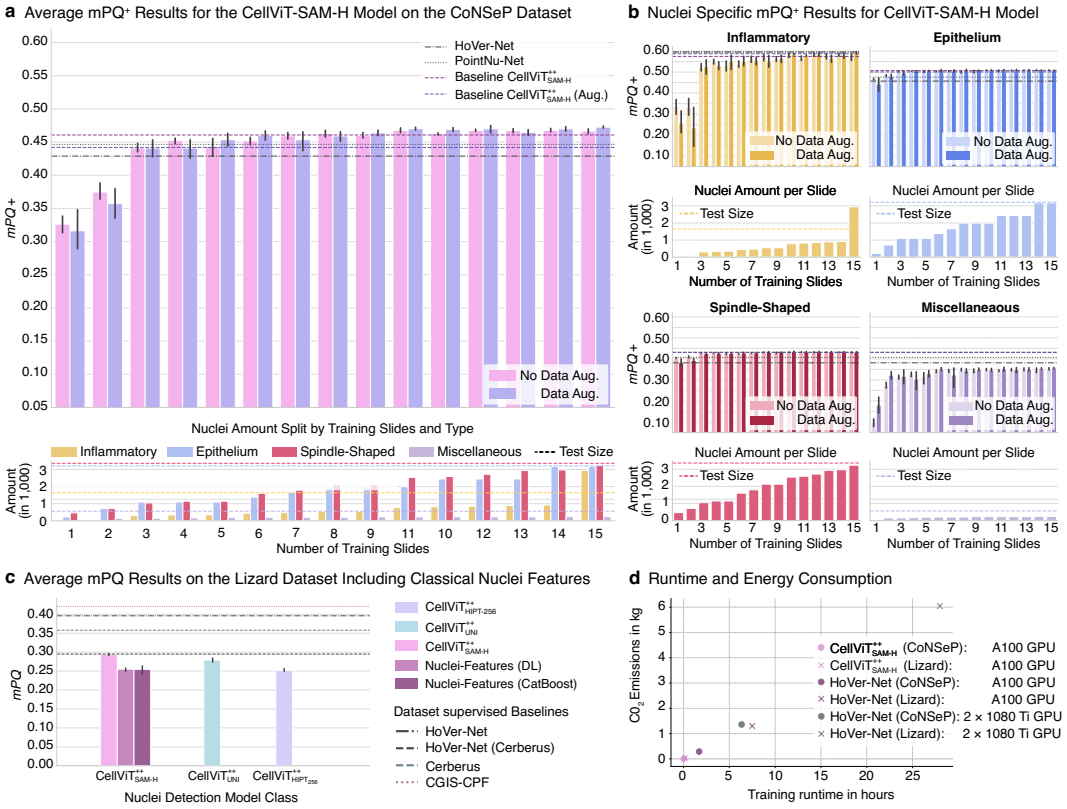


Figure 5.9: Experimental evaluation on colon tissue cell datasets.
a) Performance ($mPQ+$) on the CoNSeP dataset across varying training slide amounts. **b**) Nuclei-specific performance on CoNSeP. **c**) Average mPQ on the Lizard dataset compared to top-performing networks, with CellViT$^{++}_{\text{SAM-H}}$ evaluated using ViT token embeddings and classical nuclei features with DL and CatBoost classifiers. HoVer-Net Cerberus refers to the retrained version by cerberus. **d**) Runtime and energy consumption. Results in a) and b) are averages of 5 runs and baseline results are based on 5-fold cross-validation with test set evaluation, c) 3-fold CV; error bars indicate standard deviation (SD). Adapted from [116].

Comparison with Conventional Feature Engineeering Approaches on Lizard To address the performance gap observed between CellViT^{++} and supervised models on the Lizard dataset, we examined if the quality of our token embeddings contributed to the discrepancy. For this purpose, we compared the CellViT^{++} embeddings of HIPT-256, UNI, and SAM-H image encoder with conventional handcrafted nuclei features (histomics). For each nucleus identified by CellViT$^{++}_{\text{SAM-H}}$, we extracted 128 pre-defined nuclear features [121] features such as color, texture, shape, spatial, morphology, and orientation and trained a classifier similar to the token-based classification module. We also included classical machine learning algorithms such as SVM and CatBoost. The results are shown in Figure 5.9c. DL features achieved an mPQ score of 0.294 ± 0.002, while handcrafted features with a Deep Learning classifier reached 0.255 ± 0.002. The best classical machine learning model (CatBoost) using these handcrafted features achieved an mPQ of 0.255 ± 0.009, all lower than the deep cell features of CellViT$^{++}_{\text{SAM-H}}$ and CellViT$^{++}_{\text{UNI}}$ ($0.279 \pm 0.004\, mPQ$), but on par with CellViT$^{++}_{\text{HIPT}-256}$ ($0.252 \pm 0.003\, mPQ$). Further results are given in the Appendix (Table A.28).

Reducing CO_2 Emissions and Training Time The comparison of training runtime and energy consumption is presented in Figure 5.9d. Our CellViT^{++} model required only $81\,\text{s}$ for training on the CoNSeP ($9.23\,\text{WH}$) and $12\,\text{min}$ for training on the Lizard dataset ($92.16\,\text{WH}$) (fastest HoVer-Net approx. $102\,\text{min}$ CoNSeP and $450\,\text{min}$ Lizard). Even with a hyperparameter search involving 100 runs, our CO_2 footprint remained lower than for one training of the HoVer-Net network. This efficiency can be attributed to our caching mechanism for the first training run. Once the caching is complete, hyperparameter tuning becomes computationally inexpensive.

5.3.4.4 Characterization of the Tumor Microenvironment in Breast Cancer

Comparing mF_1, recall, and precision scores for the CellViT^{++} models on NuCLS reveals a sufficient average performance (Figure 5.10b). However, recall is lower for rare cell types, such as macrophages and mitotic tumor cells, while precision is higher. This suggests that although these cells are identified less frequently when detected, the detections are more accurate. In line with this, our models demonstrated strong performance on the PanopTILs dataset, achieving high F_1-scores for TILs (CellViT$^{++}_{\text{SAM-H}}$: $0.801 \pm 0.006\, mF_1$). Additionally, for epithelial cells, we achieved consistent mF_1-scores of 0.800 ± 0.003, and for stromal cells of 0.643 ± 0.006, as presented in Figure 5.10b.

5.3.4.5 Automated Cell Dataset Generation from IF Stainings

Using the breast tissue subset of the SegPath dataset (CD3/CD20 and MIST1), we extracted around $5{,}000$ lymphocytes and more than $27{,}000$ plasma cells from the SegPath dataset (Figure 5.10c). In comparison, the NuCLS training dataset

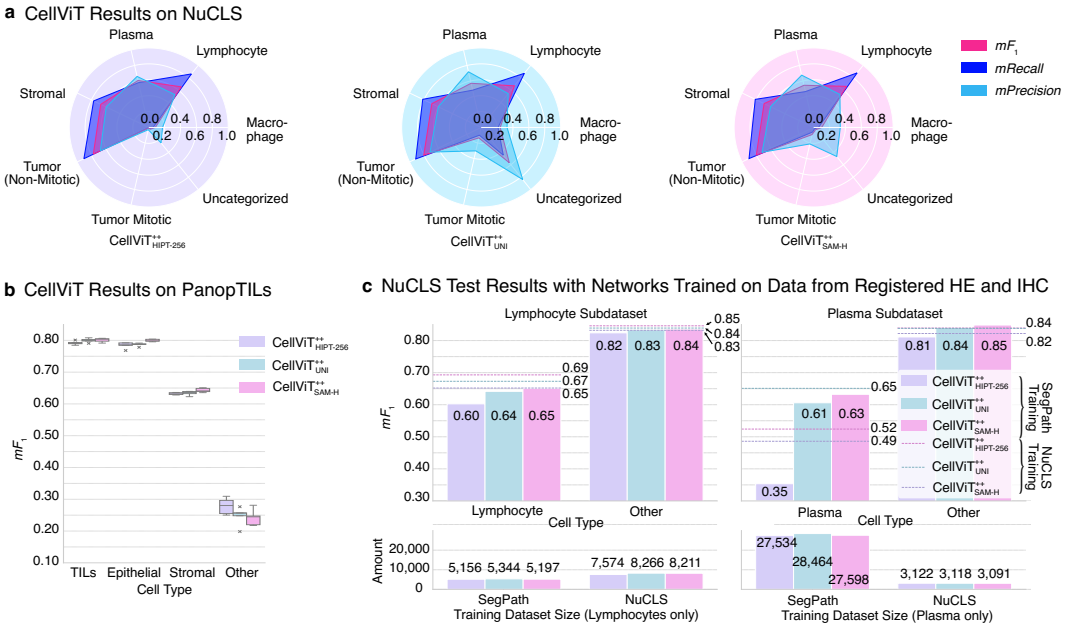


Figure 5.10: Experimental evaluation on breast cancer tissue.
a) Comparison of mF_1-score, precision, and recall for different CellViT^{++} models on the NuCLS dataset with all cell types included. **b**) CellViT^{++} performance on the PanopTILs dataset. **c**) Detection performance of our network on the NuCLS test set, trained with automatically derived labels from the SegPath dataset versus fully supervised training on the NuCLS training dataset for lymphocytes and plasma nuclei. The lower panel shows the number of training nuclei in both datasets. Adapted from [116].

containes approximately 7,500 lymphocytes and around 3,000 plasma cells. Trained on the automatically generated cell datasets, the CellViT$^{++}_{\text{SAM-H}}$ model achieved an mF_1-score of 0.651 for the detection of lymphocytes and 0.632 for plasma cells. In contrast, training on the NuCLS dataset (expert annotated) resulted in an mF_1-score of 0.693 (-0.042) for lymphocytes and 0.524 (+0.108) for plasma cells (Figure 5.10c). In particular, we trained a classifier for plasma cells using CellViT$^{++}_{\text{UNI}}$ tokens on the NuCLS data, achieving an F_1-score of 0.651, slightly surpassing the results obtained with our automatically generated dataset. On average, the performance of classification modules trained on the automatically generated SegPath cell dataset approached that of those trained on the expert-level annotated NuCLS datasets (lymphocytes) and, with a sufficient amount of annotated cell data, even exceeded it (plasma cells).

5.3.4.6 Dealing with Low Prevalence when Detecting Mitotic Figures

MIDOG++ results are presented in Table 5.6. Our findings indicate that the model with 200 additional non-mitotic cells per mitotic figure performs best. A detailed

Table 5.6: Results (F_1) on the MIDOG++ test dataset, categorized by organ and sample origin.
For the CellViT^{++} models, we also incorporated additional non-annotated cells to enhance the distinction between mitotic and non-mitotic cells during training. Specifically, 0, 20, or 200 (ratios 1:1, 1:20 and 1:200) additional non-mitotic figures were added per annotated mitotic figure. Adapted from [116].

Models		RetinaNet	CellViT$^{++}_{\text{SAM-H}}$		
Organs	Origin	Baseline	Ratio 1:1	Ratio 1:20	Ratio 1:200
Breast Cancer	Human	0.71 ± 0.02	0.50 ± 0.01	0.55 ± 0.01	0.60 ± 0.01
Neuroendocrine Tumor		0.59 ± 0.01	0.36 ± 0.08	0.45 ± 0.02	0.50 ± 0.00
Melanoma		0.81 ± 0.01	0.61 ± 0.09	0.67 ± 0.03	0.71 ± 0.02
Cutaneous Mast Cell	Canine	0.82 ± 0.01	0.63 ± 0.03	0.66 ± 0.02	0.70 ± 0.01
Lung Cancer		0.68 ± 0.02	0.34 ± 0.03	0.41 ± 0.01	0.43 ± 0.02
Lymphoma		0.73 ± 0.01	0.47 ± 0.04	0.51 ± 0.01	0.58 ± 0.01
Soft Tissue Sarcoma		0.69 ± 0.01	0.53 ± 0.04	0.53 ± 0.01	0.57 ± 0.02

analysis of the precision and recall scores (Tables A.35, A.36) confirms that for rare events such as mitosis, the recall remained lower than precision. On average, the model achieved a precision of 0.66 ± 0.14, higher than the recall of 0.54 ± 0.09. The performance also varies by tissue: for example, in human melanoma, the model reached a precision of 0.83 ± 0.03 and a recall of 0.62 ± 0.03, whereas in human breast cancer tissue, the precision was 0.77 ± 0.04 and the recall 0.49 ± 0.03. Despite the applicability of our approach, it falls short of the baseline detection model RetinaNet. This degradation occurs because even a small false positive rate or a small amount of missed mitotic figures can lead to a significant number of misclassifications when applied to millions of cells (remember: Mitotic figure prevalence in the dataset is estimated to be around 0.16%). A visualization is given in Figure A.7 in the Appendix.

5.4 Chapter Conclusion

The importance of segmenting nuclei in clinical contexts underlines the need for automated systems. In Section 5.2, we introduced a novel DL-based method to segment and detect nuclei in digitized H&E tissue samples simultaneously. This method was inspired by the success of previous works using large-scale pretrained Vision Transformers [45, 197, 270, 292, 303], particularly by the contributions made by Chen et al. [44] (HIPT-256) and Kirillov et al. [142] (SAM). Our proposed CellViT network achieved state-of-the-art performance in both nuclei instance segmentation and nuclei detection on the PanNuke dataset, outperforming existing methods in nuclei detection.

To extend the scope of the CellViT model, which nuclei classes are restricted to the five PanNuke classes, we propose the CellViT^{++} framework in Section 5.3. In this approach, deep cell features (embeddings) are directly extracted together with binary instance segmentation masks. These deep cell features are used to train cell classifiers for novel taxonomies without the need to retrain the segmentation model. Furthermore, CellViT++ does not rely on data augmentation, thus nuclear segmentations and embedding vectors can be cached after the first training epoch. This caching is done only once per dataset to accelerate the hyperparameter search. In order to assess the generalization capability of the CellViT^{++} framework, a set of experiments was carried out using heterogeneous datasets (origin, scanning setup, tissue types, taxonomies). In our experiment, we included the clinically important cancer types colorectal cancer and breast cancer, both of which have high incidence and mortality rates, posing substantial challenges to healthcare systems

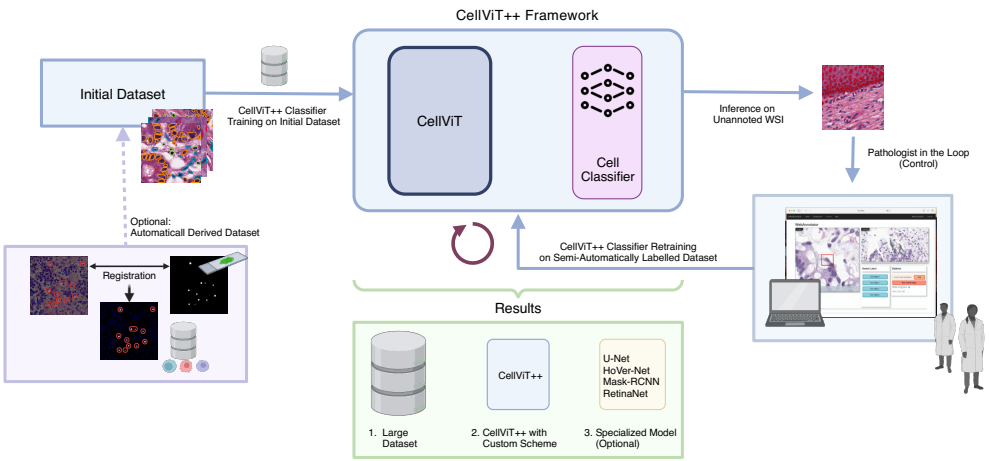


Figure 5.11: Suggested workflow for minimal human intervention training. Adapted from [116]. Created with [57].

worldwide [21, 80, 143, 285]. In general, CellViT^{++} demonstrated promising results well and excelled in scenarios with limited labeled data. Notably, the segmentation decoders of CellViT^{++} were not fine-tuned, which highlights its strong zero-shot segmentation performance. Using the SegPath dataset, we further showed that the training data for the classifier can be derived automatically. When evaluated on a separate dataset, the method's performance was on a par with, or even better than that of networks trained on manually labelled datasets.

However, there is still room for improvement. One aspect to enhance is the lack of a quality control (QC) module incorporated in our present pipeline. As outlined by Schömig-Markiefka et al. [242], the performance of segmentation models decreases when applied to real-world WSIs that might be affected by blurriness, scanning artifacts or tissue folding. This can be seens as a pipeline optimization problem instead of a model failure. Furthermore, CellViT^{++} is effective in low-data settings, but there are other models that can outperform it when provided with huge annotated datasets. The performance of the classification modules saturates after a few thousand samples, while dedicated instance segmentation networks that are entirely trained on these datasets can surpass our performance. However, the availability of such large datasets remains an obstacle, such that CellViT^{++} is suitable for most of the applications. In addition, it is important to keep in mind that the performance of CellViT^{++} may be reduced for rare cell types, such as mitotic figures (e.g., only 0.16% of all cells in the MIDOG++ dataset). In such cases, the classification module must achieve a high precision and recall next to high accuracy. An example helps to illustrate this. Suppose there are $1{,}000{,}000$ cells, of which $1{,}600$ belong to a rare cell class. Even a classifier with 99.9% accuracy, a recall of 87.5% and a precision of 63.6% would predict 800 cells as false positive

Based on our results, we suggest the following workflow for nuclei segmentation in H&E images with CellViT^{++} (Figure 5.11): First, collect a small set of nuclei annotations, either manually or automatically. Use this dataset to adapt CellViT^{++} and build a baseline model. Then, inference is performed on an unlabeled dataset, followed by active labeling in which a pathologist verifies and corrects the model's predictions [121]. These semi-automatic annotations increase the size of the training set, which can be used to train models like Cerberus [97] or RetinaNet [168] to determine if they perform better than CellViT^{++}.

In conclusion, CellViT and its improvement CellViT^{++} are a significant advance in digital pathology. Some initial publications by other research groups already have shown that these models provide a good performance in a variety of tasks [101, 129, 148, 166, 224]. For instance, Guo et al. [101] showed that CellViT yielded the best qualitative results on detecting cells kidney tissue samples, compared to STARDIST and CellPose. We outline our contributions as follows:

Contribution 1: We present a framework for panoptic nuclei segmentation in H&E tissue samples that can be easily adapted to new cell taxonomies.

Contribution 2: We provide detailed studies on nine datasets with varying properties, including differences in scanning devices and resolution, hospital origin, tissue type, and cell classification taxonomy. The results demonstrate the robustness of our framework in low-data scenarios and its ability to achieve competitive performance on external datasets. Our experiments verify the computational efficiency, saving annotation and training time.

Contribution 3: We introduce a workflow designed for the automated curation of nuclei datasets to train CellViT^{++}. This enables the examination of research hypotheses without a labor-intensive annotation process.

Part III

Conclusion and Future Directions

6 Discussion

This thesis advanced the research in computational pathology by providing new methods for automatic image segmentation. By taking inspiration from the current research streams from computer vision, such as Vision Transformers and foundation models, and applying them to computational pathology, we developed promising solutions for tissue and cell-level segmentation. The following Chapter summarizes the overall impact of this work and reflects on the open challenges.

The conclusions derived here are based on the respective original publications and set into the context of this thesis.[1]

6.1 Extending WSI Preprocessing

At the beginning of our research project, we prioritized the development of the preprocessing pipeline. We considered preprocessing as an essential component, particularly given the specifics of the data type (enormous data size, pyramid format). Existing solutions were not satisfactory, as they were either algorithm-specific and had to be adjusted for new algorithms, or introduced bottlenecks concerning data throughput. Our goal was to create a preprocessing library that is able to support several Deep Learning methods, including slide-level classification and multiscale image segmentation (Chapter 3).

[1] Section 6.1: BVM publication [117], Section 6.2: CMIG publication [76], Section 6.3: MedIA (CellViT) [119] and CellViT++ Pre-print [116]

We prioritized both offline training dataset preparation and online inference pipelines by providing PyTorch-compatible dataset interfaces. We used multiprocessing to gain a $78.36 \pm 6.63\%$ reduction in runtime compared to single-process libraries like TIAToolBox [222]. The resulting library, PathoPatcher, is open-source and was employed for the tissue and cell segmentation algorithms described in Chapters 4 and 5.

DICOM is the well-known standard in radiology for medical image format and data communication. In pathology, proprietary formats are still widely used, although the DICOM standard has already been adapted for digital pathology. For example, Leica already implemented the DICOM standard into their new GT450DX scanner and got FDA approval. Other file formats can be converted into DICOM leveraging tools like the WSIDICOMIZER by IMI-Bigpicture. We conducted several experiments to investigate the practical applicability of DICOM and identified areas for further improvements. Among the different DICOM implementations compared in this thesis (Leica, IMI-Bigpicture), the compatibility and processing speed differed. Thus, when it comes to speed, we recommend sticking to vendor-specific formats compatible with OpenSlide instead of enforcing DICOM. The flexibility of the DICOM standard causes multiple implementation patterns which results in incompatibilities between different manufacturers. Further investigation is required to understand the differences between the DICOM implementations of Leica and IMI-Bigpicture (WSIDICOMIZER). This includes examining file formats, data access patterns, and backend discrepancies. Overall, further research is required to address challenges in data harmonization. Besides, the practical integration of pathology data into PACS systems, commonly used in radiology, remains an open question.

Despite the versatility of our pipeline, some limitations remain. One possible improvement is the integration of more sophisticated quality control (QC) tools. Clinical routine data does not always have optimal quality. During the tissue preparation and scanning, image quality problems may arise caused by focus artifacts, pen markers on slides (especially for retrospective cohorts), tissue folding, dark spots, or the slide edge being scanned. Some of these artifacts are presented in Figure 6.1. Failing to detect these artifacts can lead to silent failure of AI algorithms [281]. It is, therefore, all the more important to integrate QC into our pipeline. Although several open-source tools have been developed in the past years, they all exhibit insufficient performance compared to commercial products, as noted by Weng et al. [281]. To address this, they recently published a CPath QC tool called GrandQC. We plan to extend our preprocessing pipeline using this open-source tool to detect common artifacts and improve QC within our clinical pipelines.

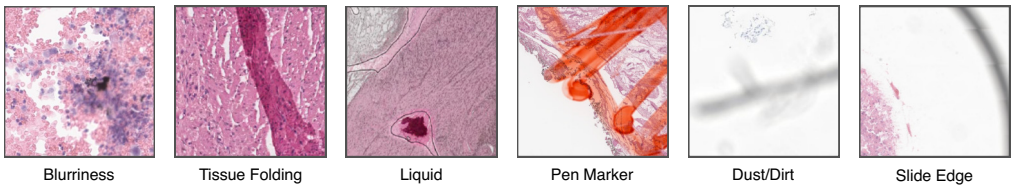


Figure 6.1: **Exemplary scanning artifacts.**

6.2 Potential of Deep Learning Techniques in Tissue Segmentation

Tissue segmentation is important to capture the tissue composition and draw conclusions concerning disease types, treatment response, and patient risk assessment. Applications range from detecting tumor areas as a diagnostic procedure to measuring tumor thickness as a prognostic factor in melanoma [245]. Furthermore, tissue segmentation in WSIs can reduce pathologists' cognitive workload by highlighting diagnostically relevant regions [36].

In Chapter 4, we proposed the **M**emory **A**ttention **F**ramework (MAF), which extends conventional encoder-decoder segmentation architectures, such as the U-Net or DeepLab, by an attention mechanism to query neighboring tissue context to incorporate spatial context during segmentation. Our experiments using three datasets with different tissue types demonstrated that our approach is superior to patch-based segmentation algorithms and outperforms other context-integrating algorithms, such as the msY-Net. Inspired by pathological slide evaluation, the MAF considers the tissue context for local tissue segmentation. Thus, the MAF avoids the trade-off between fine-granularity segmentation quality and the selection of a suitably large FOV required by traditional methods [130]. The advantages of the MAF are particularly evident in the segmentation of complex structures, as examined on the internal RCC datasets with ten tissue classes. Although the improvements were less pronounced on the CY16 and Paip 2019 datasets, we still observed notable performance gains. The method consistently reduced false positive tumor areas in healthy tissue.

One major challenge of our method is inherent to segmentation tasks: Annotating WSI is time-consuming and costly. To address this, active labeling or semisupervised learning could be incorporated to accelerate the annotation process and to identify tissue regions that should be annotated [131, 223, 227]. Another approach is to replace the CNN encoders used in this work with ViT-based basic models, e.g., the UNI and Virchow models used in Chapter 5, to investigate whether they also achieve satisfactory performance due to their extensive pretraining with less data. Our experiment on the internal pancreatic cancer dataset indicates that

in-domain pretraining for CNNs results in an increased performance. However, zero-shot segmentation models like the Segment Anything Model (SAM) currently do not achieve satisfactory results [60]. Nonetheless, they may still be practical for generating ground truth masks or as image encoders.

6.3 New Network Architectures for Cell Segmentation

Alongside tissue segmentation, panoptic nuclei segmentation, which refers to the simultaneous instance-wise segmentation of individual nuclei into semantic classes, is crucial for clinical applications. Manually identifying a large number of nuclei is time-consuming and prone to inter-observer variability such that there is a need for automatic solutions.

In Chapter 5, we introduced CellViT and the extension CellViT^{++} for automated panoptic nuclei segmentation. Our work was inspired by the success of previous works using large-scale trained vision transformers, specifically by the HIPT-256 and SAM models. First, we built the CellViT model. Using the public pan-cancer PanNuke benchmark dataset, we demonstrated that this network achieves SOTA performance in nuclei detection and instance segmentation. Histological and natural image pretraining yielded significantly better results than randomly initialized networks.

The CellViT model has been employed in several independent studies. For instance, in CD20+ B cell quantification for lung tissue, CellViT could identify nuclei in CD20 stained sections and revealed the relationship between B cell clusters and granulomas in mice with M. tuberculosis infection [148]. Li et al. [166] have achieved first place in the Ocelot Challenge for tumor cell detection by using a CellViT adaptation with an additional tissue segmentation branch. CellViT outperformed methods, including StarDist and CellPose, for acute lymphoblastic leukemia [224] and kidney sample analyses [101]. Jaume et al. [129] showed that CellViT is capable of combining molecular and morphological information by linking cell detections to spatial transcriptomics.

The mentioned external validations and our experiments on the MoNuSeg dataset confirm the model's ability to generalize to new cohorts and perform zero-shot nuclei segmentation. However, the type of tissue or tumor may differ and physicians or researchers could require a specific cell classification or taxonomy that differs from the high-level PanNuke taxonomy (e.g., TILs instead of all inflammatory cells). This means that models like CellViT and similar approaches, such as HoVer-Net, must be trained or fine-tuned on large, accurately annotated datasets and precise segmentation masks. Collecting these datasets is costly and time-consuming. To address this shortage, we propose the CellViT^{++} framework. The main idea is to introduce a light-weight cell classification module based on ViT tokens from the segmentation encoder. In this manner, instead of fine-tuning

the entire segmentation model, CellViT^{++} directly performs zero-shot binary nuclei segmentation and uses the classification head to assign class labels to each nucleus. The deep cell features (ViT tokens that are spatially connected to the detected cells) are directly extracted during the forward pass of images through the network, which adds no further computational cost. The classification module can be trained using either ground truth segmentation masks or simply cell detections without contour information to accelerate the annotation process.

To evaluate the performance of the proposed CellViT^{++} framework, we conducted experiments using datasets with varying scanning resolutions, tissue types, hospital origin, scanning devices, and cell types. In low-data settings, our approach yielded SOTA performance. Nevertheless, when there is a sufficient amount of labeled segmentation data available, for instance in the Lizard dataset with more than $418{,}000$ labeled nuclei, dedicated networks trained entirely on these datasets outperformed CellViT^{++}. This is an expected result since the CellViT^{++} model performs zero-shot segmentation without fine-tuning the segmentation branches. On the other hand, this has the advantage that the CellViT^{++} model can be trained for new datasets in a computationally and data-efficient manner. For instance, training CellViT^{++} on the Lizard dataset required just 12 minutes, compared to 7.5 hours for the HoVer-Net model and achieved comparable performance on the CoNSeP dataset to the current SOTA solution by just using approx. 20% of the training data. The reduced runtime contributes to a lower carbon footprint, significantly decreasing the environmental impact by more than 95%.

Building on these advancements, we still see the potential for further improvements. One such enhancement is the integration of a quality control (QC) module in our current pipeline, which we already addressed in Section 6.1 and is a current limitation of our preprocessing pipeline. In addition, the performance of the classification module is restricted by the token size. If two nuclei share a token and the two nuclei belong to different categories, then they cannot be distinguished. Also, further work is necessary to improve inference speed. Although the inference time of CellViT^{++} has been reduced by 40.39% compared to the original CellViT publication [116, 119], gigapixel WSIs still take 10-15 minutes to process which may be a problem in clinical settings.

In conclusion, CellViT^{++}, built on top of CellViT is another step forward in CPath. These models' integration into clinical practice is an opportunity not only to increase diagnostic accuracy but also to gain insights into tumor biology. We want to encourage the use of the framework by releasing the framework under an open-source license.

6.4 Merging Tissue and Cell Segmentation for Comprehensive Tumor Analysis

Quantitative analysis of histopathological samples will provide signifcant adavantages in enhancing diagnostic accuracy and identifying novel prognostic markers. Beyond obvious applications such as detecting abnormal regions within samples (e.g., tumor regions), more sophisticated patterns can also be uncovered. In this study, subject to further validation, we identified subTME structures that hold prognostic value for pancreatic cancer. In combination with cell segmentation algorithms, these findings may enable a more precise characterization of the tumor microenvironment.

As manual quantification of slides is time consuming and prone to high intra- and inter-observer variability [7, 238, 286], these tasks can now be performed automatically and, importantly, reproducible. The combination of WSI and DL algorithms for segmentation, classification, and detection helps histopathologists to obtain stable and quantitative results with minimum labor costs and improved diagnostic objectivity [164]. Although computational costs and algorithm runtimes remain considerable, they are still faster and more cost-effective than manual quantitative evaluation. While challenges persist, the field is progressing toward translating these automated approaches into clinical practice, with the first algorithms gaining FDA approval for clinical use [187, 299]. To establish trust in these methods, their robustness must be rigorously assessed, particularly concerning sensitivity to variations in staining protocols and imaging conditions.

As outlined in Chapter 1, WSI classification algorithms face challenges related to interpretability, often being either non-transparent or misleading in their explanations [133]. In clinical settings, despite their remarkable performance in identifying genetic alterations as prognostic factors [30], their primary value currently lies

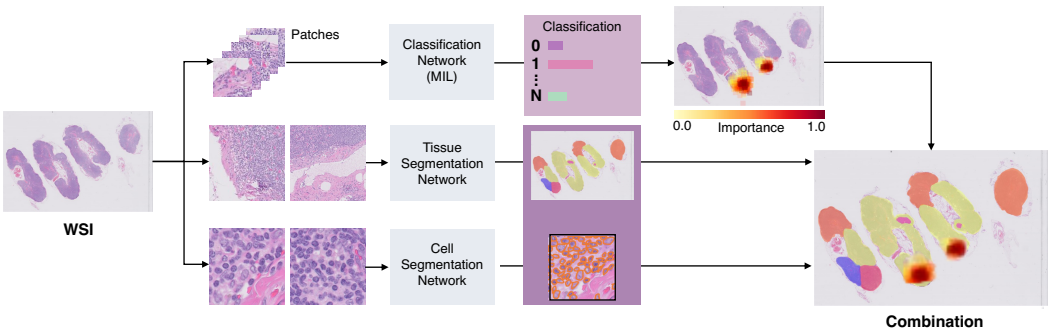


Figure 6.2: Combination of WSI classification and segmentation algorithms for an encompassing slide assessment.

in tasks such as slide triaging and workload reduction. As long as interpretability issues remain unresolved, their applications in identifying novel prognostic factors will be limited [133]. However, we see potential in integrating all three algorithm classes (classification, tissue segmentation, and cell detection) to provide a comprehensive analysis of histopathological slides. A vision for this integration is presented in Figure 6.2.

7 Outlook

"All models are wrong, but some are useful." - George E. P. Box

or

"Between the idea and the reality, between the motion and the act, falls the shadow." - Thomas S. Eliot

---

I could not decide which quote better fits this outlook. DL models offer great potential in computational pathology, but considerable challenges persist. The transition from algorithms to their application in clinical settings remains lengthy. The claims made in numerous publications must first be validated through independent studies, and real-world applicability needs to be rigorously tested. This Chapter provides a brief outlook of the current state and future research trends in CPath.

In this thesis, we developed novel segmentation algorithms for tissue (macro) and cell-level (micro) segmentation. We provide algorithmic solutions that allow the quantitative assessment of histological slides. Our work demonstrates promising performance across a range of public and private datasets, indicating the robustness and accuracy of the proposed methods in controlled environments. However, despite these encouraging results, real-world application remains an open challenge. The transition from algorithm development in controlled "sandbox" environments (prepared, cleaned datasets) to deployment in clinical settings is still pending. In practice, digital slides from multiple laboratories have a different appearance. Several aspects affect this, such as staining protocols, variations

in the scanning devices, and lacking QC. Although humans can compensate for these deviations, with algorithms, it often results in a reduction in performance. This is called a lack of domain generalization. To detect such limitations, comprehensive multi-center clinical studies are essential, as the real-world diversity of data is usually inadequately represented in benchmark datasets. The MIDOG++ dataset represents a notable exception, specifically designed to evaluate domain generalization with data from multiple laboratories employing different staining protocols and scanners; however, it remains an exception. Since our algorithms are all open-source, they have already been tested and used by other scientific institutions in research settings, with preliminary success [101, 129, 148, 166, 224]. We hope this external validation of our research will provide additional insights for improvements.

Digital pathology is a disruptive technology likely to transform laboratory workflows from slide acquisition to data management. Practical limitations still constrain the routine deployment of DL methods. However, they are expected to rise with the acquisition of slide scanners, ongoing standardization of image formats, and FDA-approved digital setups. In the near future, DL applications in pathology may take over classification tasks for case prioritization and routine checks, lowering the amount of manual slide assessment. Segmentation tools may serve as guidance systems to highlight specific ROI for the pathologist and help quantify the slide, such as measuring tissue and cellular composition. In particular, quantitative cellular analysis enables research that was previously impossible due to time constraints. Importantly, these technologies are designed to assist and not replace medical professionals, increasing productivity and addressing workforce shortages in pathology. Similarly to radiology, pathology will undergo a digital transformation in the coming years.

The combination of tissue and cell segmentation might uncover novel histological patterns to identify new biomarkers. The algorithms developed in this thesis provide the groundwork for this, and our initial experiments on our internal pancreatic cancer cohort have already yielded the first promising results. We hope that our algorithms provide insight into cancer mechanisms to improve patient care. We have many applications that we envision for our methods. We intend to apply the tissue- and cell-level segmentation models on two datasets: the pancreatic cancer dataset (Selocan) already introduced in Chapter 4, and an internal non-small cell lung cancer (NSCLC) dataset. To this end, we already performed initial experiments on the pancreatic cancer cohort using the tissue segmentation approach. Specifically, we revealed histological patterns in the stroma that correlate with patient survival. We will continue our research here to investigate tissue topologies and interactions further. One promising approach is the use of graph neural networks as they are capable of capturing the tissue architecture through

the node and edge definitions. The second cohort comprises approximately $4{,}320$ lung cancer patients with medical records. Keyl et al. [140] already developed a marker-based model that should be extended by histological parameters such as quantitative tissue composition. At the moment, the samples are scanned and the algorithm trained on publicly available training data [144]. The next step toward comprehensive tumor analysis is then to integrate tissue segmentations with cell segmentations. By combining these segmentations, we aim to uncover prognostic tissue interactions that may provide deeper insights into tumor behavior.

Some studies argue that the future of digital pathology and the application of Deep Learning will primarily rely on methods that leverage historical data from clinical systems and not on manual annotation [183, 293]. Specifically, the combination of slide images and diagnostic reports to automatically generate labels for training seems promising. While we do not dispute this hypothesis, our work demonstrates that, with our CellViT^{++} framework, it is possible to generate large-scale cell-level datasets with minimal human effort by combining H&E and IHC stainings. These advancements might potentially accelerate the creation of large-scale cell-level datasets while reducing the need for manual annotation, which can be used to extend our models for further clinical tasks. We are currently planning to apply this strategy to pancreatic cancer samples using multiplexed IHC, mass spectrometry imaging, or matrix-assisted laser desorption/ionization (MALDI) imaging in combination with H&E stainings [9], thereby extending the CellViT^{++} framework for the Selocan cohort.

A recent study by Komura et al. [145] reported that the number of publications in DL applied to histopathology has increased by more than eight times between 2018 (301 publications) and 2024 ($2{,}535$ publications). Next to the growth, also new research directions emerged, e.g., multimodal data fusion, which combines histology with laboratory values, radiological imaging, genetic data, and immunohistochemistry. Initial studies, including that by Lipkova et al. [170], suggest that multimodal data approaches hold promise as they incorporate all the patient information. Although the methods discussed in this thesis are limited to single-modality data (WSI), it is expected that they will be extended to multimodal fusion. Another promising development is the application of foundation models. Although the first models, HIPT-256, UNI, or Virchow, were used for cell segmentation in this thesis, their full potential has not yet been fully uncovered. Their domain generalization capability should be further investigated to see if they can improve the performance of algorithms in diverse multi-center trials. Further trends likely shaping the future of digital pathology are: The integration of expert models into agent systems, spatial transcriptomics, genomics-histology correlation, and 3D histological mapping as a new imaging modality.

Within this broader context, this work introduces novel segmentation algorithms for tissue- and cell-level analysis to enable the quantitative assessment of histological slides. Our algorithms achieved remarkable results across a diverse range of histopathological datasets, covering 23 tissue types and pathological conditions. First experiments on an internal pancreatic cancer cohort revealed that these algorithms have the potential to uncover tumor dynamics and prognostic tissue patterns. By making our algorithms available as open-source solutions, we aim to facilitate further validation and support the integration into clinical practice. With this work, we introduced novel DL-based methods to address key challenges in the field of histopathology.

References

References

[1] N. Abraham and N. M. Khan. "A novel focal tversky loss function with improved attention u-net for lesion segmentation". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 683–687. DOI: 10.1109/ISBI.2019.8759329.

[2] M. Ali. *PyCaret: an open source, low-code machine learning library in Python*. PyCaret version 1.0.0. Apr. 2020. URL: https://www.pycaret.org.

[3] S. Ali and A. Madabhushi. "An integrated region-, boundary-, shape-based active contour for multiple object overlap resolution in histological imagery". In: *IEEE Transactions on Medical Imaging* 31.7 (Apr. 2012), pp. 1448–1460. DOI: 10.1109/TMI.2012.2190089.

[4] M. Amgad et al. "Report on computational assessment of tumor infiltrating lymphocytes from the international immuno-oncology biomarker working group". In: *npj Breast Cancer* 6.1 (May 2020). ISSN: 2374-4677. DOI: 10.1038/s41523-020-0154-2.

[5] M. Amgad et al. "NuCLS: A scalable crowdsourcing approach and dataset for nucleus classification and segmentation in breast cancer". In: *GigaScience* 11 (2022). ISSN: 2047-217X. DOI: 10.1093/gigascience/giac037.

[6] M. Aubreville et al. *MIDOG++: a comprehensive multi-domain dataset for mitotic figure detection*. 2023. DOI: 10.6084/M9.FIGSHARE.C.6615571.V1.

[7] A. S. Azam et al. "Digital pathology for reporting histopathology samples, including cancer screening samples – definitive evidence from a multisite study". In: *Histopathology* 84.5 (Jan. 2024), pp. 847–862. ISSN: 1365-2559. DOI: 10.1111/his.15129. URL: http://dx.doi.org/10.1111/his.15129.

[8] R. Balestriero et al. *A cookbook of self-supervised learning*. 2023. DOI: 10.48550/ARXIV.2304.12210.

[9] B. Balluff, R. M. Heeren, and A. M. Race. "An overview of image registration for aligning mass spectrometry imaging with clinically relevant imaging modalities". In: *Journal of Mass Spectrometry and Advances in the Clinical Lab* 23 (Jan. 2022), pp. 26–38. ISSN: 2667-145X. DOI: 10.1016/j.jmsacl.2021.12.006. URL: http://dx.doi.org/10.1016/j.jmsacl.2021.12.006.

[10] P. Bankhead et al. “QuPath: open source software for digital pathology image analysis”. In: *Scientific Reports* 7.1 (Dec. 2017). ISSN: 2045-2322. DOI: 10.1038/s41598-017-17204-5.

[11] H. Bao, L. Dong, S. Piao, and F. Wei. *BEiT: BERT pre-training of image transformers*. 2021. DOI: 10.48550/ARXIV.2106.08254.

[12] A. Bardes, J. Ponce, and Y. Lecun. “VICReg: variance-invariance-covariance regularization for self-supervised learning”. In: *ICLR 2022-International Conference on Learning Representations*. 2022.

[13] P. Barmpoutis et al. “A digital pathology workflow for the segmentation and classification of gastric glands: Study of gastric atrophy and intestinal metaplasia cases”. In: *PLOS ONE* 17.12 (Dec. 2022), e0275232. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0275232.

[14] D. Bender and K. Sartipi. “HL7 FHIR: An Agile and RESTful approach to healthcare information exchange”. In: *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*. 2013, pp. 326–331. DOI: 10.1109/CBMS.2013.6627810.

[15] K. Bera, N. Braman, A. Gupta, V. Velcheti, and A. Madabhushi. “Predicting cancer outcomes with radiomics and artificial intelligence in radiology”. In: *Nature Reviews Clinical Oncology* 19.2 (Oct. 2021), pp. 132–146. ISSN: 1759-4782. DOI: 10.1038/s41571-021-00560-7.

[16] K. Bera, K. A. Schalper, D. L. Rimm, V. Velcheti, and A. Madabhushi. “Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology”. In: *Nature Reviews Clinical Oncology* 16.11 (Aug. 2019), pp. 703–715. ISSN: 1759-4782. DOI: 10.1038/s41571-019-0252-y.

[17] A. G. Berman, W. R. Orchard, M. Gehrung, and F. Markowetz. “SliDL: a toolbox for processing whole-slide images in deep learning”. In: *PLOS ONE* 18.8 (Aug. 2023), e0289499. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0289499.

[18] R. Bommasani et al. *On the opportunities and risks of foundation models*. 2021. DOI: 10.48550/ARXIV.2108.07258.

[19] S. P. Border and P. Sarder. “From what to why, the growing need for a focus shift toward explainability of ai in digital pathology”. In: *Frontiers in Physiology* 12 (Jan. 2022). ISSN: 1664-042X. DOI: 10.3389/fphys.2021.821217.

[20] G. Bradski. “The OpenCV library”. In: *Dr. Dobb’s Journal of Software Tools* (Nov. 2000).

[21] F. Bray et al. "Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries". In: *CA: A Cancer Journal for Clinicians* 74.3 (Apr. 2024), pp. 229–263. ISSN: 1542-4863. DOI: 10.3322/caac.21834.

[22] R. Brixtel et al. "Whole slide image quality in digital pathology: review and perspectives". In: *IEEE Access* 10 (Dec. 2022), pp. 131005–131035. ISSN: 2169-3536. DOI: 10.1109/access.2022.3227437.

[23] T. B. Brown et al. *Language models are few-shot learners*. 2020. DOI: 10.48550/ARXIV.2005.14165.

[24] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. "Albumentations: fast and flexible image augmentations". In: *Information* 11.2 (Feb. 2020), p. 125. DOI: 10.3390/info11020125.

[25] A. Bychkov and M. Schubert. *Constant demand, patchy supply*. https://thepathologist.com/outside-the-lab/constant-demand-patchy-supply. 2023. (Visited on 11/21/2024).

[26] C. J. Cai, S. Winter, D. Steiner, L. Wilcox, and M. Terry. "Hello AI: uncovering the onboarding needs of medical practitioners for human-ai collaborative decision-making". In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (Nov. 2019), pp. 1–24. ISSN: 2573-0142. DOI: 10.1145/3359206.

[27] Cambridge Dictionary. *Medicine - american dictionary*. https://dictionary.cambridge.org/. 2019. (Visited on 11/21/2024).

[28] G. Campanella, C. Vanderbilt, and T. Fuchs. "Computational pathology at health system scale–self-supervised foundation models from billions of images". In: *AAAI 2024 Spring Symposium on Clinical Foundation Models*. 2024.

[29] G. Campanella et al. "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images". In: *Nature Medicine* 25.8 (July 2019), pp. 1301–1309. ISSN: 1546-170X. DOI: 10.1038/s41591-019-0508-1.

[30] G. Campanella et al. *A clinical benchmark of public self-supervised pathology foundation models*. 2024. DOI: 10.48550/ARXIV.2407.06508.

[31] G. Campanella et al. *A clinical benchmark of public self-supervised pathology foundation models - leaderboard*. https://github.com/fuchs-lab-public/OPAL/tree/main/SSL_benchmarks. 2024. (Visited on 10/12/2024).

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. "End-to-end object detection with transformers". In: *Computer Vision – ECCV 2020*. Springer International Publishing, 2020, pp. 213–229. DOI: 10.1007/978-3-030-58452-8_13.

[33] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. "Deep clustering for unsupervised learning of visual features". In: *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 139–156. DOI: 10.1007/978-3-030-01264-9_9.

[34] M. Caron et al. "Emerging properties in self-supervised vision transformers". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021. DOI: 10.1109/iccv48922.2021.00951.

[35] L. Carter. "Report of the review of NHS pathology services in England". In: *Chaired by Lord Carter of Coles* (2006).

[36] L. Chan, M. Hosseini, C. Rowsell, K. Plataniotis, and S. Damaskinos. "HistoSegNet: semantic segmentation of histological tissue type in whole slide images". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019, pp. 10661–10670. DOI: 10.1109/iccv.2019.01076.

[37] T. H. Chan, F. J. Cendra, L. Ma, G. Yin, and L. Yu. "Histopathology whole slide image analysis with heterogeneous graph representation learning". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023, pp. 15661–15670. DOI: 10.1109/cvpr52729.2023.01503.

[38] H. Chen, X. Qi, L. Yu, and P. Heng. "DCAN: deep contour-aware networks for accurate gland segmentation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, June 2016, pp. 2487–2496. DOI: 10.1109/CVPR.2016.273.

[39] H. Chen, D. Li, and Z. Bar-Joseph. "SCS: cell segmentation for high-resolution spatial transcriptomics". In: *Nature Methods* 20.8 (July 2023), pp. 1237–1243. ISSN: 1548-7105. DOI: 10.1038/s41592-023-01939-3.

[40] J. Chen et al. *Transunet: transformers make strong encoders for medical image segmentation*. 2021. DOI: 10.48550/arXiv.2102.04306.

[41] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (Apr. 2018), pp. 834–848. ISSN: 2160-9292. DOI: 10.1109/tpami.2017.2699184.

[42] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. *Rethinking atrous convolution for semantic image segmentation*. 2017. DOI: 10.48550/ARXIV.1706.05587.

[43] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 833–851. DOI: 10.1007/978-3-030-01234-2_49.

[44] R. J. Chen et al. "Scaling Vision Transformers to gigapixel images via hierarchical self-supervised learning". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022, pp. 16123–16134. DOI: 10.1109/cvpr52688.2022.01567.

[45] R. J. Chen et al. "Towards a general-purpose foundation model for computational pathology". In: *Nature Medicine* 30.3 (Mar. 2024), pp. 850–862. ISSN: 1546-170X. DOI: 10.1038/s41591-024-02857-3.

[46] S. Chen, C. Ding, M. Liu, J. Cheng, and D. Tao. "CPP-Net: context-aware polygon proposal network for nucleus segmentation". In: *IEEE Transactions on Image Processing* 32 (Jan. 2023), pp. 980–994. ISSN: 1941-0042. DOI: 10.1109/TIP.2023.3237013.

[47] T. Chen and C. Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[48] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 1597–1607.

[49] X. Chen, H. Fan, R. Girshick, and K. He. *Improved baselines with momentum contrastive learning*. 2020. DOI: 10.48550/ARXIV.2003.04297.

[50] X. Chen and K. He. "Exploring simple siamese representation learning". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2021. DOI: 10.1109/cvpr46437.2021.01549.

[51] X. Chen, S. Xie, and K. He. "An empirical study of training self-supervised Vision Transformers". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021. DOI: 10.1109/iccv48922.2021.00950.

[52] J. Cheng and J. C. Rajapakse. "Segmentation of clustered nuclei with shape markers and marking function". In: *IEEE Transactions on Biomedical Engineering* 56.3 (Nov. 2009), pp. 741–748. DOI: 10.1109/TBME.2008.2008635.

[53] S. Cheng. *Artificial neural networks*. Lecture Notes, Ruhr University Bochum, Winter Semester 2020/2021. 2020.

[54] O. Ciga, T. Xu, and A. L. Martel. "Self supervised contrastive learning for digital histopathology". In: *Machine Learning with Applications* 7 (Mar. 2022), p. 100198. ISSN: 2666-8270. DOI: 10.1016/j.mlwa.2021.100198.

[55] D. Clunie. *PixelMed Java DICOM Toolkit*. PixelMed Publishing TM. https://www.pixelmed.com/dicomtoolkit.html. 2015.

[56] Common Crawl. *Common crawl*. https://commoncrawl.org. 2024. (Visited on 09/12/2024).

[57] Created in BioRender. *Hörst, F*. https://BioRender.com/5gl07vn. 2025.

[58] H. M. DeLisser, P. J. Newman, and S. M. Albelda. "Molecular and functional aspects of PECAM-1/CD31". In: *Immunology Today* 15.10 (Oct. 1994), pp. 490–495. ISSN: 0167-5699. DOI: 10.1016/0167-5699(94)90195-3.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: a large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2009. DOI: 10.1109/cvpr.2009.5206848.

[60] R. Deng et al. *Segment anything model (SAM) for digital pathology: assess zero-shot segmentation on whole slide imaging*. 2023. DOI: 10.48550/ARXIV.2304.04155.

[61] *Der körper des menschen: einführung in bau und funktion*. 18th ed. Stuttgart, Germany: Georg Thieme Verlag, 2020. DOI: 10.1055/b000000452.

[62] N. Detlefsen et al. "TorchMetrics - measuring reproducibility in PyTorch". In: *Journal of Open Source Software* 7.70 (Feb. 2022), p. 4101. ISSN: 2475-9066. DOI: 10.21105/joss.04101.

[63] Deutsche Röntgengesellschaft. *Dreidimensionale körperwelten - 50 jahre computertomographie*. https://www.drg.de/de-DE/10046/50-jahre-ct-teil-2/. 2024. (Visited on 11/25/2024).

[64] J. Dippel et al. *RudolfV: a foundation model by pathologists for pathologists*. 2024. DOI: 10.48550/ARXIV.2401.04079.

[65] S. Dooper et al. "Gigapixel end-to-end training using streaming and attention". In: *Medical Image Analysis* 88 (Aug. 2023), p. 102881. ISSN: 1361-8415. DOI: 10.1016/j.media.2023.102881.

[66] A. V. Dorogush, V. Ershov, and A. Gulin. *CatBoost: gradient boosting with categorical features support*. 2018. DOI: 10.48550/arXiv.1810.11363.

[67] A. Dosovitskiy et al. *An image is worth 16x16 words: Transformers for image recognition at scale*. 2020. DOI: 10.48550/ARXIV.2010.11929.

[68] M. S. Durkee, R. Abraham, M. R. Clark, and M. L. Giger. "Artificial intelligence and cellular segmentation in tissue microscopy images". In: *The American Journal of Pathology* 191.10 (Oct. 2021), pp. 1693–1701. ISSN: 0002-9440. DOI: 10.1016/j.ajpath.2021.05.022.

[69] M. Dusenberry, F. Hu, N. Jindal, and D. Eriksson. *Deep-histopath*. GitHub. https://github.com/CODAIT/deep-histopath. 2019.

[70] A. Echle, N. T. Rindtorff, T. J. Brinker, T. Luedde, A. T. Pearson, and J. N. Kather. "Deep learning in cancer pathology: a new generation of clinical biomarkers". In: *British Journal of Cancer* 124.4 (Nov. 2020), pp. 686–696. ISSN: 1532-1827. DOI: 10.1038/s41416-020-01122-x.

[71] B. Ehteshami Bejnordi et al. "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer". In: *JAMA* 318.22 (Dec. 2017), p. 2199. ISSN: 0098-7484. DOI: 10.1001/jama.2017.14585.

[72] O. S. M. El Nahhas et al. "From whole-slide image to biomarker prediction: end-to-end weakly supervised deep learning in computational pathology". In: *Nature Protocols* (Sept. 2024). ISSN: 1750-2799. DOI: 10.1038/s41596-024-01047-2.

[73] O. S. M. El Nahhas et al. "Regression-based deep-learning predicts molecular biomarkers from pathology slides". In: *Nature Communications* 15.1 (Feb. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-45589-1.

[74] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe. "Whitening for self-supervised representation learning". In: *International conference on machine learning*. PMLR. 2021, pp. 3015–3024.

[75] R. Escobar Díaz Guerrero, L. Carvalho, T. Bocklitz, J. Popp, and J. L. Oliveira. "Software tools and platforms in digital pathology: a review for clinicians and computer scientists". In: *Journal of Pathology Informatics* 13 (June 2022), p. 100103. ISSN: 2153-3539. DOI: 10.1016/j.jpi.2022.100103.

[76] O. Ester et al. "Valuing Vicinity: memory attention framework for context-based semantic segmentation in histopathology". In: *Computerized Medical Imaging and Graphics* 107 (July 2023), p. 102238. ISSN: 0895-6111. DOI: 10.1016/j.compmedimag.2023.102238.

[77] A. J. Evans et al. "US Food and Drug Administration approval of whole slide imaging for primary diagnosis: a key milestone is reached and new questions are raised". In: *Archives of Pathology and Laboratory Medicine* 142.11 (Apr. 2018), pp. 1383–1387. ISSN: 1543-2165. DOI: 10.5858/arpa.2017-0496-cp.

[78] T. Evans et al. "The explainability paradox: challenges for xai in digital pathology". In: *Future Generation Computer Systems* 133 (Aug. 2022), pp. 281–296. ISSN: 0167-739X. DOI: 10.1016/j.future.2022.03.009.

[79] G. Faa, M. Fraschini, and L. Barberini. "Reproducibility and explainability in digital pathology: the need to make black-box artificial intelligence systems more transparent". In: *Journal of Public Health Research* 13.4 (Oct. 2024). ISSN: 2279-9036. DOI: 10.1177/22799036241284898.

[80] J. Ferlay et al. *Global cancer observatory: cancer today*. Accessed: 2024-11-21. Lyon, France: International Agency for Research on Cancer, 2024. URL: https://gco.iarc.who.int/today.

[81] A. Filiot et al. “Scaling self-supervised learning for histopathology with masked image modeling”. In: *medRxiv* (July 2023). DOI: 10.1101/2023.07.21.23292757.

[82] S. Foersch et al. “Deep learning for diagnosis and survival prediction in soft tissue sarcoma”. In: *Annals of Oncology* 32.9 (Sept. 2021), pp. 1178–1187. ISSN: 0923-7534. DOI: 10.1016/j.annonc.2021.06.007.

[83] F. Fraggetta, S. Garozzo, G. F. Zannoni, L. Pantanowitz, and E. D. Rossi. “Routine digital pathology workflow: the catania experience”. In: *Journal of Pathology Informatics* 8.1 (Jan. 2017), p. 51. ISSN: 2153-3539. DOI: 10.4103/jpi.jpi_58_17.

[84] J. Gamper et al. *PanNuke dataset extension, insights and baselines*. 2020. DOI: 10.48550/arXiv.2003.10778.

[85] O. G. F. Geessink et al. “Computer aided quantification of intratumoral stroma yields an independent prognosticator in rectal cancer”. In: *Cellular Oncology* 42.3 (Mar. 2019), pp. 331–341. ISSN: 2211-3436. DOI: 10.1007/s13402-019-00429-z.

[86] J. George, E. Gkousis, A. Feast, S. Morris, J. Pollard, and J. Vohra. “Estimating the cost of growing the NHS cancer workforce in England by 2029”. In: (Oct. 2020).

[87] S. Gillies, C. van der Wel, J. Van den Bossche, M. W. Taves, J. Arnott, B. C. Ward, et al. *Shapely*. Zenodo. 2024. DOI: 10.5281/ZENODO.5597138.

[88] R. Girshick. “Fast R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Dec. 2015. DOI: 10.1109/ICCV.2015.169.

[89] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Region-based convolutional networks for accurate object detection and segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (Jan. 2016), pp. 142–158. ISSN: 2160-9292. DOI: 10.1109/tpami.2015.2437384.

[90] A. Goode, B. Gilbert, J. Harkes, D. Jukic, and M. Satyanarayanan. *About OpenSlide*. https://openslide.org/. 2024. (Visited on 11/12/2024).

[91] A. Goode, B. Gilbert, J. Harkes, D. Jukic, and M. Satyanarayanan. “OpenSlide: A vendor-neutral software foundation for digital pathology”. In: *Journal of pathology informatics* 4.1 (Aug. 2013), p. 27. DOI: 10.4103/2153-3539.119005.

[92] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT Press, 2016.

[93] Google Scholar. *nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation*. scholar.google.de. 2024. (Visited on 03/12/2024).

[94] G. Graf von Westphalen et al. *Histologie*. https://flexikon.doccheck.com/de/Histologie. 2024. (Visited on 11/12/2024).

[95] S. Graham et al. "Hover-Net: simultaneous segmentation and classification of nuclei in multi-tissue histology images". In: *Medical Image Analysis* 58 (Dec. 2019), p. 101563. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.101563.

[96] S. Graham et al. "Lizard: a large-scale dataset for colonic nuclear instance segmentation and classification". In: *Proceedings of the IEEE/CVF international conference on computer vision*. IEEE, 2021, pp. 684–693. DOI: 10.1109/iccvw54120.2021.00082.

[97] S. Graham et al. "One model is all you need: multi-task learning enables simultaneous histology image segmentation and classification". In: *Medical Image Analysis* 83 (Jan. 2023), p. 102685. ISSN: 1361-8415. DOI: 10.1016/j.media.2022.102685.

[98] J.-B. Grill et al. "Bootstrap your own latent - a new approach to self-supervised learning". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 21271–21284.

[99] B. T. Grünwald et al. "Spatially confined sub-tumor microenvironments in pancreatic cancer". In: *Cell* 184.22 (Oct. 2021), 5577–5592.e18. ISSN: 0092-8674. DOI: 10.1016/j.cell.2021.09.022.

[100] F. Gu, N. Burlutskiy, M. Andersson, and L. K. Wilén. "Multi-resolution networks for semantic segmentation in whole slide images". In: *Computational Pathology and Ophthalmic Medical Image Analysis*. Springer, 2018, pp. 11–18. DOI: 10.1007/978-3-030-00949-6_2.

[101] J. Guo et al. *Assessment of cell nuclei AI foundation models in kidney pathology*. 2024. DOI: 10.48550/ARXIV.2408.06381.

[102] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu. *Beyond self-attention: external attention using two linear layers for visual tasks*. 2021. DOI: 10.48550/arXiv.2105.02358.

[103] S. L. Gurina TS. *Histology, Staining.* 24th ed. Treasure Island, Florida, USA: StatPearls Publishing LLC., May 2023.

[104] N. Harder et al. "Automatic discovery of image-based signatures for ipilimumab response prediction in malignant melanoma". In: *Scientific Reports* 9.1 (May 2019). ISSN: 2045-2322. DOI: 10.1038/s41598-019-43525-8.

[105] C. R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[106] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu. "Swin UNETR: swin transformers for semantic segmentation of brain tumors in MRI images". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer International Publishing, 2022, pp. 272–284. DOI: 10.1007/978-3-031-08999-2_22.

[107] A. Hatamizadeh et al. “UNETR: transformers for 3D medical image segmentation”. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2022, pp. 1748–1758. DOI: 10.1109/WACV51458.2022.00181.

[108] J. Haubold et al. “BOA: a CT-based body and organ analysis for radiologists at the point of care”. In: *Investigative Radiology* 59.6 (Nov. 2023), pp. 433–441. ISSN: 0020-9996. DOI: 10.1097/rli.0000000000001040.

[109] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick. “Masked autoencoders are scalable vision learners”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022. DOI: 10.1109/cvpr52688.2022.01553.

[110] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. “Momentum contrast for unsupervised visual representation learning”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. DOI: 10.1109/cvpr42600.2020.00975.

[111] K. He, G. Gkioxari, P. Dollar, and R. Girshick. “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017. DOI: 10.1109/ICCV.2017.322.

[112] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. DOI: 10.1109/cvpr.2016.90.

[113] L. Henschel, S. Conjeti, S. Estrada, K. Diers, B. Fischl, and M. Reuter. “FastSurfer - A fast and accurate deep learning based neuroimaging pipeline”. In: *NeuroImage* 219 (Oct. 2020), p. 117012. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2020.117012.

[114] M. D. Herrmann et al. “Implementing the DICOM standard for digital pathology”. In: *Journal of Pathology Informatics* 9.1 (Jan. 2018), p. 37. ISSN: 2153-3539. DOI: 10.4103/jpi.jpi_42_18.

[115] M. Z. Hoque, A. Keskinarkaus, P. Nyberg, and T. Seppänen. “Stain normalization methods for histopathology image analysis: a comprehensive review and experimental comparison”. In: *Information Fusion* 102 (Feb. 2024), p. 101997. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2023.101997.

[116] F. Hörst, M. Rempe, H. Becker, L. Heine, J. Keyl, and J. Kleesiek. *CellViT++: energy-efficient and adaptive cell segmentation and classification using foundation models*. 2025. DOI: 10.48550/ARXIV.2501.05269.

[117] F. Hörst, S. H. Schaheer, G. Baldini, F. H. Bahnsen, J. Egger, and J. Kleesiek. "Accelerating artificial intelligence-based whole slide image analysis with an optimized preprocessing pipeline". In: *Bildverarbeitung für die Medizin 2024*. Springer Fachmedien Wiesbaden, 2024, pp. 356–361. DOI: 10.1007/978-3-658-44037-4_91.

[118] F. Hörst et al. "Histology-based prediction of therapy response to neoadjuvant chemotherapy for esophageal and esophagogastric junction adenocarcinomas using deep learning". In: *JCO Clinical Cancer Informatics* 7 (Aug. 2023). ISSN: 2473-4276. DOI: 10.1200/cci.23.00038.

[119] F. Hörst et al. "CellViT: Vision Transformers for precise cell segmentation and classification". In: *Medical Image Analysis* 94 (May 2024), p. 103143. ISSN: 1361-8415. DOI: 10.1016/j.media.2024.103143.

[120] S.-C. Huang, A. Pareek, M. Jensen, M. P. Lungren, S. Yeung, and A. S. Chaudhari. "Self-supervised learning for medical image classification: a systematic review and implementation guidelines". In: *npj Digital Medicine* 6.1 (Apr. 2023). ISSN: 2398-6352. DOI: 10.1038/s41746-023-00811-0.

[121] Z. Huang et al. "A pathologist-ai collaboration framework for enhancing diagnostic accuracies and efficiencies". en. In: *Nat. Biomed. Eng.* (June 2024). DOI: 10.1038/s41551-024-01223-5.

[122] P. Iakubovskii. *Segmentation models PyTorch*. GitHub. https://github.com/qubvel/segmentation_models.pytorch. 2019.

[123] M. Ilse, J. Tomczak, and M. Welling. "Attention-based deep multiple instance learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 2127–2136.

[124] T. Ilyas, Z. I. Mannan, A. Khan, S. Azam, H. Kim, and F. De Boer. "TSFD-Net: tissue specific feature distillation network for nuclei segmentation and classification". en. In: *Neural Networks* 151 (July 2022), pp. 1–15. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2022.02.020.

[125] IMI-Bigpciture. *WSIDICOMIZER*. GitHub. https://github.com/imi-bigpicture/wsidicomizer. 2024.

[126] IMI-Bigpciture. *WSIDICOMIZER*. GitHub. https://github.com/imi-bigpicture/wsidicom. 2024.

[127] Institut für Arbeitsmarkt- und Berufsforschung. "Durchschnittliche jährliche arbeitszeit pro erwerbstätigen (voll- Und teilzeit) in deutschland von 2001 bis 2023". In: *Statista* (Mar. 2024). URL: https://de.statista.com/statistik/daten/studie/4047/umfrage/entwicklung-der-jaehrlichen-arbeitszeit-pro-erwerbstaetigen/.

[128] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation". In: *Nature Methods* 18.2 (Dec. 2020), pp. 203–211. ISSN: 1548-7105. DOI: 10.1038/s41592-020-01008-z.

[129] G. Jaume et al. "HEST-1k: a dataset for spatial transcriptomics and histology image analysis". In: *Advances in Neural Information Processing Systems*. Dec. 2024.

[130] C. Jin, R. Tanno, M. Xu, T. Mertzanidou, and D. C. Alexander. "Foveation for segmentation of mega-pixel histology images". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 561–571. DOI: 10.1007/978-3-030-59722-1_54.

[131] X. Jin, H. An, J. Wang, K. Wen, and Z. Wu. "Reducing the annotation cost of whole slide histology images using active learning". In: *2021 3rd International Conference on Image Processing and Machine Vision (IPMV)*. IPMV 2021. ACM, May 2021, pp. 47–52. DOI: 10.1145/3469951.3469960.

[132] M. R. Junttila and F. J. de Sauvage. "Influence of tumour micro-environment heterogeneity on therapeutic response". In: *Nature* 501.7467 (Sept. 2013), pp. 346–354. ISSN: 1476-4687. DOI: 10.1038/nature12626.

[133] J. R. Kaczmarzyk, J. H. Saltz, and P. K. Koo. *Explainable ai for computational pathology identifies model limitations and tissue biomarkers*. 2024. DOI: 10.48550/ARXIV.2409.03080.

[134] J. R. Kaczmarzyk et al. "Open and reusable deep learning for pathology with WSInfer and QuPath". In: *npj Precision Oncology* 8.1 (Jan. 2024). ISSN: 2397-768X. DOI: 10.1038/s41698-024-00499-9.

[135] A. E. Kalyuzhny. "Primary antibodies". In: *Immunohistochemistry*. Springer International Publishing, 2016, pp. 3–9. DOI: 10.1007/978-3-319-30893-7_2.

[136] M. Kang, H. Song, S. Park, D. Yoo, and S. Pereira. "Benchmarking self-supervised learning on diverse pathology datasets". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023, pp. 3344–3354. DOI: 10.1109/cvpr52729.2023.00326.

[137] M. Kang, H. Song, S. Park, D. Yoo, and S. Pereira. "Benchmarking self-supervised learning on diverse pathology datasets". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023, pp. 3344–3354. DOI: 10.1109/CVPR52729.2023.00326.

[138] J. N. Kather et al. "Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer". In: *Nature Medicine* 25.7 (June 2019), pp. 1054–1056. ISSN: 1546-170X. DOI: 10.1038/s41591-019-0462-y.

[139] J. N. Kather et al. “Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study”. In: *PLOS Medicine* 16.1 (Jan. 2019), e1002730. ISSN: 1549-1676. DOI: 10.1371/journal.pmed.1002730.

[140] J. Keyl et al. “Decoding pan-cancer treatment outcomes using multimodal real-world data and explainable artificial intelligence”. In: *Nature Cancer* (Jan. 2025). ISSN: 2662-1347. DOI: 10.1038/s43018-024-00891-1.

[141] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar. “Panoptic segmentation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019. DOI: 10.1109/cvpr.2019.00963.

[142] A. Kirillov et al. “Segment anything”. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2023. DOI: 10.1109/iccv51070.2023.00371.

[143] L. Klimeck, T. Heisser, M. Hoffmeister, and H. Brenner. “Colorectal cancer: a health and economic problem”. In: *Best Practice & Research Clinical Gastroenterology* 66 (Oct. 2023), p. 101839. ISSN: 1521-6918. DOI: 10.1016/j.bpg.2023.101839.

[144] C. Kludt et al. “Next-generation lung cancer pathology: development and validation of diagnostic and prognostic algorithms”. In: *Cell Reports Medicine* 5.9 (Sept. 2024), p. 101697. ISSN: 2666-3791. DOI: 10.1016/j.xcrm.2024.101697.

[145] D. Komura, M. Ochi, and S. Ishikawa. “Machine learning methods for histopathological image analysis: updates in 2024”. In: *Computational and Structural Biotechnology Journal* 27 (2025), pp. 383–400. ISSN: 2001-0370. DOI: 10.1016/j.csbj.2024.12.033.

[146] D. Komura et al. “Restaining-based annotation for cancer histology segmentation to overcome annotation-related limitations among pathologists”. In: *Patterns* 4.2 (Feb. 2023), p. 100688. ISSN: 2666-3899. DOI: 10.1016/j.patter.2023.100688.

[147] N. A. Koohbanani, M. Jahanifar, A. Gooya, and N. Rajpoot. “Nuclear instance segmentation using a proposal-free spatially aware deep learning framework”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 622–630. DOI: 10.1007/978-3-030-32239-7_69.

[148] D. Koyuncu et al. “B cells in perivascular and peribronchiolar granuloma-associated lymphoid tissue and B-cell signatures identify asymptomatic mycobacterium tuberculosis lung infection in diversity outbred mice”. In: *Infection and Immunity* 92.7 (July 2024). ISSN: 1098-5522. DOI: 10.1128/iai.00263-23.

[149] 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), pp. 1550–1560. DOI: 10.1109/TMI.2017.2677499.

[150] N. Kumar et al. "A multi-organ nucleus segmentation challenge". In: *IEEE Transactions on Medical Imaging* 39.5 (May 2020), pp. 1380–1391. DOI: 10.1109/TMI.2019.2947628.

[151] T. Kurc et al. "Segmentation and classification in digital pathology for glioma research: challenges and deep learning approaches". In: *Frontiers in Neuroscience* 14 (Feb. 2020). ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00027.

[152] J. van der Laak, G. Litjens, and F. Ciompi. "Deep learning in histopathology: the path to the clinic". In: *Nature Medicine* 27.5 (May 2021), pp. 775–784. ISSN: 1546-170X. DOI: 10.1038/s41591-021-01343-4.

[153] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. *Quantifying the carbon emissions of machine learning*. 2019. DOI: 10.48550/arXiv.1910.09700.

[154] M. W. Lafarge and V. H. Koelzer. "Detecting cells in histopathology images with a ResNet ensemble model". In: *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology*. Springer Nature Switzerland, 2024, pp. 123–129. DOI: 10.1007/978-3-031-55088-1_11.

[155] S. K. Lam, A. Pitrou, and S. Seibert. "Numba: a llvm-based Python JIT compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. SC15. ACM, Nov. 2015. DOI: 10.1145/2833157.2833162.

[156] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

[157] H. H. Lee et al. *Foundation models for biomedical image segmentation: a survey*. 2024. DOI: 10.48550/ARXIV.2401.07654.

[158] Leica Biosystems. *Aperio image file types*. 2024. (Visited on 11/12/2024).

[159] J. J. Levy, L. A. Salas, B. C. Christensen, A. Sriharan, and L. J. Vaickus. "PathFlowAI: a high-throughput workflow for preprocessing, deep learning and interpretation in digital pathology". In: *Pac Symp Biocomput* 25 (2020), pp. 403–414.

[160] J. Levy, C. Haudenschild, C. Barwick, B. Christensen, and L. Vaickus. "Topological feature extraction and visualization of whole slide images using graph neural networks". In: *Biocomputing 2021*. WORLD SCIENTIFIC, Nov. 2020. DOI: 10.1142/9789811232701_0027.

[161] F. Li et al. "Deep learning-based predictive biomarker of pathological complete response to neoadjuvant chemotherapy from histological images in breast cancer". In: *Journal of Translational Medicine* 19.1 (Aug. 2021). ISSN: 1479-5876. DOI: 10.1186/s12967-021-03020-z.

[162] J. Li, K. V. Sarma, K. C. Ho, A. Gertych, B. S. Knudsen, and C. W. Arnold. "A multi-scale u-net for semantic segmentation of histological images from radical prostatectomies". In: *AMIA Annual Symposium Proceedings*. American Medical Informatics Association. 2017, p. 1140.

[163] J. Li et al. "A systematic collection of medical image datasets for deep learning". In: *ACM Computing Surveys* 56.5 (Nov. 2023), pp. 1–51. ISSN: 1557-7341. DOI: 10.1145/3615862.

[164] X. Li et al. "A comprehensive review of computer-aided whole-slide image analysis: from datasets to feature extraction, segmentation, classification and detection approaches". In: *Artificial Intelligence Review* 55.6 (Jan. 2022), pp. 4809–4878. ISSN: 1573-7462. DOI: 10.1007/s10462-021-10121-0. URL: http://dx.doi.org/10.1007/s10462-021-10121-0.

[165] Y. Li, J. Wu, and Q. Wu. "Classification of breast cancer histology images using multi-size and discriminative patches based on deep learning". In: *IEEE Access* 7 (Feb. 2019), pp. 21400–21408. DOI: 10.1109/ACCESS.2019.2898044.

[166] Z. Li, W. Li, H. Mai, T. Zhang, and Z. Xiong. "Enhancing cell detection in histopathology images: a ViT-based u-net approach". In: *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology*. Springer Nature Switzerland, 2024, pp. 150–160. DOI: 10.1007/978-3-031-55088-1_14.

[167] M. Liao et al. "Automatic segmentation for cell images based on bottleneck detection and ellipse fitting". In: *Neurocomputing* 173 (Jan. 2016), pp. 615–622. DOI: 10.1016/j.neucom.2015.08.006.

[168] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. "Focal loss for dense object detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (Feb. 2020), pp. 318–327. ISSN: 1939-3539. DOI: 10.1109/tpami.2018.2858826.

[169] M. Linkert et al. "Metadata matters: access to image data in the real world". In: *Journal of Cell Biology* 189.5 (May 2010), pp. 777–782. ISSN: 0021-9525. DOI: 10.1083/jcb.201004104.

[170] J. Lipkova et al. "Artificial intelligence for multimodal data integration in oncology". In: *Cancer Cell* 40.10 (Oct. 2022), pp. 1095–1110. ISSN: 1535-6108. DOI: 10.1016/j.ccell.2022.09.012.

[171] G. Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (Dec. 2017), pp. 60–88. ISSN: 1361-8415. DOI: 10.1016/j.media.2017.07.005.

[172] D. Liu, D. Zhang, Y. Song, H. Huang, and W. Cai. "Panoptic feature fusion net: a novel instance segmentation paradigm for biomedical and biological images". In: *IEEE Transactions on Image Processing* 30 (Jan. 2021), pp. 2045–2059. DOI: 10.1109/TIP.2021.3050668.

[173] P. Liu et al. "Software tools for 2D cell segmentation". In: *Cells* 13.4 (Feb. 2024), p. 352. ISSN: 2073-4409. DOI: 10.3390/cells13040352.

[174] S. Liu, M. Amgad, D. More, M. A. Rathore, R. Salgado, and L. A. D. Cooper. "A panoptic segmentation dataset and deep-learning approach for explainable scoring of tumor-infiltrating lymphocytes". In: *npj Breast Cancer* 10.1 (June 2024). ISSN: 2374-4677. DOI: 10.1038/s41523-024-00663-1.

[175] Y.-W. Lo and C.-H. Yang. "Enhancing cell detection via fc-hardnet and tissue segmentation: OCELOT 2023 challenge approach". In: *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology*. Springer Nature Switzerland, 2024, pp. 130–137. DOI: 10.1007/978-3-031-55088-1_12.

[176] M. Y. Lu, D. F. K. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood. "Data-efficient and weakly supervised computational pathology on whole-slide images". In: *Nature Biomedical Engineering* 5.6 (Mar. 2021), pp. 555–570. ISSN: 2157-846X. DOI: 10.1038/s41551-020-00682-w.

[177] M. Y. Lu, D. F. K. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood. "Data-efficient and weakly supervised computational pathology on whole-slide images". In: *Nature Biomedical Engineering* 5.6 (Mar. 2021), pp. 555–570. ISSN: 2157-846X. DOI: 10.1038/s41551-020-00682-w.

[178] W. Lu, S. Graham, M. Bilal, N. Rajpoot, and F. Minhas. "Capturing cellular topology in multi-gigapixel pathology images". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, June 2020, pp. 1049–1058. DOI: 10.1109/cvprw50498.2020.00138.

[179] M. Macenko et al. "A method for normalizing histology slides for quantitative analysis". In: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, June 2009, pp. 1107–1110. DOI: 10.1109/isbi.2009.5193250.

[180] T. Magadza and S. Viriri. "Deep learning for brain tumor segmentation: a survey of state-of-the-art". In: *Journal of Imaging* 7.2 (Jan. 2021), p. 19. ISSN: 2313-433X. DOI: 10.3390/jimaging7020019.

[181] N. Malpica et al. "Applying watershed algorithms to the segmentation of clustered nuclei". In: *Cytometry* 28.4 (Dec. 1998), pp. 289–297. DOI: 10.1002/(sici)1097-0320(19970801)28:4<289::aid-cyto3>3.0.co;2-7.

[182] A. Marcolini, N. Bussola, E. Arbitrio, M. Amgad, G. Jurman, and C. Furlanello. "histolab: a Python library for reproducible digital pathology preprocessing with automated testing". In: *SoftwareX* 20 (Dec. 2022), p. 101237. ISSN: 2352-7110. DOI: 10.1016/j.softx.2022.101237.

[183] N. Marini et al. "Unleashing the potential of digital pathology data by training computer-aided diagnosis models without human annotations". In: *npj Digital Medicine* 5.1 (July 2022). ISSN: 2398-6352. DOI: 10.1038/s41746-022-00635-4. URL: http://dx.doi.org/10.1038/s41746-022-00635-4.

[184] K. Martinez and J. Cupitt. "VIPS - a highly tuned image processing software architecture". In: *IEEE International Conference on Image Processing 2005*. IEEE, 2005, pp. II–574. DOI: 10.1109/icip.2005.1530120.

[185] D. Mason et al. *pydicom/pydicom: pydicom 3.0.1*. Zenodo. 2024. DOI: 10.5281/ZENODO.1291985.

[186] C. Matsoukas, J. F. Haslum, M. Sorkhei, M. Söderberg, and K. Smith. *Pretrained ViTs yield versatile representations for medical images*. 2023. DOI: 10.48550/ARXIV.2303.07034.

[187] C. McGenity et al. "Artificial intelligence in digital pathology: a systematic review and meta-analysis of diagnostic test accuracy". In: *npj Digital Medicine* 7.1 (May 2024). ISSN: 2398-6352. DOI: 10.1038/s41746-024-01106-8. URL: http://dx.doi.org/10.1038/s41746-024-01106-8.

[188] L. McInnes, J. Healy, N. Saul, and L. Großberger. "UMAP: uniform manifold approximation and projection". In: *Journal of Open Source Software* 3.29 (Sept. 2018), p. 861. ISSN: 2475-9066. DOI: 10.21105/joss.00861.

[189] S. Mehta, E. Mercan, J. Bartlett, D. Weaver, J. G. Elmore, and L. Shapiro. "Y-Net: joint segmentation and classification for diagnosis of breast biopsy images". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 893–901. DOI: 10.1007/978-3-030-00934-2_99.

[190] MIC-DKFZ. *nnUNet*. https://github.com/MIC-DKFZ/nnUNet. 2024. (Visited on 03/12/2024).

[191] N. Michielli et al. "Stain normalization in digital pathology: clinical multicenter evaluation of image quality". In: *Journal of Pathology Informatics* 13 (Sept. 2022), p. 100145. ISSN: 2153-3539. DOI: 10.1016/j.jpi.2022.100145.

[192] M. Milam and C. Koo. "The current status and future of FDA-approved artificial intelligence tools in chest radiology in the United States". In: *Clinical Radiology* 78.2 (Feb. 2023), pp. 115–122. ISSN: 0009-9260. DOI: 10.1016/j.crad.2022.08.135.

[193] P. Mildenberger, M. Eichelberg, and E. Martin. "Introduction to the DICOM standard". In: *European Radiology* 12.4 (Sept. 2001), pp. 920–927. ISSN: 1432-1084. DOI: 10.1007/s003300101100.

[194] J. Millward, Z. He, and A. Nibali. "Dense prediction of cell centroids using tissue context and cell refinement". In: *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology*. Springer Nature Switzerland, 2024, pp. 138–149. DOI: 10.1007/978-3-031-55088-1_13.

[195] H. Mohan. *Textbook of pathology*. 6th ed. New Delhi, India: Jaypee Brothers Medical, Feb. 2010.

[196] M. Moor et al. "Foundation models for generalist medical artificial intelligence". In: *Nature* 616.7956 (Apr. 2023), pp. 259–265. ISSN: 1476-4687. DOI: 10.1038/s41586-023-05881-4.

[197] M. Moor et al. "Foundation models for generalist medical artificial intelligence". In: *Nature* 616.7956 (Apr. 2023), pp. 259–265. ISSN: 1476-4687. DOI: 10.1038/s41586-023-05881-4. URL: http://dx.doi.org/10.1038/s41586-023-05881-4.

[198] I. Mori. "Current status of whole slide image (WSI) standardization in japan". In: *ACTA HISTOCHEMICA ET CYTOCHEMICA* 55.3 (June 2022), pp. 85–91. ISSN: 1347-5800. DOI: 10.1267/ahc.22-00009.

[199] P. Moritz et al. "Ray: A distributed framework for emerging ai applications". In: *13th USENIX symposium on operating systems design and implementation (OSDI 18)*. 2018, pp. 561–577.

[200] P. Moulin, K. Grünberg, E. Barale-Thomas, and J. v. der Laak. "IMI—Bigpicture: a central repository for digital pathology". In: *Toxicologic Pathology* 49.4 (Feb. 2021), pp. 711–713. ISSN: 1533-1601. DOI: 10.1177/0192623321989644.

[201] S. Mukhopadhyay et al. "Whole slide imaging versus microscopy for primary diagnosis in surgical pathology: a multicenter blinded randomized noninferiority study of 1992 cases (pivotal study)". In: *American Journal of Surgical Pathology* 42.1 (Jan. 2018), pp. 39–52. ISSN: 0147-5185. DOI: 10.1097/pas.0000000000000948.

[202] National Electrical Manufacturers Association (NEMA). *DICOM whole slide imaging (WSI)*. https://dicom.nema.org/dicom/dicomwsi/. 2022. (Visited on 11/12/2024).

[203] *National pathology programme - digital first: clinical transformation through pathology innovation*. London, England: NHS England, 2014.

[204] P. Naylor, M. Laé, 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 (Aug. 2019), pp. 448–459. DOI: 10.1109/TMI.2018.2865709.

[205] C. Neuner, S. Jabari, and S. Vilz. *WSI processing pipeline*. GitHub. https://github.com/FAU-DLM/wsi_processing_pipeline. 2023.

[206] M. K. K. Niazi, A. V. Parwani, and M. N. Gurcan. "Digital pathology and artificial intelligence". In: *The Lancet Oncology* 20.5 (May 2019), e253–e261. ISSN: 1470-2045. DOI: 10.1016/s1470-2045(19)30154-8.

[207] NVIDIA CORPORATION. *RAPIDS cuCIM*. GitHub. https://github.com/rapidsai/cucim. 2020.

[208] Okunator. *okunator/cellseg_models.pytorch: v0.1.23*. Zenodo. 2022. DOI: 10.5281/ZENODO.7064617.

[209] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis. "CuPy: a NumPy-compatible library for NVIDIA gpu calculations". In: *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*. 2017.

[210] Open Microscopy Environment. *Working with whole slide images*. https://docs.openmicroscopy.org/bio-formats/5.9.1/developers/wsi.html. 2018. (Visited on 11/12/2024).

[211] M. Oquab et al. *DINOv2: learning robust visual features without supervision*. 2023. DOI: 10.48550/ARXIV.2304.07193.

[212] N. Otsu. "A threshold selection method from gray-level histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (Jan. 1979), pp. 62–66. ISSN: 2168-2909. DOI: 10.1109/tsmc.1979.4310076.

[213] L. Pantanowitz, A. Sharma, A. B. Carter, T. Kurc, A. Sussman, and J. Saltz. "Twenty years of digital pathology: an overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives". In: *Journal of Pathology Informatics* 9.1 (Jan. 2018), p. 40. ISSN: 2153-3539. DOI: 10.4103/jpi.jpi_69_18.

[214] A. Paszke et al. *PyTorch: an imperative style, high-performance deep learning library*. 2019. DOI: 10.48550/ARXIV.1912.01703.

[215] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. "Context encoders: feature learning by inpainting". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. DOI: 10.1109/cvpr.2016.278.

[216] *Pathologie im fokus: aufgaben - herausforderungen - perspektiven*. Berlin, Germany: Deutsche Gesellschaft für Pathologie e.V., 2021.

[217] D. Patterson et al. *Carbon emissions and large neural network training*. 2021. DOI: 10.48550/ARXIV.2104.10350.

[218] S. Paul, B. Yener, and A. W. Lund. *C2P-GCN: cell-to-patch graph convolutional network for colorectal cancer grading*. 2024. DOI: 10.48550/ARXIV.2403.04962.

[219] A. M. Pavone et al. “Digital pathology: a comprehensive review of open-source histological segmentation software”. In: *BioMedInformatics* 4.1 (Jan. 2024), pp. 173–196. ISSN: 2673-7426. DOI: 10.3390/biomedinformatics4010012.

[220] F. Pedregosa et al. “Scikit-learn: machine learning in Python”. In: *Journal of Machine Learning Research* 12 (Oct. 2011), pp. 2825–2830. DOI: 10.5555/1953048.2078195.

[221] M. Pocevičiūtė, G. Eilertsen, and C. Lundström. “Survey of xai in digital pathology”. In: *Artificial Intelligence and Machine Learning for Digital Pathology*. Springer International Publishing, 2020, pp. 56–88. DOI: 10.1007/978-3-030-50402-1_4.

[222] J. Pocock et al. “TIAToolbox as an end-to-end library for advanced tissue image analytics”. In: *Communications Medicine* 2.1 (Sept. 2022), p. 120. ISSN: 2730-664X. DOI: 10.1038/s43856-022-00186-5.

[223] J. Qiu et al. “Abstract: adaptive region selection for active learning in whole slide image semantic segmentation”. In: *Bildverarbeitung für die Medizin 2024*. Springer Fachmedien Wiesbaden, 2024, pp. 11–11. DOI: 10.1007/978-3-658-44037-4_6.

[224] C. S. Raghaw, A. Sharma, S. Bansal, M. Z. U. Rehman, and N. Kumar. “CoT-CoNet: An optimized coupled transformer-convolutional network with an adaptive graph reconstruction for leukemia detection”. In: *Computers in Biology and Medicine* 179 (Sept. 2024), p. 108821. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2024.108821.

[225] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy. “Do Vision Transformers see like convolutional neural networks?” In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 12116–12128.

[226] V. Ramanathan, P. Pati, M. McNeil, and A. L. Martel. “Ensemble of prior-guided expert graph models for survival prediction in digital pathology”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Springer Nature Switzerland, 2024, pp. 262–272. DOI: 10.1007/978-3-031-72086-4_25.

[227] R. Rashmi, G. V. S. Sudhamsh, and S. Girisha. “A semi-supervised learning approach for tissue semantic segmentation in whole slide images”. In: *IEEE Access* 12 (2024), pp. 120482–120497. ISSN: 2169-3536. DOI: 10.1109/access.2024.3438568.

[228] J. Raufeisen et al. “Cyto R-CNN and CytoNuke Dataset: Towards reliable whole-cell segmentation in bright-field histological images”. In: *Computer Methods and Programs in Biomedicine* 252 (July 2024), p. 108215. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2024.108215.

[229] S. E. A. Raza et al. "Micro-Net: a unified model for segmentation of various objects in microscopy images". In: *Medical Image Analysis* 52 (Feb. 2019), pp. 160–173. DOI: 10.1016/j.media.2018.12.003.

[230] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: unified, real-time object detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016, pp. 779–788. DOI: 10.1109/cvpr.2016.91.

[231] S. Reiss, C. Seibold, A. Freytag, E. Rodner, and R. Stiefelhagen. "Every annotation counts: multi-label deep supervision for medical image segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2021, pp. 9532–9542. DOI: 10.1109/cvpr46437.2021.00941.

[232] A. Rogozhnikov. "Einops: clear and reliable tensor manipulations with einstein-like notation". In: *International Conference on Learning Representations*. 2022.

[233] G. Rolls. *An introduction to specimen processing*. https://www.leicabiosystems.com/knowledge-pathway/an-introduction-to-specimen-processing/. 2024. (Visited on 12/15/2024).

[234] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[235] J. Ryu et al. "OCELOT: overlapped cell on tissue dataset for histopathology". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2023, pp. 23902–23912. DOI: 10.1109/cvpr52729.2023.02289.

[236] R. Salgado et al. "The evaluation of tumor-infiltrating lymphocytes (TILs) in breast cancer: recommendations by an International TILs Working Group 2014". In: *Annals of Oncology* 26.2 (Feb. 2015), pp. 259–271. ISSN: 0923-7534. DOI: 10.1093/annonc/mdu450.

[237] J. Saltz et al. "Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images". In: *Cell Reports* 23.1 (Apr. 2018), 181–193.e7. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2018.03.086.

[238] A. Schmidt, P. Morales-Álvarez, and R. Molina. "Probabilistic Modeling of Inter- and Intra-observer Variability in Medical Image Segmentation". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2023, pp. 21040–21049. DOI: 10.1109/iccv51070.2023.01929. URL: http://dx.doi.org/10.1109/ICCV51070.2023.01929.

[239] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. "Cell detection with star-convex polygons". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 265–273. DOI: 10.1007/978-3-030-00934-2_30.

[240] R. Schmitz et al. "Multi-scale fully convolutional neural networks for histopathology image segmentation: from nuclear aberrations to the global tissue architecture". In: *Medical image analysis* 70 (May 2021), p. 101996. DOI: 10.1016/j.media.2021.101996.

[241] L. A. Schoenpflug and V. H. Koelzer. "SoftCTM: cell detection by soft instance segmentation and consideration of cell-tissue interaction". In: *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology*. Springer Nature Switzerland, 2024, pp. 109–122. DOI: 10.1007/978-3-031-55088-1_10.

[242] B. Schömig-Markiefka et al. "Quality control stress test for deep learning-based diagnostic model in digital pathology". In: *Modern Pathology* 34.12 (Dec. 2021), pp. 2098–2108. ISSN: 0893-3952. DOI: 10.1038/s41379-021-00859-x.

[243] C. Schuhmann et al. *LAION-400M: open dataset of CLIP-filtered 400 million image-text pairs*. 2021. DOI: 10.48550/ARXIV.2111.02114.

[244] Sectra Medical Systems AB. *Sectra and leica biosystems first in the world to gain FDA clearance to utilize DICOM images for pathology diagnostics*. https://medical.sectra.com/news-press-releases/news-item/BE0A9A62F2C3A460/. 2024. (Visited on 11/12/2024).

[245] A. R. C. Silva and R. J. D. C. Vieira. "Predictive factors of melanoma thickness". In: *Anais Brasileiros de Dermatologia* 97.5 (Sept. 2022), pp. 601–605. ISSN: 0365-0596. DOI: 10.1016/j.abd.2021.12.002.

[246] Simon Häger, Product Manager Sectra Digital Pathology Solution. *Introduction to the DICOM standard for digital pathology and its importance for workflow efficiency*. https://medical.sectra.com/resources/introduction-dicom-standard-digital-pathology/. 2016. (Visited on 12/11/2024).

[247] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot. "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images". In: *IEEE Transactions on Medical Imaging* 35.5 (May 2016), pp. 1196–1206. DOI: 10.1109/TMI.2016.2525803.

[248] A. H. Song et al. "Artificial intelligence for digital and computational pathology". In: *Nature Reviews Bioengineering* 1.12 (Oct. 2023), pp. 930–949. ISSN: 2731-6092. DOI: 10.1038/s44222-023-00096-8.

[249] Y. Song et al. "Accurate cervical cell segmentation from overlapping clumps in pap smear images". In: *IEEE Transactions on Medical Imaging* 36.1 (Jan. 2017), pp. 288–300. DOI: 10.1109/TMI.2016.2606380.

[250] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu. "Cellpose: a generalist algorithm for cellular segmentation". In: *Nature methods* 18.1 (Jan. 2021), pp. 100–106. DOI: 10.1038/s41592-020-01018-x.

[251] Y. Sucaet. *Pathomation - what WSI data are really made of*. https://realdata.pathomation.com/what-wsi-data-are-really-made-of/. 2020. (Visited on 12/11/2024).

[252] P. Sun et al. "A computational tumor-infiltrating lymphocyte assessment method comparable with visual reporting guidelines for triple-negative breast cancer". In: *EBioMedicine* 70 (Aug. 2021), p. 103492. ISSN: 2352-3964. DOI: 10.1016/j.ebiom.2021.103492.

[253] *Supplement 145: whole slide microscopic image iod and sop classes*. 24th ed. Rosslyn, Virgina, USA: NEMA, Aug. 2010.

[254] S. Takahashi et al. "Comparison of Vision Transformers and convolutional neural networks in medical image analysis: a systematic review". In: *Journal of Medical Systems* 48.1 (Sept. 2024). ISSN: 1573-689X. DOI: 10.1007/s10916-024-02105-8.

[255] M. Tan and Q. Le. "EfficientNet: rethinking model scaling for convolutional neural networks". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 6105–6114.

[256] A. Tareef et al. "Multi-pass fast watershed for accurate segmentation of overlapping cervical cells". In: *IEEE Transactions on Medical Imaging* 37.9 (Mar. 2018), pp. 2044–2059. DOI: 10.1109/TMI.2018.2815013.

[257] J. M. Taube et al. "The society for immunotherapy of cancer statement on best practices for multiplex immunohistochemistry (IHC) and immunofluorescence (IF) staining and validation". In: *Journal for ImmunoTherapy of Cancer* 8.1 (May 2020), e000155. ISSN: 2051-1426. DOI: 10.1136/jitc-2019-000155.

[258] D. Tellez, G. Litjens, J. van der Laak, and F. Ciompi. "Neural image compression for gigapixel histopathology image analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.2 (Feb. 2021), pp. 567–578. ISSN: 1939-3539. DOI: 10.1109/tpami.2019.2936841.

[259] D. Tellez et al. "Extending unsupervised neural image compression with supervised multitask learning". In: *Proceedings of the Third Conference on Medical Imaging with Deep Learning*. Vol. 121. Proceedings of Machine Learning Research. PMLR, June 2020, pp. 770–783.

[260] The pandas development team. *pandas-dev/pandas: pandas*. Zenodo. 2024. DOI: 10.5281/ZENODO.3509134.

[261] The Royal College of Pathologists. *The pathology workforce*. https://www.rcpath.org/discover-pathology/public-affairs/the-pathology-workforce.html. 2019. (Visited on 11/21/2024).

[262] H. Tokunaga, Y. Teramoto, A. Yoshizawa, and R. Bise. "Adaptive weighting multi-field-of-view cnn for semantic segmentation in pathology". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12597–12606. DOI: 10.1109/CVPR.2019.01288.

[263] H. Touvron et al. *LLaMA: open and efficient foundation language models*. 2023. DOI: 10.48550/ARXIV.2302.13971.

[264] S. Van der Walt et al. "scikit-image: image processing in Python". In: *PeerJ* 2 (June 2014), e453. DOI: 10.7717/peerj.453.

[265] M. Van Rijthoven, M. Balkenhol, K. Siliņa, J. Van Der Laak, and F. Ciompi. "HookNet: multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images". In: *Medical Image Analysis* 68 (Feb. 2021), p. 101890. DOI: 10.1016/j.media.2020.101890.

[266] A. Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[267] M. Veta, P. J. van Diest, R. Kornegoor, A. Huisman, M. A. Viergever, and J. P. W. Pluim. "Automatic nuclei segmentation in H&E stained breast cancer histopathology images". In: *PLOS ONE* 8.7 (July 2013), null. DOI: 10.1371/journal.pone.0070221.

[268] M. Veta et al. "Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge". In: *Medical Image Analysis* 54 (May 2019), pp. 111–121. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.02.012.

[269] P. Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature Methods* 17 (Feb. 2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[270] E. Vorontsov et al. "A foundation model for clinical-grade computational pathology and rare cancers detection". In: *Nature Medicine* 30.10 (July 2024), pp. 2924–2935. ISSN: 1546-170X. DOI: 10.1038/s41591-024-03141-0.

[271] Q. D. Vu et al. "Methods for segmentation and classification of digital microscopy tissue images". In: *Frontiers in Bioengineering and Biotechnology* 7 (Apr. 2019). ISSN: 2296-4185. DOI: 10.3389/fbioe.2019.00053.

[272] H. Wang et al. *Mixed transformer u-net for medical image segmentation*. 2021. DOI: 10.48550/arXiv.2111.04734.

[273] S. Wang, D. M. Yang, R. Rong, X. Zhan, and G. Xiao. "Pathology image analysis using segmentation deep learning algorithms". In: *The American journal of pathology* 189.9 (Sept. 2019), pp. 1686–1698. DOI: 10.1016/j.ajpath.2019.05.007.

[274] X. Wang et al. "Transformer-based unsupervised contrastive learning for histopathological image classification". In: *Medical Image Analysis* 81 (Oct. 2022), p. 102559. ISSN: 1361-8415. DOI: 10.1016/j.media.2022.102559.

[275] X. Wang et al. "RetCCL: clustering-guided contrastive learning for whole-slide image retrieval". In: *Medical Image Analysis* 83 (Jan. 2023), p. 102645. ISSN: 1361-8415. DOI: 10.1016/j.media.2022.102645.

[276] J. Wasserthal et al. "TotalSegmentator: robust segmentation of 104 anatomic structures in CT images". In: *Radiology: Artificial Intelligence* 5.5 (Sept. 2023). ISSN: 2638-6100. DOI: 10.1148/ryai.230024.

[277] M. Weigert and U. Schmidt. "Nuclei instance segmentation and classification in histopathology images with stardist". In: *2022 IEEE International Symposium on Biomedical Imaging Challenges (ISBIC)*. Mar. 2022, pp. 1–4. DOI: 10.1109/ISBIC56247.2022.9854534.

[278] M. Weigert et al. "Content-aware image restoration: pushing the limits of fluorescence microscopy". In: *Nature Methods* 15.12 (Nov. 2018), pp. 1090–1097. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0216-7.

[279] Weights & Biases. *Weights & Biases docs - sweeps*. https://docs.wandb.ai/guides/sweeps/. 2024. (Visited on 12/30/2024).

[280] T. Wen et al. "Review of research on the instance segmentation of cell images". In: *Computer Methods and Programs in Biomedicine* 227 (Dec. 2022), p. 107211. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2022.107211.

[281] Z. Weng et al. "GrandQC: a comprehensive solution to quality control problem in digital pathology". In: *Nature Communications* 15.1 (Dec. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-54769-y.

[282] S. Wienert et al. "Detection and segmentation of cell nuclei in virtual microscopy images: a minimum-model approach". In: *Scientific Reports* 2.1 (July 2012). DOI: 10.1038/srep00503.

[283] R. Wightman. *PyTorch image models*. GitHub. https://github.com/rwightman/pytorch-image-models. 2019. DOI: 10.5281/zenodo.4414861.

[284] G. Wölflein et al. *Benchmarking pathology feature extractors for whole slide image classification*. 2023. DOI: 10.48550/ARXIV.2311.11772.

[285] World Health Organization: Regional Office for Europe. *World cancer report*. IARC, Jan. 2020.

[286] B. Wu and G. Moeckel. "Application of digital pathology and machine learning in the liver, kidney and lung diseases". In: *Journal of Pathology Informatics* 14 (2023), p. 100184. ISSN: 2153-3539. DOI: 10.1016/j.jpi.2022.100184. URL: http://dx.doi.org/10.1016/j.jpi.2022.100184.

[287] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng. "Hyperparameter optimization for machine learning models based on bayesian optimization". In: *Journal of Electronic Science and Technology* 17.1 (Mar. 2019), pp. 26–40. DOI: 10.11989/JEST.1674-862X.80904120.

[288] E. Wulczyn et al. "Deep learning-based survival prediction for multiple cancer types using histopathology images". In: *PLOS ONE* 15.6 (June 2020), e0233678. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0233678.

[289] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. "SegFormer: simple and efficient design for semantic segmentation with transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.

[290] Z. Xie et al. "SimMIM: a simple framework for masked image modeling". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022, pp. 9643–9653. DOI: 10.1109/cvpr52688.2022.00943.

[291] F. Xing and L. Yang. "Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review". In: *IEEE Reviews in Biomedical Engineering* 9 (Jan. 2016), pp. 234–263. ISSN: 1941-1189. DOI: 10.1109/rbme.2016.2515127.

[292] H. Xu et al. "A whole-slide foundation model for digital pathology from real-world data". In: *Nature* 630.8015 (May 2024), pp. 181–188. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07441-w.

[293] H. Xu et al. "A whole-slide foundation model for digital pathology from real-world data". In: *Nature* 630.8015 (May 2024), pp. 181–188. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07441-w. URL: http://dx.doi.org/10.1038/s41586-024-07441-w.

[294] X. Yang, H. Li, and X. Zhou. "Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 53.11 (Nov. 2006), pp. 2405–2414. DOI: 10.1109/TCSI.2006.884469.

[295] K. Yao, K. Huang, J. Sun, and A. Hussain. "PointNu-Net: keypoint-assisted convolutional neural network for simultaneous multi-tissue histology nuclei segmentation and classification". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 8.1 (June 2023), pp. 802–813. DOI: 10.1109/TETCI.2023.3281864.

[296] R. Yerushalmi, R. Woods, P. M. Ravdin, M. M. Hayes, and K. A. Gelmon. "Ki67 in breast cancer: prognostic and predictive potential". In: *The Lancet Oncology* 11.2 (Feb. 2010), pp. 174–183. ISSN: 1470-2045. DOI: 10.1016/s1470-2045(09)70262-1.

[297] M. D. Zarella et al. "A practical guide to whole slide imaging: a white paper from the digital pathology association". In: *Archives of Pathology and Laboratory Medicine* 143.2 (Oct. 2018), pp. 222–234. ISSN: 1543-2165. DOI: 10.5858/arpa.2018-0343-ra.

[298] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. "Barlow twins: self-supervised learning via redundancy reduction". In: *International conference on machine learning*. PMLR. 2021, pp. 12310–12320.

[299] D. Y. Zhang, A. Venkat, H. Khasawneh, R. Sali, V. Zhang, and Z. Pei. "Implementation of Digital Pathology and Artificial Intelligence in Routine Pathology Practice". In: *Laboratory Investigation* 104.9 (Sept. 2024), p. 102111. ISSN: 0023-6837. DOI: 10.1016/j.labinv.2024.102111. URL: http://dx.doi.org/10.1016/j.labinv.2024.102111.

[300] Y. Zhong, X. Li, H. Mei, and S. Xiong. "Probability-based nuclei detection and critical-region guided instance segmentation". In: *Pattern Recognition and Computer Vision*. Springer Nature Singapore, 2023, pp. 122–135. DOI: 10.1007/978-981-99-8558-6_11.

[301] J. Zhou et al. *iBOT: image BERT pre-training with online tokenizer*. 2021. DOI: 10.48550/ARXIV.2111.07832.

[302] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang. "UNet++: a nested u-net architecture for medical image segmentation". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer International Publishing, 2018, pp. 3–11. DOI: 10.1007/978-3-030-00889-5_1.

[303] E. Zimmermann et al. *Virchow2: scaling self-supervised mixed magnification models in pathology*. 2024. DOI: 10.48550/ARXIV.2408.00738.

Appendix

Transparency Statement

Enclosed is the transparency statement for each Chapter, detailing the specific publications each Chapter draws upon. It includes a comprehensive breakdown of the corresponding Sections from the original publications and outlines any additions or modifications made in this dissertation.

Background and Related Works

Multiscale Segmentation Approaches in Digital Pathology

This Section is based on the "Related Work" Section of the Valuing Vicinity publication [76].

Micro-level Analysis: Cellular Segmentation

This Section is based on the "Related Work" Section of the CellViT publication [119].

Self-Supervised Learning and the Advent of Foundation Models

Most of this Chapter has been specifically written for this thesis. Certain parts, however, draw on the foundation model overview presented in the Appendix of the CellViT++ publication [116]. Additionally, content related to the Segment Anything and HIPT-256 models is based on the CellViT publication.

A Comprehensive Framework for Whole-Slide Image Preprocessing

This Chapter is substantially based on our BVM publication 2024 [117]. I extended the work by incorporating an analysis regarding the DICOM file format. The structure and content of this Chapter have been adapted and expanded from the mentioned publication[1] as follows:

- **Introduction:** The "Introduction" Section is based on the similarly titled part of the publication.

[1] Mit freundlicher Genehmigung von Springer Nature

- **Integrating Preprocessing into Hospital IT Systems:** This part was added to this thesis and is not based on the publication.
- **Methods:** This Section is an expansion and elaboration of the "Material and Methods" Section from the publication. We extended the software requirements as well as the technical overview and added an overview the file formats.
- **Experimental Setup:** This Section is based upon Section 2.3 "Experimental setup" from the original publication. We extended the description of the experiments and added further experiments regarding the DICOM implementations.
- **Results and Analysis:** This Section is based on the "Results" Section of the publication, extended by the DICOM-based evaluation.
- **Chapter Conclusion:** This Section is based on the "Discussion" Section of the publication.

Whole Tissue Segmentation

This Chapter is based on the joint publication in Computerized Medical Imaging and Graphics from 2023 [76]. I shortened the Methods and Results Sections, such that the method and reasoning is still understandable, but the focus lies on the application using the Selocan cohort.

- **Context and Objectives:** This Section was added by myself to motivate the domain specific network architecture.
- **Methods:** This part is based upon the similar named Section of the original publication, but rephrased and shortened. The mathematical notation has been adapted.
- **Experimental Setup:** This Section is a shortened and rephrased version of the original publication.
- **Results and Analysis:** The results and experiments presented were conducted as part of the initial publication process. While some of these findings are included in this thesis, the focus is primarily on the MAF, excluding the ablation studies from the publication.
- **Clinical Application:** This Section was specifically written for this thesis and is not part of the CMIG publication.
- **Chapter Conclusion:** This Section was mainly written new for this thesis, but some contributions are based on the statements within the "Discussion" Section and "Introduction" of the publication.

Enhancing Cell-level Analysis: CellViT and beyond

CellViT: A Novel Approach to Cell Segmentation

This Section in Chapter 5 is substantially based on our Medical Image Analysis publication from 2024 [119].

- **Introduction:** The introductory description is based on the "Introduction" Section of the publication.
- **Methods:** This part is adapted from the similarly titled Section of the publication with language improvements and rephrasing to match the overall style and structure of this thesis.
- **Experimental Setup:** This part is adapted from the similarly titled Section of the publication with language improvements and rephrasing to match the overall style and structure of this thesis.
- **Results and Analysis:** This Section is based on the "Results" and "Discussion and Conclusion" Section of the publication, with minor adaptations.

Additionally, Tables A.10-A.16 in the Appendix have been either included or adapted from the publication. The same applies to Figures A.2-A.3 in the Appendix.

CellViT++: Enhancing Cellular Analysis Capabilities

This Section in Chapter 5 is substantially based on our Preprint from 2025 [116].

- **Introduction and Enhancements and Methodological Innovations:** The introductory contextualization is based on the introductory description of the publication.
- **Methods:** This part is adapted from the similarly titled Section in the Appendix of the publication with language improvements and rephrasing to match the overall style and structure of this thesis. We just included the methodological description of the cell classification module, AutoML training, implementation improvements and SegPath description from the publication in this Section.
- **Experimental Setup:** This Section is based on both the "Results" Section and the Appendix of the publication. We combined both Sections to explain the dataset description and procedures for each experiment. However, the full dataset description from the publication's Appendix has been included in this thesis's Appendix as well. Additionally, the evaluation metrics were included.
- **Results and Analysis:** This Section is based on the "Results" Section of the publication, with minor adaptations and truncations.

Additionally, Tables A.18-A.38 in the Appendix have been either included or adapted from the publication. The same applies to the Figures A.4 and A.7 in the

Appendix.

Chapter Conclusion

The Chapter conclusion 5.4 is based on the "Discussion and Conclusion" Section of the CellViT publication [119] as well as on the "Discussion" Section of the CellViT++ publication [116].

Declaration

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly. The legal binding declaration ("Eidesstattliche Versicherung") is appended at the end of this dissertation. All scientific content, ideas, and analyses presented in this thesis are my own original work or properly cited where derived from other sources.

I acknowledge the use of the following writing assisting tools in the preparation of this thesis: Grammarly (Version 2025), DeepL Translate and Write (Version 2024/2025). For the language editing of the original CellViT++ publication, ChatGPT (GPT 3.5 and GPT 4/4o) has been used. All these tools were used solely for language editing purposes, specifically for grammar correction, clarity improvement, and text coherence enhancement. I declare that these tools were not used to generate, modify, or enhance any scientific content, analysis, or conclusions.

Supplementary Material

CellViT: A Novel Approach to Cell Segmentation

STARDIST and CPP-Net

Due to the new probability branch PD and the new radial distances branch RD, the loss function of STARDIST changes to:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{PD}} + \mathcal{L}_{\text{RD}} + \mathcal{L}_{\text{NT}}$$

with the individual loss branches

$$\begin{aligned}\mathcal{L}_{\text{PD}} &= \lambda_{\text{PD}_{\text{BCE}}}\mathcal{L}_{\text{BCE}} \\ \mathcal{L}_{\text{SD}} &= \lambda_{\text{SD}_{\text{MSE}}}\mathcal{L}_{\text{MSE}} \\ \mathcal{L}_{\text{NT}} &= \lambda_{\text{NT}_{\text{DICE}}}\mathcal{L}_{\text{DICE}} + \lambda_{\text{NT}_{\text{BCE}}}\mathcal{L}_{\text{BCE}}\end{aligned}$$

and weighting factors $\lambda_{\text{PD}_{\text{BCE}}} = \lambda_{\text{SD}_{\text{MSE}}} = \lambda_{\text{NT}_{\text{DICE}}} = \lambda_{\text{NT}_{\text{BCE}}} = 1$. For $\mathcal{L}_{\text{PD}}$ and $\mathcal{L}_{\text{PD}}$, the loss is weighted by the ground truth object probabilities. When using the CPP-Net networks, we used the STARDIST loss function, but changed the nuclei type loss to $\mathcal{L}_{\text{NT}} = \lambda_{\text{NT}_{\text{FT}}}\mathcal{L}_{\text{FT}} + \lambda_{\text{NT}_{\text{DICE}}}\mathcal{L}_{\text{DICE}} + \lambda_{\text{NT}_{\text{BCE}}}\mathcal{L}_{\text{BCE}}$, with $\lambda_{\text{NT}_{\text{FT}}} = 0.5$, $\lambda_{\text{NT}_{\text{DICE}}} = 0.2$ and $\lambda_{\text{NT}_{\text{BCE}}} = 0.5$, as we achieved superior results with this setting.

Sampling Weights

The calculation of the weighting factor of the tissue class can be calculated directly via

$$w_{\text{Tissue}}(i, \gamma_{\text{s}}) = \frac{N_{\text{Train}}}{\gamma_{\text{s}}\left(\sum_{j \in [1, N_{\text{Train}}] \| c_{\text{T},j} = c_{\text{T},i}} 1\right) + (1 - \gamma_{\text{s}})N_{\text{Train}}}$$

as each patch can only belong to one tissue class denoted by $c_{\text{T},i}$. For cell weighting, it must be considered that each patch can contain multiple nuclei from different cell

classes. Therefore, we create a binary vector $\boldsymbol{c_i} \in \{0,1\}^C$, where each entry is set to 1 for each existing nuclei type c in the patch. To get a reference value for scaling similar to the tisse, we calculate $N_{\text{Cell}} = \sum_{i=1}^{N_{\text{Train}}} \|\boldsymbol{c}_i\|_1$. The cell weighting for each training image i is then calculated by

$$w_{\text{Cell}}(i, \gamma_{\text{s}}) = (1 - \gamma_{\text{s}}) + \gamma_{\text{s}} \sum_{j=1}^{C} c_{ij} \frac{N_{\text{Cell}}}{\gamma_{\text{s}} \sum_{k=1}^{N_{\text{Train}}} c_{kj} + (1 - \gamma_{\text{s}}) N_{\text{Cell}}},$$

with c_{ij} the vector entry of $\boldsymbol{c}_i$ at position j. The training images are randomly sampled in a training epoch with replacement based on their sampling weights $p_i(\gamma_{\text{s}})$.

Supplementary Tables

A Comprehensive Framework for Whole-Slide Image Preprocessing

Table A.1: Throughput (s/100 patches) comparison of the CuCIM and OpenSlide back-ends across varying number of parallel processing. Data is grouped by patch count, with mean values and standard deviations (in brackets) calculated from five independent runs. The experiments were conducted using CuCIM v23.06.00 and OpenSlide v3.4.1. Whole-slide images were acquired with an Aperio AT2 scanner and stored in the default .svs file format.

Patch-Amount		<3,000	3,000 – 5,000	5,000 – 20,000	20,000 – 50,000	>50,000
Naming		Tiny	Small	Medium	Large	Huge
Back-end	Processes					
CuCIM	1	1.721 (0.390)	1.696 (0.511)	3.273 (0.244)		
	2	0.883 (0.184)	0.880 (0.264)	1.663 (0.130)		
	4	0.497 (0.089)	0.515 (0.124)	0.876 (0.068)		
	8	0.450 (0.027)	0.467 (0.018)	0.474 (0.018)	0.426 (0.029)	0.445 (0.022)
	16	0.472 (0.029)	0.459 (0.019)	0.417 (0.017)	0.427 (0.026)	0.460 (0.031)
OpenSlide	1	1.722 (0.357)	1.700 (0.520)	3.303 (0.269)		
	2	0.871 (0.203)	0.892 (0.267)	1.641 (0.117)		
	4	0.608 (0.073)	0.678 (0.108)	0.866 (0.045)		
	8	0.606 (0.051)	0.658 (0.094)	0.624 (0.052)	0.783 (0.032)	0.804 (0.024)
	16	0.598 (0.058)	0.680 (0.091)	0.650 (0.051)	0.789 (0.029)	0.822 (0.033)

Table A.2: Throughput (s/100 patches) comparison of all integrated WSI processing back-ends in our preprocessing framework. Reported is the mean with SD in brackets across all images for varying amount of parallel processes. All experiments were rerun after upgrading OpenSlide to v4.0.0 for DICOM support. For the experiments with CuCIM, OpenSlide (SVS), and WSI-DICOM, we used identical slides (5,000 - 20,000 patches, medium), while for the OpenSlide DICOM experiments a new dataset was acquired with an Aperio GT 450DX scanner, with matching patch distribution.

Back-end	CuCIM	OpenSlide	WSIDICOM	OpenSlide
Cohort	Initial Cohort	Initial Cohort	Initial Cohort (DICOM)	Additional DICOM Cohort
Dataformat	svs	svs	dcm	dcm
Conversion	-	-	WSIDICOMIZER	-
Scanner	Aperio AT2	Aperio AT2	Aperio AT2	Aperio 450 DX
Processes				
1	3.200 (0.246)	3.193 (0.243)	3.406 (0.092)	2.040 (0.126)
2	1.611 (0.126)	1.611 (0.121)	3.078 (0.031)	1.017 (0.064)
4	0.820 (0.061)	0.849 (0.038)	3.076 (0.027)	0.524 (0.032)
8	0.474 (0.015)	0.658 (0.056)	3.087 (0.032)	0.315 (0.019)
16	0.433 (0.016)	0.668 (0.053)	3.120 (0.027)	0.274 (0.006)
Verification using 520 px patch size and target resolution of 0.50 µm/px				
8	7.487 (0.140)	11.372 (0.098)	22.780 (0.950)	4.802 (0.044)

Table A.3: Runtime decrease comparison between PathoPatcher and TIAToolbox (TIA) when using 8 parallel processes. We compare PathoPatcher with CuCIM and OpenSlide as image back-ends. Experiments have been performed on a subset of 3 images per group. We report the average across all WSI together with the standard deviation in brackets.

Throughput (s/100 patches)	Tiny	Small	Medium	Large	Huge
OpenSlide	0.614 (0.046)	0.597 (0.028)	0.664 (0.006)	0.783 (0.032)	0.804 (0.024)
CuCIM	0.442 (0.037)	0.456 (0.020)	0.460 (0.012)	0.426 (0.029)	0.445 (0.022)
TIA	0.748 (0.098)	0.718 (0.023)	0.804 (0.002)	0.928 (0.031)	0.948 (0.032)
Relative Speed Reduction					
CuCIM/TIA	0.402 (0.094)	0.363 (0.047)	0.428 (0.017)	0.541 (0.018)	0.531 (0.008)
CuCIM/OpenSlide	0.279 (0.056)	0.233 (0.065)	0.308 (0.025)	0.456 (0.021)	0.448 (0.011)
OpenSlide/TIA	0.174 (0.069)	0.169 (0.013)	0.173 (0.006)	0.157 (0.008)	0.151 (0.004)

Table A.4: Important Resources and identifiers used for the preprocessing experiments.

Resource	Source	Identifier	Reference
PathoPatcher Requirements			-
cuCIM 23.06.00	Rapidsai	github.com/rapidsai/cucim	-
CuPY 13.3.0	Pip	cupy.dev	Okuta et al. [209]
GeoJSON 3.1.0	Pip	python-geojson.readthedocs.io/en/latest	-
NumPy 1.23.5	Pip	numpy.org	Harris et al. [105]
OpenSlide 3.4.1	Conda-forge	openslide.org	Goode et al. [91]
OpenSlide 4.0.0	Conda-forge	openslide.org	Goode et al. [91]
openslide-python 1.2.0	Pip	openslide.org/api/python	Goode et al. [91]
openslide-python 1.3.1	Pip	openslide.org/api/python	Goode et al. [91]
opencv-python-headless 4.5.4.58	Pip	opencv.org	Bradski [20]
pandas 2.2.3	Pip	pandas.pydata.org	The pandas development team [260]
PathoPatch 1.0.3	Pip	github.com/TIO-IKIM/PathoPatcher	Hörst et al. [117]
Pillow 10.4.0	Conda-forge	pillow.readthedocs.io	-
PyDicom 2.4.4	Pip	pydicom.github.io	Mason et al. [185]
Python 3.10.12	Conda-forge	www.python.org	-
Rasterio 1.3.5.post1	Pip	rasterio.readthedocs.io/en/latest	-
scikit-image 0.24.0	Pip	scikit-image.org	Van der Walt et al. [264]
scipy 1.14.1	Pip	scipy.org	Virtanen et al. [269]
Shapely 1.8.5.post1	Pip	github.com/shapely/shapely	Gillies et al. [87]
torch 2.2.1	Pip	pytorch.org	Paszke et al. [214]
torchvision 0.17.1	Pip	pytorch.org	-
WSIDICOM 0.20.4	Pip	github.com/imi-bigpicture/wsidicom	-
WSIDICOMIZER 0.14.1	Pip	github.com/imi-bigpicture/wsidicomizer	-
Comparison Methods			
CLAM	GitHub	github.com/mahmoodlab/CLAM (commit: 9482cbc72df522087cfbaa3e6b52da5207a7980a)	Lu et al. [176]
Deep-Histopath	GitHub	github.com/CODAIT/deep-histopath (commit: c8baf8d47b6c08c0f6c7b1fb6d5dd6b77e711c33)	Dusenberry et al. [69]
PathFlowAI	GitHub	github.com/jlevy44/PathFlowAI (commit: 5bd66ec800efbe413e16f5f0b8fd30278147c0ac)	Levy et al. [159]
FAU-DLM	GitHub	github.com/FAU-DLM/wsi_processing_pipeline (commit: d141a2cde57b10737be0c8e35c3720cb027758d0)	Neuner et al. [205]
SliDL	GitHub	github.com/markowetzlab/slidl (commit: 821f5357b2f8926d366adb0108e894a3da98d17c)	Berman et al. [17]
TIAToolbox	GitHub	github.com/TissueImageAnalytics/tiatoolbox (commit: 51f504b9c831481c2d8cccf4aeb8f4d2db9d44eb)	Pocock et al. [222]

Whole Tissue Segmentation

Table A.5: RCC baseline 5-fold CV $\overline{DICE_{\text{Total}}}$ for ascending resolutions. Encoder pretrained on ImageNet. Best in bold, second best underlined. Resolution for subsequent experiments was selected based on the U-Net + ResNet18 performance to achieve a fair comparison with msY-Net. Adapted from [76].

	U-Net		DeepLabV3	
Resolution	ResNet-18	ResNet-50	ResNet-18	ResNet-50
1.383 µm/px	<u>0.44</u>	<u>0.45</u>	<u>0.47</u>	0.47
2.766 µm/px	**0.45**	**0.48**	**0.49**	**0.50**
5.533 µm/px	0.42	<u>0.45</u>	0.46	<u>0.48</u>
11.066 µm/px	0.33	0.39	0.42	0.44

Table A.6: CY16 baseline 5-fold CV $\overline{DICE_{\text{Tumor}}}$ results for ascending resolutions. Encoder pretrained on ImageNet. Best in bold, second best underlined. Resolution for subsequent experiments was selected based on the U-Net + ResNet18 performance to achieve a fair comparison with msY-Net. Adapted from [76].

	U-Net		DeepLabV3	
Resolution	ResNet-18	ResNet-50	ResNet-18	ResNet-50
0.243 µm/px	0.68	0.68	0.74	0.73
0.486 µm/px	**0.73**	0.72	<u>0.76</u>	0.75
0.972 µm/px	0.71	<u>0.73</u>	<u>0.76</u>	<u>0.76</u>
1.944 µm/px	<u>0.72</u>	**0.75**	**0.77**	**0.78**
3.888 µm/px	0.63	0.72	0.73	<u>0.76</u>

Table A.7: Paip 2019 baseline 5-fold CV $\overline{DICE_{\text{Total}}}$ results for ascending resolutions. Encoder pretrained on ImageNet. Best in bold, second best underlined. Resolution for subsequent experiments was selected based on the U-Net + ResNet18 performance to achieve a fair comparison with msY-Net. Adapted from [116].

	U-Net		DeepLabV3	
Resolution	ResNet-18	ResNet-50	ResNet-18	ResNet-50
0.502 µm/px	0.66	0.67	0.70	0.70
1.004 µm/px	**0.70**	<u>0.71</u>	**0.73**	<u>0.74</u>
2.008 µm/px	**0.70**	**0.72**	**0.73**	**0.76**
4.016 µm/px	<u>0.68</u>	0.70	<u>0.70</u>	0.73
8.032 µm/px	0.58	0.62	0.62	0.65

Table A.8: MAF hyperparameter for all training runs. Adapted from [116].

Parameter	Value
Loss	Cross-Entropy
Optimizer	SGD
	$\eta = 1 \cdot 10^{-4}$ ($\eta = 2 \cdot 10^{-4}$ for msY-Net)
Training	epochs = 100 (early-stopping = 10), batch-size = 32
	lr-scheduling = exponential decay (factor 0.95)
Attention	$D = 128$ (hidden attention dim), 8 Heads

Table A.9: Important resources and identifiers used for the MAF experiments.

Resource	Source	Identifier	Reference
MAF Requirements			-
einops 0.4.1	Pip	einops.rocks/	Rogozhnikov [232]
GeoJSON 2.5.0	Pip	python-geojson.readthedocs.io/en/latest	-
NumPy 1.21.0	Pip	numpy.org	Harris et al. [105]
OpenSlide 3.4.1	Conda-forge	openslide.org	Goode et al. [91]
openslide-python 1.2.0	Pip	openslide.org/api/python	Goode et al. [91]
opencv-python-headless 4.5.5.62	Pip	opencv.org	Bradski [20]
Pillow 9.0.1	Conda-forge	pillow.readthedocs.io	-
Python 3.8.15	Conda-forge	www.python.org	-
segmentation-models-pytorch 0.2.1	Pip	github.com/qubvel-org/segmentation_models.pytorch	Iakubovskii [122]
scikit-learn 1.2.0	Pip	scikit-learn.org	Pedregosa et al. [220]
scipy 1.4.1	Pip	scipy.org	Virtanen et al. [269]
Shapely 1.8.1.post1	Pip	github.com/shapely/shapely	Gillies et al. [87]
timm 0.4.12	Pip	github.com/huggingface/pytorch-image-models	Wightman [283]
torch 1.10.0	Pip	pytorch.org	Paszke et al. [214]
torchvision 0.11.1	Pip	pytorch.org	Paszke et al. [214]
Comparison Methods			
msY-Net	GitHub	github.com/ipmi-icns-uke/multiscale (commit: ef2859a76f03be02c7845ba70f91e28354d29dba)	Schmitz et al. [240]
Pretrained Models			
RN50-CCL	GitHub	github.com/Xiyue-Wang/RetCCL (Checkpoint with MD5 d9947c03f4fe6edde7f517546f49ce93)	Wang et al. [275]
RN50-Lunit	GitHub	github.com/lunit-io/benchmark-ssl-pathology (Checkpoint bt_rn50_ep200.torch MD5 e5621a2350d4023b78870fd75dc27862)	Kang et al. [137]

CellViT: A Novel Approach to Cell Segmentation

Table A.10: Precision (P), Recall (R) and F_1-score (F_1) for detection and classification across the three PanNuke splits for each nuclei type. The centroid of each nucleus was used for computing detection metrics for segmentation networks. Adapted from [119]. *TSFD-Net was not evaluated on the official three-fold splits of the PanNuke dataset and left out by the comparison **Model retrained by ourselves ***Models trained on downscaled $0.50\,\mu m/px$ PanNuke images.

Model	Decoder	Hyperparameters	Detection			Classification																	
						Neoplastic			Epithelial			Inflammatory			Connective			Dead					
			P_d	R_d	$F_{1,d}$	P_{Neo}	R_{Neo}	$F_{1,Neo}$	P_{Epi}	R_{Epi}	$F_{1,Epi}$	P_{Inf}	R_{Inf}	$F_{1,Inf}$	P_{Con}	R_{Con}	$F_{1,Con}$	P_{Dead}	R_{Dead}	$F_{1,Dead}$			
DIST			0.74	0.71	0.73	0.49	0.55	0.50	0.38	0.33	0.35	0.42	0.45	0.42	0.42	0.37	0.39	0.00	0.00	0.00			
Mask-RCNN			0.76	0.68	0.72	0.55	0.63	0.59	0.52	0.52	0.52	0.46	0.54	0.50	0.42	0.43	0.42	0.17	0.30	0.22			
Micro-Net			0.78	0.82	0.80	0.59	0.66	0.62	0.63	0.54	0.58	0.59	0.46	0.52	0.50	0.45	0.47	0.23	0.17	0.19			
HoVer-Net			0.82	0.79	0.80	0.58	0.67	0.62	0.54	0.60	0.56	0.56	0.51	0.54	0.52	0.47	0.49	0.28	0.35	0.31			
TSFD-Net*			0.84	0.87	0.85	0.60	0.71	0.65	0.56	0.58	0.57	0.59	0.58	0.57	0.55	0.49	0.53	0.33	0.40	0.43			
STARDIST (ResNet50) **	STARDIST	CPP-Net	0.85	0.80	0.82	0.69	0.69	0.69	0.73	0.68	0.70	0.62	0.53	0.57	0.54	0.49	0.51	0.39	0.09	0.10			
STARDIST (ResNet50) **	STARDIST	CellViT	0.85	0.79	0.82	0.70	0.66	0.68	0.71	0.66	0.68	0.58	0.58	0.58	0.54	0.49	0.51	0.39	0.34	0.36			
CellViT$_{HIPT-256}$ – Raw	HoVer-Net	CellViT	0.80	0.77	0.78	0.61	0.64	0.63	0.63	0.59	0.61	0.55	0.46	0.50	0.45	0.43	0.44	0.43	0.16	0.23			
CellViT$_{HIPT-256}$ – Over	HoVer-Net	CellViT	0.79	0.78	0.78	0.62	0.63	0.62	0.65	0.59	0.62	0.54	0.47	0.50	0.44	0.45	0.44	**0.46**	0.16	0.24			
CellViT$_{HIPT-256}$ – Aug	HoVer-Net	CellViT	0.83	0.82	0.82	0.70	0.69	0.69	0.68	0.71	0.69	0.58	**0.59**	0.58	0.54	0.51	0.52	0.38	0.35	0.36			
CellViT$_{HIPT-256}$ – No-FC	HoVer-Net	CellViT	0.82	**0.83**	0.82	0.69	**0.70**	0.69	0.70	0.69	0.70	0.58	0.58	0.58	0.53	0.51	0.52	0.40	0.33	0.36			
CellViT$_{Random}$ (no pretrain)	HoVer-Net	CellViT	0.79	0.81	0.80	0.63	0.65	0.64	0.63	0.62	0.72	0.54	0.57	0.55	0.49	0.46	0.48	0.30	0.34	0.31			
CellViT$_{HIPT-256}$	HoVer-Net	CellViT	0.83	0.82	0.82	0.69	**0.70**	0.69	0.68	0.71	0.70	0.59	0.58	0.58	0.53	0.51	0.52	0.39	0.35	0.37			
CellViT$_{SAM-B}$	HoVer-Net	CellViT	0.83	0.82	**0.83**	0.70	**0.70**	0.70	0.70	0.72	0.71	0.59	0.58	**0.59**	0.54	**0.52**	**0.53**	**0.46**	0.29	0.36			
CellViT$_{SAM-L}$	HoVer-Net	CellViT	0.84	0.82	**0.83**	0.71	**0.70**	0.70	0.71	0.72	0.72	0.59	0.58	0.58	0.54	**0.52**	**0.53**	0.42	0.36	**0.39**			
CellViT$_{SAM-H}$	HoVer-Net	CellViT	0.84	0.81	**0.83**	0.72	0.69	**0.71**	0.72	**0.73**	**0.73**	0.59	0.57	0.58	0.55	**0.52**	**0.53**	0.43	0.32	0.36			
CellViT$_{HIPT-256}$	STARDIST	CPP-Net	0.84	0.75	0.79	0.64	0.60	0.62	0.65	0.56	0.60	**0.64**	0.45	0.52	**0.58**	0.47	0.47	0.30	0.27	0.28			
CellViT$_{HIPT-256}$	STARDIST	CellViT	0.83	0.79	0.81	0.71	0.65	0.68	0.68	0.68	0.68	0.59	0.57	0.58	0.52	0.49	0.50	0.37	**0.38**	0.37			
CellViT$_{SAM-H}$	STARDIST	CPP-Net	0.84	0.78	0.81	0.68	0.66	0.67	0.71	0.62	0.66	0.57	0.57	0.57	0.54	0.45	0.49	0.36	0.32	0.32			
CellViT$_{SAM-H}$	STARDIST	CellViT	0.84	0.80	0.82	0.72	0.68	0.70	**0.74**	0.71	0.72	0.60	0.57	0.58	0.53	0.51	0.52	0.44	0.34	0.38			
CellViT$_{HIPT-256}$	CPP-Net	CPP-Net	0.85	0.76	0.80	0.69	0.62	0.65	0.70	0.62	0.65	0.57	0.55	0.56	0.53	0.46	0.49	0.32	**0.38**	0.33			
CellViT$_{HIPT-256}$	CPP-Net	CellViT	**0.87**	0.76	0.81	0.73	0.64	0.68	0.71	0.65	0.68	0.58	0.57	0.58	0.55	0.47	0.51	0.37	0.37	0.37			
CellViT$_{SAM-H}$	CPP-Net	CPP-Net	0.86	0.78	0.82	0.72	0.67	0.70	0.73	0.68	0.70	0.62	0.55	0.58	0.55	0.50	0.52	0.27	0.14	0.18			
CellViT$_{SAM-H}$	CPP-Net	CellViT	**0.87**	0.78	0.82	**0.74**	0.67	0.70	**0.74**	0.70	0.72	0.60	0.57	0.58	0.57	0.49	**0.53**	0.41	0.36	0.38			
CellViT$_{HIPT-256}$ $(0.50\,\mu m/px)$***	HoVer-Net	CellViT	0.86	0.60	0.71	0.72	0.59	0.65	0.71	0.58	0.64	0.60	0.38	0.47	0.53	0.32	0.40	0.43	0.04	0.07			
CellViT$_{SAM-H}$ $(0.50\,\mu m/px)$***	HoVer-Net	CellViT	0.88	0.63	0.73	0.74	0.62	0.67	0.74	0.61	0.67	0.60	0.42	0.49	0.56	0.34	0.42	0.49	0.04	0.08			

Table A.11: Average mPQ and bPQ across the 19 tissue types of the PanNuke dataset for three-fold cross-validation. The standard deviation (SD) of the splits is provided in the final row. STARDIST models with with ResNet50 (RN50) encoder were retrained with CPP-Net hyperparameters (CPP-HP) and CellViT hyperparameters (CellViT-HP) for comparison. For the CellViT models, just the architecture with HoVer-Net decoder (HV-Net) is given. Adapted from [119]. *TSFD-Net was not evaluated on the official three-fold splits of the PanNuke dataset and left out by the comparison **STARDIST trained by Chen et al. [46] ***Model retrained by ourselves.

	HoVer-Net		TSFD-Net*		STARDIST**		STARDIST***				CPP-Net		CellViT$_{\text{HIPT-256}}$		CellViT$_{\text{SAM-H}}$	
					RN50 encoder		RN50 encoder CPP-HP		RN50 encoder CellViT-HP		RN50 encoder		HV-Net decoder CellViT-HP		HV-Net decoder CellViT-HP	
Tissue	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ
Adrenal	0.4812	0.6962	0.5223	0.6900	0.4868	0.6972	0.4928	0.6954	0.4834	0.6884	0.4922	0.7031	0.4950	0.7009	**0.5134**	**0.7086**
Bile Duct	0.4714	0.6696	0.5000	0.6284	0.4651	0.6690	0.4632	0.6583	0.4680	0.6564	0.4650	0.6739	0.4721	0.6705	**0.4887**	**0.6784**
Bladder	0.5792	0.7031	0.5738	0.6773	0.5793	0.6986	0.5643	0.6949	0.5730	0.6901	**0.5932**	0.7057	0.5756	0.7056	0.5844	**0.7068**
Breast	0.4902	0.6470	0.5106	0.6245	0.5064	0.6666	0.4948	0.6585	0.4889	0.6497	0.5066	0.6718	0.5089	0.6641	**0.5180**	**0.6748**
Cervix	0.4438	0.6652	0.5204	0.6561	0.4628	0.6690	0.4752	0.6739	0.4781	0.6685	0.4779	**0.6880**	0.4893	0.6862	**0.4984**	0.6872
Colon	0.4095	0.5575	0.4382	0.5370	0.4205	0.5779	0.4230	0.5704	0.4087	0.5555	0.4269	0.5888	0.4245	0.5700	**0.4485**	**0.5921**
Esophagus	0.5085	0.6427	0.5438	0.6306	0.5331	0.6655	0.5200	0.6508	0.5175	0.6446	0.5410	**0.6755**	0.5373	0.6619	**0.5454**	0.6682
Head & Neck	0.4530	0.6331	0.4937	0.6277	0.4768	0.6433	0.4660	0.6305	0.4629	0.6215	0.4667	0.6468	0.4901	0.6472	**0.4913**	**0.6544**
Kidney	0.4424	0.6836	0.5517	0.6824	**0.5880**	0.6998	0.5090	0.6888	0.4750	0.6800	0.5092	0.7001	0.5409	0.6993	0.5366	**0.7092**
Liver	0.4974	0.7248	0.5079	0.6675	0.5145	0.7231	0.4899	0.7106	0.5034	0.7051	0.5099	0.7271	0.5065	0.7160	**0.5224**	**0.7322**
Lung	0.4004	0.6302	0.4274	0.5941	0.4128	0.6362	0.3627	0.6087	0.3931	0.6205	0.4234	0.6364	0.4102	0.6317	**0.4314**	**0.6426**
Ovarian	0.4863	0.6309	0.5253	0.6431	0.5205	0.6668	0.5106	0.6573	0.5204	0.6547	0.5276	**0.6792**	0.5260	0.6596	**0.5390**	0.6722
Pancreatic	0.4600	0.6491	0.4893	0.6241	0.4585	0.6601	0.4548	0.6516	0.4526	0.6439	0.4680	**0.6742**	**0.4769**	0.6643	0.4719	0.6658
Prostate	0.5101	0.6615	0.5431	0.6406	0.5067	0.6748	0.4905	0.6561	0.4812	0.6457	0.5261	**0.6903**	0.5164	0.6695	**0.5321**	0.6821
Skin	0.3429	0.6234	0.4354	0.6074	0.3610	0.6289	0.3826	0.6349	0.3709	0.6197	0.3547	0.6192	0.3661	0.6400	**0.4339**	**0.6565**
Stomach	**0.4726**	0.6886	0.4871	0.6529	0.4477	0.6944	0.4239	0.6769	0.4194	0.6642	0.4553	**0.7043**	0.4475	0.6918	0.4705	0.7022
Testis	0.4754	0.6890	0.4843	0.6435	0.4942	0.6869	0.4819	0.6848	**0.5141**	0.6812	0.4917	**0.7006**	0.5091	0.6883	0.5127	0.6955
Thyroid	0.4315	0.6983	0.5154	0.6692	0.4300	0.6962	0.4246	0.6962	0.4175	0.6921	0.4344	0.7094	0.4412	0.7035	**0.4519**	**0.7151**
Uterus	0.4393	0.6393	0.5068	0.6204	0.4480	0.6599	0.4452	0.6455	0.4683	0.6428	**0.4790**	0.6622	0.4737	0.6516	0.4737	**0.6625**
Average	0.4629	0.6596	0.5040	0.6377	0.4796	0.6692	0.4671	0.6602	0.4682	0.6539	0.4815	0.6767	0.4846	0.6696	**0.4980**	**0.6793**
SD	0.0076	0.0036	-	-	-	-	0.0489	0.0340	0.0496	0.0348	-	-	0.0503	0.0340	0.0413	0.0318

Table A.12: Average mPQ and bPQ across the 19 tissue types of the PanNuke dataset for three-fold cross-validation for models trained on downscaled $0.50\,\mu m/px$ PanNuke images. The standard deviation (SD) of the splits is provided in the final row. Just the CellViT architecture with HoVer-Net decoder (HV-Net) is given. For comparison, we also included the networks trained and evaluated on original $0.25\,\mu m/px$ PanNuke images in the first two columns. Adapted from [119]. *Models trained on downscaled $0.50\,\mu m/px$ PanNuke images

	$CellViT_{HIPT-256}$		$CellViT_{SAM-H}$		$CellViT_{HIPT-256}$*		$CellViT_{SAM-H}$*	
	HV-Net decoder CellViT-HP		HV-Net decoder CellViT-HP		HV-Net decoder CellViT-HP $0.50\,\mu m/px$		HV-Net decoder CellViT-HP $0.50\,\mu m/px$	
Tissue	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ
Adrenal	0.4950	0.7009	0.5134	0.7086	0.3947	0.5967	0.4226	0.6139
Bile Duct	0.4721	0.6705	0.4887	0.6784	0.3594	0.5278	0.3791	0.5587
Bladder	0.5756	0.7056	0.5844	0.7068	0.3205	0.4221	0.3423	0.4457
Breast	0.5089	0.6641	0.5180	0.6748	0.4260	0.5761	0.4592	0.6097
Cervix	0.4893	0.6862	0.4984	0.6872	0.3713	0.5302	0.3967	0.5618
Colon	0.4245	0.5700	0.4485	0.5921	0.3139	0.4352	0.3485	0.4680
Esophagus	0.5373	0.6619	0.5454	0.6682	0.4485	0.5604	0.4574	0.5793
Head & Neck	0.4901	0.6472	0.4913	0.6544	0.2597	0.3954	0.2821	0.4136
Kidney	0.5409	0.6993	0.5366	0.7092	0.3517	0.4805	0.3831	0.5203
Liver	0.5065	0.7160	0.5224	0.7322	0.3634	0.5415	0.3673	0.5659
Lung	0.4102	0.6317	0.4314	0.6426	0.3040	0.4261	0.3161	0.4489
Ovarian	0.5260	0.6596	0.5390	0.6722	0.4454	0.5691	0.4714	0.6033
Pancreatic	0.4769	0.6643	0.4719	0.6658	0.3395	0.4914	0.3465	0.5194
Prostate	0.5164	0.6695	0.5321	0.6821	0.3764	0.5243	0.3999	0.5404
Skin	0.3661	0.6400	0.4339	0.6565	0.2552	0.4481	0.2948	0.4835
Stomach	0.4475	0.6918	0.4705	0.7022	0.2948	0.5029	0.3105	0.5259
Testis	0.5091	0.6883	0.5127	0.6955	0.3856	0.5307	0.4031	0.5771
Thyroid	0.4412	0.7035	0.4519	0.7151	0.3527	0.6090	0.3758	0.6209
Uterus	0.4737	0.6516	0.4737	0.6625	0.3615	0.4972	0.3783	0.5384
Average	0.4846	0.6696	0.4980	0.6793	0.3539	0.5087	0.3755	0.5366
SD	0.0503	0.0340	0.0413	0.0318	0.0546	0.0618	0.0541	0.0613

Table A.13: Comparison of network parameters and approximated Multiply-Accumulate-Operations (MACs) across cell segmentation models, including the CNN based HoVer-Net and STARDIST-ResNet50, and our Vision Transformer variants with multiple decoder configurations. The number of MACs is an approximation to measure the computational complexity of a neural network. Train MACs were computed with a batch size of 1 and RGB images sized at 256×256 px for all networks. In the inference calculation, we used a patch size of $1{,}024 \times 1{,}024$ px for CellViT models and 256×256 px for reference methods (HoVer-Net, STARDIST-ResNet50), maintaining a batch size of 1. To facilitate a comparison within the same field of view, we adjusted the batch size to 36 for HoVer-Net and 16 for STARDIST-ResNet50 (indicated in parentheses), aligning with the output sizes of 164×164 px for HoVer-Net and 256×256 px for STARDIST-ResNet50. Adapted from [119].

Modell	Decoder	Parameters (in Million)	Total Train MACs (in Million)	Total Inference MACs (in Million)
HoVer-Net	HoVer-Net	37.6	148.9	148.9 (5361.6 for CellViT FOV)
STARDIST (ResNet50)	STARDIST	123.2	292.2	292.2 (4674.8 for CellViT FOV)
$\text{CellViT}_{\text{HIPT-256}}$	HoVer-Net	46.8	132.9	2125.9
$\text{CellViT}_{\text{HIPT-256}}$	STARDIST	46.8	133.1	2127.9
$\text{CellViT}_{\text{HIPT-256}}$	CPP-Net	46.8	133.5	2135.6
$\text{CellViT}_{\text{SAM-B}}$	HoVer-Net	146.1	200.1	3200.6
$\text{CellViT}_{\text{SAM-L}}$	HoVer-Net	367.8	207.1	3306.5
$\text{CellViT}_{\text{SAM-H}}$	HoVer-Net	699.7	214.2	3413.4
$\text{CellViT}_{\text{SAM-H}}$	STARDIST	699.7	214.3	3415.4
$\text{CellViT}_{\text{SAM-H}}$	CPP-Net	699.7	214.8	3423.1

Table A.14: Selected data augmentation techniques with probability and additional hyperparameters. Data augmentation is implemented with Albumentations. STARDIST just uses spatial transformations. For CPP-Net we used the same augmentations as for HoVer-Net decoder, because they achieved superior results. Adapted from [119].

Augmentation Technique	Probability	Hyperparamter
90-degree rotation	0.5	
Horizontal flipping	0.5	
Vertical flipping	0.5	
Downscaling	0.15	max-scale: 0.5 min-scale: 0.5
Blurring	0.2	blur-limit: 10
Gaussian noise	0.25	var_limit: 50
Color jittering	0.2	brightness: 0.25 contrast: 0.25 saturation: 0.1 hue: 0.05
Superpixel representation	0.1	p_replace: 0.1 n_segments: 200 max-size: $H/2$
Zoom blur	0.1	max-factor: 1.05
Random cropping with resizing	0.1	crop-level: 0.5-1.0 of input size
Elastic transformation	0.2	sigma: 25 alpha: 0.5 alpha-affine: 15
Normalization	1.0	Mean: $[0.5, 0.5, 0.5]$ STD: $[0.5, 0.5, 0.5]$

Table A.15: CellViT hyperparameter for all training runs on the PanNuke dataset. Weights (λ_i) for loss function were chosen heuristically, inspired by TSFD-Net. Parameters were carefully defined to ensure a balanced range across combined loss functions, preventing dominance of one branch. Additionally, test runs on CellViT$_{256}$ were used to select candidate values. MSE and MSGE loss parameters were increased, such that training for the HV branch is increased. We note that the impact of each λ_i is most pronounced within the respective branch, considering that three different segmentation decoder branches are used and the encoder is frozen for the first 25 epochs. Adapted from [119].

Parameter	Value
Loss	$\lambda_{\text{NP}_{\text{FT}}} = 1$, $\lambda_{\text{NP}_{\text{FT}}} = 1$, $\lambda_{\text{NP}_{\text{DICE}}} = 1$, $\lambda_{\text{HV}_{\text{MSE}}} = 2.5$, $\lambda_{\text{HV}_{\text{MSGE}}} = 8$, $\lambda_{\text{NT}_{\text{FT}}} = 0.5$, $\lambda_{\text{NT}_{\text{DICE}}} = 0.2$, $\lambda_{\text{NT}_{\text{BCE}}} = 0.5$, $\lambda_{\text{TC}_{\text{CE}}} = 0.1$, $\alpha_{\text{FT}} = 0.7$, $\beta_{\text{FT}} = 0.3$, $\gamma_{\text{FT}} = 4/3$, $\varepsilon_{\text{FT}} = 1 \cdot 10^{-6}$
Sampling	$\gamma_{\text{s}} = 0.85$
Optimizer	AdamW
Training	$\eta = 3 \cdot 10^{-4}$, $\lambda = 1 \cdot 10^{-4}$, $\beta_1 = 0.85$, $\beta_2 = 0.85$, epochs = 130, batch-size = 16, lr-scheduling = 0.85

Table A.16: CPP-Net hyperparameter for all training runs on the PanNuke dataset. Adapted from [119].

Parameter	Value
Loss	Compare Appendix Material
Sampling	$\gamma_s = 0.0$
Optimizer	Adam
Training	$\eta = 3 \cdot 10^{-4}$, $\lambda = 1 \cdot 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, epochs = 130, batch-size = 16, lr-scheduling = reducelronplateau (multiply by 0.5)

Table A.17: Important Resources and identifiers used for the CellViT experiments. Comparison methods just include methods trained or evaluated by us.

Resource	Source	Identifier	Reference
CellViT Requirements			-
Albumentations 1.3.0	Pip	albumentations.ai	Buslaev et al. [24]
CSBDeep 0.7.4	Pip	csbdeep.bioimagecomputing.com	Weigert et al. [278]
cuCIM 22.12.00	Rapidsai	github.com/rapidsai/cucim	-
CuPY 13.2.0	Pip	cupy.dev	Okuta et al. [209]
GeoJSON 3.0.0	Pip	python-geojson.readthedocs.io/en/latest	-
NumPy 1.23.5	Pip	numpy.org	Harris et al. [105]
OpenSlide 3.4.1	Conda-forge	openslide.org	Goode et al. [91]
openslide-python 1.2.0	Pip	openslide.org/api/python	Goode et al. [91]
opencv-python-headless 4.5.4.58	Pip	opencv.org	Bradski [20]
pandarallel 1.6.5	Pip	github.com/nalepae/pandarallel	-
pandas 1.5.3	Pip	pandas.pydata.org	The pandas development team [260]
Pillow 10.0.1	Conda-forge	pillow.readthedocs.io	-
Python 3.10.12	Conda-forge	www.python.org	-
Rasterio 1.3.5.post1	Pip	rasterio.readthedocs.io/en/latest	-
scikit-image 0.19.3	Pip	scikit-image.org	Van der Walt et al. [264]
scikit-learn 1.2.1	Pip	scikit-learn.org	Pedregosa et al. [220]
scipy 1.8.1	Pip	scipy.org	Virtanen et al. [269]
Shapely 1.8.5.post1	Pip	github.com/shapely/shapely	Gillies et al. [87]
stardist 0.8.5	Pip	stardist.net	Weigert et al. [277]
torch 1.13.1	Pip	pytorch.org	Paszke et al. [214]
torchmetrics 0.11.4	Pip	lightning.ai/docs/torchmetrics	Detlefsen et al. [62]
torchvision 0.14.1	Pip	pytorch.org	Paszke et al. [214]
ujson 5.8.0	Pip	pypi.org/project/ujson	-
WandB 0.15.4	Pip	wandb.ai	-
Comparison Methods			
STARDIST/CPP-Net	GitHub	github.com/csccscscscsc/cpp-net (commit: 1f804df89d557823108485c0df6d1ad1591c28f5)	Chen et al. [46]
Foundation Models			-
HIPT	GitHub	github.com/mahmoodlab/HIPT (commit: 7336ee7d4fc70a348358cab31984cba093a77983)	Chen et al. [44]
Segment Anything	GitHub	github.com/facebookresearch/segment-anything	Kirillov et al. [142]
Datasets			-
CoNSeP	University of Warwick	warwick.ac.uk/fac/cross_fac/tia/data/hovernet	Graham et al. [95]
PanNuke	University of Warwick	warwick.ac.uk/fac/cross_fac/tia/data/pannuke	Gamper et al. [84]
MoNuSeg	Grand Challenge	monuseg.grand-challenge.org	Kumar et al. [149, 150]

CellViT++: Enhancing Cellular Analysis Capabilities

Table A.18: Summary of the cell datasets used for CellViT^{++} token analysis. Adapted from [116].

Dataset	Cell Classes	Organs	Nuclei Amount	Patches/Slide Amount	Source	Resolution Magnification	Seg. Mask	Note
Ocelot	Tumor, Non-tumor	Kidney, Head/Neck, Prostate, Stomach, Endometrium, Bladder	113,026	Total 303 Slides (173 train, 65 val, 65 test), cut into 663 tiles (400 train, 137 val,126 test) with size 1,024 × 1,024 px	TCGA	0.25 µm/px / ×40	No	The authors removed 4 test tiles due to missing annotations
MIDOG++	Mitotic figures, Non-mitotic figures	Breast (Human), Neuroendocrine Tumor (Human), Melanoma (Human) Cutaneous Mast Cell Tumor (Canine), Neuroendocrine Tumor (Canine), Lymphoma (Canine), Soft Tissue Sarcoma (Canine)	29,289 annotated, 7,389,755 total	503 images with various sizes, average of 6,384 × 5,102 px	UMC Utrecht, VMU Vienna, TU Berlin, AMC New York	0.23 µm/px - 0.25 µm/px	No	-
CoNSeP	Epithelial, Inflammatory, Spindle-shaped, Miscellaneous	Colon	24,332	41 patches (27 train, 14 test) with size 1,000 × 1,000 px	University Hospitals Coventry and Warwickshire (UK)	0.25 µm/px / ×40	Yes	Resized to 1,024 × 1,024 px
Lizard	Neutrophils, Lymphocytes, Plasma, Eosinophils, Epithelial, Connective	Colon	418,935	270 tiles of various sizes, 1,016 × 917 px on average at 0.50 µm/px	University Hospitals Coventry and Warwickshire (UK) TCGA, China (Four)	0.50 µm/px / ×20	Yes	Resized from 0.5 to 0.25 with Lanczos filter at input and back to 0.50 µm/px at the output
NuCLS	Lymphocytes and Plasma (superclass: sTILs), Macrophages and Stromal (superclass: stromal cells), Mitotic and Non-Mitotic Tumor (superclass: tumor cells), Miscellaneous	Breast	48,365	109 train and 15 test WSI with 1-3 annotated crops, average crop size of 362 × 382 px, average FOV size (annotated area) of 320 px	TCGA	0.20 µm/px / ×40	No	Cropped just FOV area and resize to 256 × 256 px to achieve 0.25 µm/px
PanopTILs	TILs, Stromal, Epithelial, Miscellaneous	Breast	859,759	1,709 train and 1,317 test tiles, with size 1,024 × 1,024 px, but test tiles just annotated in a narrow FOV	TCGA	0.25 µm/px / ×40	No	-
Segpath	IF-stainings of: Epithelial, Smooth muscle/Myofibroblasts, Lymphocytes, Leukocytes, Blood/lymphatic vessel, Plasma cells, Myeloid cells, Red blood cells	Bladder, Brain, Breast, Colon, Head/Neck, Kidney, Liver, Lung, Oesophagus, Ovary, Pancreas, Prostate, Sarcoma, Skin, Stomach, Testis, Thymus, Uterus	-	220 breast tiles for lymphocytes, 2,054 for plasma cells with size 984 × 984 px	University of Tokyo Hospital	0.22 µm/px / ×40	-	No nuclei have been annotated. The dataset consist of registered HE and IHC stainings to automatically derive region-wise annotations. We extract nuclei annotations by transferring binary CellViT^{++} results to the IHC mask.

Table A.19: Average mPQ and bPQ scores across the 19 tissue types of the PanNuke dataset for 3-fold CV for all CellViT variants. The standard deviation (SD) of the splits is provided in the final row. Best results are marked bold, second best underlined. Adapted from [116].

	$\text{CellViT}_{\text{HIPT-256}}$		$\text{CellViT}_{\text{UNI}}$		$\text{CellViT}_{\text{Virchow}}$		$\text{CellViT}_{\text{Virchow-2}}$		$\text{CellViT}_{\text{SAM-H}}$	
Tissue	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ
Adrenal	0.4950	0.7009	0.5033	0.6939	0.5087	0.6979	0.5028	0.6960	**0.5134**	**0.7086**
Bile Duct	0.4721	0.6705	0.4736	0.6592	0.4763	0.6643	0.4820	0.6729	**0.4887**	**0.6784**
Bladder	0.5756	0.7056	0.5789	0.6980	0.5599	0.6958	0.5595	0.6898	**0.5844**	**0.7068**
Breast	0.5089	0.6641	0.5006	0.6547	0.4999	0.6562	0.5043	0.6563	**0.5180**	**0.6748**
Cervix	0.4893	0.6862	0.4911	0.6815	0.4908	0.6766	0.4849	0.6724	**0.4984**	**0.6872**
Colon	0.4245	0.5700	0.4392	0.5729	0.4398	0.5730	0.4377	0.5728	**0.4485**	**0.5921**
Esophagus	0.5373	0.6619	0.5267	0.6525	0.5352	0.6573	0.5325	0.6544	**0.5454**	**0.6682**
Head & Neck	0.4901	0.6472	0.4827	0.6316	0.4709	0.6361	0.4904	0.6395	**0.4913**	**0.6544**
Kidney	0.5409	0.6993	**0.5610**	0.7094	0.5483	0.6940	0.5602	0.6975	0.5366	**0.7092**
Liver	0.5065	0.7160	0.5140	0.7180	0.5078	0.7180	0.5111	0.7158	**0.5224**	**0.7322**
Lung	0.4102	0.6317	0.4235	0.6250	0.4228	0.6205	**0.4314**	0.6308	**0.4314**	**0.6426**
Ovarian	0.5260	0.6596	0.5218	0.6564	0.5185	0.6529	0.5207	0.6620	**0.5390**	**0.6722**
Pancreatic	0.4769	0.6643	0.4723	0.6530	0.4700	0.6604	**0.4924**	0.6626	0.4719	**0.6658**
Prostate	0.5164	0.6695	0.5247	0.6666	0.5169	0.6682	0.5182	0.6640	**0.5321**	**0.6821**
Skin	0.3661	0.6400	0.4342	0.6320	**0.4442**	0.6425	0.4384	0.6327	0.4339	**0.6565**
Stomach	0.4475	0.6918	0.4463	0.6896	0.4354	0.6827	0.4405	0.6851	**0.4705**	**0.7022**
Testis	0.5091	0.6883	0.5024	0.6760	0.5046	0.6794	0.5044	0.6758	**0.5127**	**0.6955**
Thyroid	0.4412	0.7035	0.4596	0.7018	**0.4630**	0.7080	0.4598	0.7056	0.4519	**0.7151**
Uterus	0.4737	0.6516	0.4863	0.6481	0.4838	0.6497	**0.4873**	0.6526	0.4737	**0.6625**
Average	0.4846	0.6696	0.4917	0.6642	0.4893	0.6649	0.4926	0.6652	**0.4980**	**0.6793**
SD	0.0503	0.0340	0.0415	0.0351	0.0385	0.0338	0.0384	0.0327	0.0413	0.0318

Table A.20: Overview of the Ocelot dataset splits and amount of nuclei for the training, validation and test split. Adapted from [116].

	Training						**Validation**	**Test**
Amount	5%	10%	25%	50%	75%	100%		
Tumor Cells	1,571	5,056	12,607	22,994	32,766	42,505	16,312	12,874
Non-Tumor Cells	1,644	2,579	5,966	10,695	17,982	23,332	8,400	9,603
Total	3,215	7,635	18,573	33,689	50,748	65,837	24,712	22,477

Table A.21: Average mF_1-score of the baseline model (SoftCTM) and the best-performing CellViT^{++} model (CellViT$^{++}_{\text{SAM-H}}$) across different training data sizes, averaged over 5 runs. The performance of CellViT$^{++}_{\text{SAM-H}}$ is evaluated with and without data augmentation, with results provided for each organ along with the standard deviation. Adapted from [116].

	Organ	Bladder	Endometrium	Head & Neck	Kidney	Prostate	Stomach	Average
Amount	Network	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD
5%	SoftCTM	0.2488 ± 0.0928	0.2532 ± 0.0969	0.1820 ± 0.0916	0.3388 ± 0.1403	0.1987 ± 0.0668	0.2458 ± 0.1144	0.2692 ± 0.1105
	CellViT$^{++}_{\text{SAM-H}}$	0.5854 ± 0.0084	0.6875 ± 0.0070	0.5758 ± 0.0072	0.5575 ± 0.0120	0.6269 ± 0.0116	0.6224 ± 0.0108	0.6291 ± 0.0065
	CellViT$^{++}_{\text{SAM-H}}$ (Aug)	0.5901 ± 0.0140	0.6870 ± 0.0186	0.5750 ± 0.0152	0.5731 ± 0.0256	0.6499 ± 0.0104	0.6329 ± 0.0156	0.6362 ± 0.0082
10%	SoftCTM	0.5783 ± 0.0170	0.6355 ± 0.0223	0.5639 ± 0.0303	0.5932 ± 0.0115	0.5599 ± 0.0178	0.6543 ± 0.0182	0.6100 ± 0.0173
	CellViT$^{++}_{\text{SAM-H}}$	0.5988 ± 0.0039	0.7104 ± 0.0061	0.5894 ± 0.0088	0.5931 ± 0.0069	0.6392 ± 0.0090	0.6589 ± 0.0080	0.6507 ± 0.0035
	CellViT$^{++}_{\text{SAM-H}}$ (Aug)	0.5735 ± 0.0244	0.7181 ± 0.0086	0.5707 ± 0.0094	0.6005 ± 0.0177	0.6503 ± 0.0176	0.6749 ± 0.0084	0.6507 ± 0.0092
25%	SoftCTM	0.6693 ± 0.0084	0.7019 ± 0.0084	0.6176 ± 0.0312	0.6339 ± 0.0099	0.6340 ± 0.0093	0.7090 ± 0.0169	0.6736 ± 0.0037
	CellViT$^{++}_{\text{SAM-H}}$	0.6046 ± 0.0133	0.6964 ± 0.0111	0.5997 ± 0.0087	0.6081 ± 0.0157	0.6356 ± 0.0053	0.7080 ± 0.0084	0.6550 ± 0.0073
	CellViT$^{++}_{\text{SAM-H}}$ (Aug)	0.6091 ± 0.0066	0.7056 ± 0.0042	0.5574 ± 0.0176	0.6074 ± 0.0142	0.6371 ± 0.0223	0.7047 ± 0.0108	0.6539 ± 0.0073
50%	SoftCTM	0.6797 ± 0.0051	0.7195 ± 0.0067	0.5385 ± 0.0272	0.6573 ± 0.0060	0.6619 ± 0.0096	0.7425 ± 0.0058	0.6820 ± 0.0048
	CellViT$^{++}_{\text{SAM-H}}$	0.6287 ± 0.0132	0.7123 ± 0.0110	0.6041 ± 0.0121	0.6327 ± 0.0109	0.6594 ± 0.0083	0.6968 ± 0.0043	0.6695 ± 0.0072
	CellViT$^{++}_{\text{SAM-H}}$ (Aug)	0.6378 ± 0.0093	0.7160 ± 0.0060	0.5734 ± 0.0271	0.6457 ± 0.0170	0.6635 ± 0.0208	0.6994 ± 0.0078	0.6716 ± 0.0073
75%	SoftCTM	0.6899 ± 0.0094	0.7387 ± 0.0065	0.6001 ± 0.0333	0.6775 ± 0.0092	0.6384 ± 0.0286	0.7623 ± 0.0046	0.6986 ± 0.0038
	CellViT$^{++}_{\text{SAM-H}}$	0.6089 ± 0.0069	0.7168 ± 0.0058	0.6307 ± 0.0143	0.6235 ± 0.0121	0.6437 ± 0.0142	0.6950 ± 0.0089	0.6669 ± 0.0054
	CellViT$^{++}_{\text{SAM-H}}$ (Aug)	0.6293 ± 0.0071	0.7214 ± 0.0036	0.5898 ± 0.0192	0.6280 ± 0.0108	0.6544 ± 0.0173	0.6940 ± 0.0093	0.6694 ± 0.0053
100%	SoftCTM	0.6955 ± 0.0094	0.7459 ± 0.0074	0.6267 ± 0.0268	0.6779 ± 0.0207	0.6850 ± 0.0377	0.7660 ± 0.0066	0.7109 ± 0.0069
	CellViT$^{++}_{\text{SAM-H}}$	0.6276 ± 0.0085	0.7338 ± 0.0028	0.6359 ± 0.0095	0.6532 ± 0.0120	0.6542 ± 0.0091	0.7049 ± 0.0120	0.6827 ± 0.0028
	CellViT$^{++}_{\text{SAM-H}}$ (Aug)	0.6225 ± 0.0148	0.7311 ± 0.0076	0.6078 ± 0.0173	0.6374 ± 0.0106	0.6493 ± 0.0152	0.6825 ± 0.0110	0.6726 ± 0.0080

Table A.22: Average mF_1-score and standard deviation (SD) of the baseline model (SoftCTM) and all CellViT^{++} variants averaged over 5 runs when 100% of the training data is used for the Ocelot dataset. Adapted from [116].

	Organ	Bladder	Endometrium	Head & Neck	Kidney	Prostate	Stomach	Average
Network	Data Augmentation	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD	$mF_1 \pm$ SD
SoftCTM	Yes	0.6955 ± 0.0094	0.7459 ± 0.0074	0.6267 ± 0.0268	0.6779 ± 0.0207	0.6850 ± 0.0377	0.7660 ± 0.0066	0.7109 ± 0.0069
CellViT$^{++}_{\text{HIPT-256}}$	No	0.5832 ± 0.0023	0.7129 ± 0.0028	0.6053 ± 0.0041	0.5809 ± 0.0026	0.6158 ± 0.0030	0.6291 ± 0.0039	0.6426 ± 0.0024
	Yes	0.5855 ± 0.0059	0.7109 ± 0.0065	0.6069 ± 0.0143	0.5820 ± 0.0108	0.6134 ± 0.0055	0.6281 ± 0.0164	0.6425 ± 0.0078
CellViT$^{++}_{\text{UNI}}$	No	0.6337 ± 0.0020	0.6659 ± 0.0037	0.6399 ± 0.0040	0.6537 ± 0.0021	0.6158 ± 0.0054	0.6586 ± 0.0019	0.6537 ± 0.0019
	Yes	0.6362 ± 0.0054	0.6673 ± 0.0037	0.6426 ± 0.0027	0.6581 ± 0.0032	0.6217 ± 0.0047	0.6598 ± 0.0059	0.6565 ± 0.0022
CellViT$^{++}_{\text{Virchow}}$	No	0.5775 ± 0.0085	0.6301 ± 0.0048	0.5939 ± 0.0095	0.6058 ± 0.0046	0.5477 ± 0.0056	0.5920 ± 0.0041	0.6073 ± 0.0023
	Yes	0.5930 ± 0.0154	0.6386 ± 0.0071	0.5669 ± 0.0190	0.6091 ± 0.0076	0.5458 ± 0.0131	0.6034 ± 0.0048	0.6113 ± 0.0050
CellViT-Virchow-2	No	0.5773 ± 0.0136	0.6541 ± 0.0095	0.5966 ± 0.0141	0.6303 ± 0.0134	0.5477 ± 0.0144	0.5827 ± 0.0051	0.6158 ± 0.0095
	Yes	0.5561 ± 0.0061	0.6457 ± 0.0074	0.5958 ± 0.0200	0.6186 ± 0.0067	0.5438 ± 0.0066	0.5807 ± 0.0092	0.6071 ± 0.0032
CellViT$^{++}_{\text{SAM-H}}$	No	0.6276 ± 0.0085	0.7338 ± 0.0028	0.6359 ± 0.0095	0.6532 ± 0.0120	0.6542 ± 0.0091	0.7049 ± 0.0120	0.6827 ± 0.0028
	Yes	0.6225 ± 0.0148	0.7311 ± 0.0076	0.6078 ± 0.0173	0.6374 ± 0.0106	0.6493 ± 0.0152	0.6825 ± 0.0110	0.6726 ± 0.0080

Table A.23: Overview of the CoNSeP dataset splits and amount of nuclei for the entire training and test split as well as the tile level training subset. Adapted from [116].

	Nuclei Amount			
Tile Amount	Inflammatory	Epithelial	Spindle-Shaped	Miscellaneous
1	26	223	464	54
2	30	711	714	151
3	308	1,095	1,047	153
4	340	1,095	1,145	168
5	351	1,095	1,145	197
6	446	1,382	1,604	206
7	476	1,660	1,798	206
8	550	1,993	2,128	210
9	550	1,993	2,128	230
10	782	1,993	2,542	231
11	819	2,445	2,592	231
12	852	2,445	2,719	232
13	895	2,445	2,938	232
14	927	3,173	2,974	241
15	2,945	3,173	3,239	241
All Training Tiles	3,941	5,537	5,706	371
All Test Tiles	1,638	3,214	3,364	561

Table A.24: Binary comparison of multiple baseline models on the CoNSeP dataset. All baseline models have been trained on the training set, with subsequent validation on the test set. We report the original publication scores if available, but we also retrained the networks with the setup described in the original publication. Best results are marked bold, second best underlined. Adapted from [116].

Score	Binary Scores						
Model	F_1-Score	*DICE*	*AJI*	*AJI+*	*bPQ*	*bDQ*	*bSQ*
HoVer-Net (Orig-Publication)	-	**0.853**	<u>0.571</u>	-	<u>0.547</u>	<u>0.702</u>	**0.778**
HoVer-Net (PanNuke Baseline)	0.691	0.802	0.492	0.524	0.461	0.609	0.755
HoVer-Net (self-trained)	0.731	0.836	0.535	<u>0.563</u>	0.505	0.656	0.767
Pointnu-Net (self-trained)	0.737	0.782	0.525	0.561	0.522	0.686	0.759
CellViT$^{++}_{\text{HIPT-256}}$	<u>0.752</u>	0.815	0.527	0.556	0.504	0.663	0.758
CellViT$^{++}_{\text{UNI}}$	0.720	0.818	0.525	0.552	0.492	0.649	0.756
CellViT$^{++}_{\text{Virchow}}$	0.722	0.808	0.493	0.511	0.451	0.607	0.741
CellViT$^{++}_{\text{Virchow-2}}$	0.725	0.811	0.493	0.513	0.457	0.612	0.743
CellViT$^{++}_{\text{SAM-H}}$	**0.772**	<u>0.845</u>	**0.578**	**0.608**	**0.548**	**0.709**	<u>0.771</u>

Table A.25: Comparison of baseline models on the CoNSeP dataset. All baseline models have been trained on the training set, with subsequent validation on the test set. We report the original publication scores if available, but we also retrained the networks with the setup described in the original publication. The CellViT^{++} variants have all been trained and validated with 5 fold CV, with final evaluations of each fold on the test set. To estimate the average perfomance and distribution, we report the mean and standard deviation for our models. Adapted from [116].

Score	Class-Averaged-Scores					
Model	$mPQ \pm$ SD	$mDQ \pm$ SD	$mSQ \pm$ SD	$mPQ_+ \pm$ SD	$mDQ_+ \pm$ SD	$mSQ_+ \pm$ SD
HoVer-Net (Orig-Publication)	-	-	-	-	-	-
HoVer-Net (PanNuke Baseline)	-	-	-	-	-	-
HoVer-Net (self-trained)	0.364	0.463	<u>0.712</u>	0.429	0.550	**0.773**
Pointnu-Net (self-trained)	<u>0.383</u>	<u>0.495</u>	**0.722**	<u>0.446</u>	<u>0.588</u>	0.752
CellViT$^{++}_{\text{HIPT-256}}$	0.344 ± 0.014	0.447 ± 0.017	0.663 ± 0.007	0.398 ± 0.039	0.519 ± 0.053	0.761 ± 0.001
CellViT$^{++}_{\text{UNI}}$	0.341 ± 0.009	0.444 ± 0.011	0.669 ± 0.019	0.377 ± 0.026	0.491 ± 0.036	0.722 ± 0.073
CellViT$^{++}_{\text{Virchow}}$	0.329 ± 0.009	0.432 ± 0.011	0.667 ± 0.006	0.386 ± 0.027	0.511 ± 0.038	0.749 ± 0.002
CellViT$^{++}_{\text{Virchow-2}}$	0.331 ± 0.013	0.436 ± 0.016	0.675 ± 0.018	0.359 ± 0.023	0.474 ± 0.032	0.750 ± 0.007
CellViT$^{++}_{\text{SAM-H}}$	**0.397 ± 0.004**	**0.507 ± 0.006**	0.675 ± 0.008	**0.461 ± 0.014**	**0.596 ± 0.018**	<u>0.768 ± 0.001</u>

Table A.26: CellViT$^{++}_{\text{SAM-H}}$ performance with and without data augmentation on the CoNSeP dataset. In this experiment, the models have been trained on limited training data, starting from one ROI up to 15 ROIs. We report the performance among all cell types. For comparison, we include the baseline result with 100 % training data in the bottom part. Adapted from [116].

		Average	Inflammatory	Epithelium	Spindle-Shaped	Miscellaneous
Num-Files	Data Augmentation	$mPQ_+ \pm$ SD	$mPQ_+ \pm$ SD	$mPQ_+ \pm$ SD	$mPQ_+ \pm$ SD	$mPQ_+ \pm$ SD
1	No	0.326 ± 0.015	0.336 ± 0.040	0.467 ± 0.003	0.404 ± 0.002	0.098 ± 0.020
	Yes	0.316 ± 0.033	0.258 ± 0.059	0.444 ± 0.037	0.383 ± 0.019	0.180 ± 0.048
2	No	0.375 ± 0.015	0.331 ± 0.048	0.484 ± 0.005	0.410 ± 0.006	0.273 ± 0.016
	Yes	0.357 ± 0.026	0.235 ± 0.111	0.479 ± 0.013	0.392 ± 0.010	0.322 ± 0.020
3	No	0.441 ± 0.007	0.521 ± 0.021	0.503 ± 0.002	0.427 ± 0.002	0.314 ± 0.012
	Yes	0.441 ± 0.015	0.527 ± 0.042	0.496 ± 0.010	0.425 ± 0.005	0.317 ± 0.039
4	No	0.452 ± 0.005	0.551 ± 0.013	0.505 ± 0.000	0.427 ± 0.002	0.326 ± 0.012
	Yes	0.441 ± 0.015	0.531 ± 0.020	0.503 ± 0.006	0.426 ± 0.003	0.304 ± 0.052
5	No	0.443 ± 0.016	0.529 ± 0.027	0.503 ± 0.005	0.425 ± 0.004	0.315 ± 0.034
	Yes	0.453 ± 0.011	0.552 ± 0.038	0.507 ± 0.003	0.426 ± 0.004	0.329 ± 0.017
6	No	0.452 ± 0.007	0.537 ± 0.017	0.504 ± 0.002	0.425 ± 0.003	0.342 ± 0.008
	Yes	0.460 ± 0.007	0.552 ± 0.025	0.509 ± 0.005	0.428 ± 0.004	0.352 ± 0.008
7	No	0.459 ± 0.006	0.561 ± 0.018	0.505 ± 0.002	0.428 ± 0.001	0.343 ± 0.008
	Yes	0.453 ± 0.017	0.554 ± 0.020	0.506 ± 0.006	0.429 ± 0.002	0.323 ± 0.061
8	No	0.463 ± 0.006	0.571 ± 0.017	0.500 ± 0.005	0.431 ± 0.001	0.349 ± 0.007
	Yes	0.459 ± 0.008	0.553 ± 0.035	0.509 ± 0.004	0.431 ± 0.005	0.344 ± 0.013
9	No	0.460 ± 0.005	0.562 ± 0.013	0.503 ± 0.007	0.430 ± 0.004	0.344 ± 0.010
	Yes	0.464 ± 0.005	0.565 ± 0.020	0.508 ± 0.003	0.432 ± 0.001	0.350 ± 0.007
10	No	0.468 ± 0.004	0.580 ± 0.012	0.509 ± 0.001	0.432 ± 0.001	0.350 ± 0.006
	Yes	0.470 ± 0.003	0.587 ± 0.004	0.511 ± 0.001	0.435 ± 0.002	0.348 ± 0.007
11	No	0.462 ± 0.002	0.567 ± 0.012	0.506 ± 0.005	0.433 ± 0.002	0.344 ± 0.005
	Yes	0.469 ± 0.003	0.580 ± 0.003	0.511 ± 0.002	0.433 ± 0.003	0.353 ± 0.010
12	No	0.468 ± 0.002	0.583 ± 0.010	0.508 ± 0.002	0.431 ± 0.003	0.348 ± 0.007
	Yes	0.470 ± 0.006	0.584 ± 0.012	0.511 ± 0.002	0.434 ± 0.002	0.349 ± 0.015
13	No	0.467 ± 0.003	0.576 ± 0.015	0.510 ± 0.001	0.433 ± 0.001	0.351 ± 0.008
	Yes	0.465 ± 0.005	0.564 ± 0.019	0.511 ± 0.002	0.433 ± 0.003	0.351 ± 0.009
14	No	0.468 ± 0.003	0.584 ± 0.009	0.509 ± 0.003	0.432 ± 0.002	0.346 ± 0.013
	Yes	0.470 ± 0.004	0.580 ± 0.015	0.510 ± 0.001	0.434 ± 0.002	0.356 ± 0.002
15	No	0.466 ± 0.004	0.572 ± 0.018	0.510 ± 0.001	0.432 ± 0.002	0.353 ± 0.006
	Yes	0.473 ± 0.002	0.592 ± 0.006	0.509 ± 0.001	0.432 ± 0.003	0.357 ± 0.003
Baselines (100% Train)						
CellViT$^{++}_{\text{SAM-H}}$	No	0.461 ± 0.014	0.575 ± 0.010	0.507 ± 0.003	0.430 ± 0.002	0.330 ± 0.048
CellViT$^{++}_{\text{SAM-H}}$	Yes	0.442 ± 0.030	0.587 ± 0.007	0.501 ± 0.008	0.432 ± 0.002	0.247 ± 0.107
HoVer-Net	Yes	0.429	0.596	0.457	0.381	0.281
PointNu-Net	Yes	0.446	0.596	0.476	0.407	0.307

Table A.27: Summary of cell annotations present in the Lizard dataset split by cell type and data source. Adapted from [96].

Type/Dataset	DigestPath	CRAG	GlaS	CoNSeP	**Total**
Epithelial	70,789	99,124	31,986	2,898	204,797
Lymphocyte	49,932	27,634	9,763	1,317	88,646
Plasma	11,352	9,363	2,349	332	23,396
Neutrophil	2,262	1,673	90	30	4,055
Eosinophil	1,349	1,255	286	52	2,942
Connective	32,826	49,994	10,890	1,389	95,099
Total	168,510	189,043	55,364	6,018	418,935

Table A.28: Performance comparison on the Lizard dataset. We included the best available baseline models, all evaluated on a 3-Fold CV split. For our CellViT$^{++}_{\text{SAM-H}}$ model, we further evaluate the performance when using classical feature engineering for extracting cellular features (histomics), including the 3-layer Deep Learning classifier and CatBoost. External validation on the test set cannot be conducted, as it remains hidden. HoVer-Net Cerberus refers to the scores published in [97], whereas HoVer-Net baseline refers to the original scores published by the Lizard authors. Adapted from [116].

Model	Classifier	$DICE$ ± SD	bPQ ± SD	mPQ ± SD	mPQ_+ ± SD
HoVer-Net (Baseline)		0.828 ± 0.008	0.624 ± 0.013	0.396 ± 0.022	-
HoVer-Net (Cerberus)		-	0.584 ± 0.014	0.295 ± 0.018	0.409 ± 0.027
Cerberus		-	0.612 ± 0.010	0.358 ± 0.011	**0.425 ± 0.019**
CGIS-CPF		**0.889 ± 0.002**	**0.660 ± 0.061**	**0.421 ± 0.013**	-
CellViT$^{++}_{\text{HIPT-256}}$	DL, Token-based	0.767 ± 0.005	0.513 ± 0.011	0.252 ± 0.003	0.272 ± 0.009
CellViT$^{++}_{\text{UNI}}$	DL, Token-based	0.752 ± 0.005	0.503 ± 0.009	0.279 ± 0.004	0.293 ± 0.004
CellViT$^{++}_{\text{SAM-H}}$	DL, Token-based	0.774 ± 0.004	0.536 ± 0.009	0.294 ± 0.002	0.308 ± 0.006
CellViT$^{++}_{\text{SAM-H}}$	DL, Nuclei Features (Histomics)	0.774 ± 0.004	0.536 ± 0.009	0.255 ± 0.004	0.251 ± 0.003
CellViT$^{++}_{\text{SAM-H}}$	CatBoost, Nuclei Features (Histomics)	0.774 ± 0.004	0.536 ± 0.009	0.255 ± 0.009	0.255 ± 0.004

Table A.29: NuCLS nuclei amount for the main and super annotation classes within the corrected single annotator dataset used in this study. We excluded ambiguous labelled nuclei, as well as nuclei which center of mass lays outside the annotation field of view. Adapted from [116].

Main Annotations		Lympho-cyte	Plasma	Macro-phage	Stromal	Mitotic Tumor	Non-Mitotic Tumor	Uncategorized
Merged Super Annotations		sTILs		Stromal		Tumor		Uncategorized
Main	Train	11,162	4,233	1,153	7,214	167	16,921	291
	Test	1,748	1,065	118	1,421	44	2,544	284
Super	Train	15,395		8,367		7,088		291
	Test	2,813		1,539		2,588		284

Table A.30: Averaged (SD) 5-Fold CV results on the NuCLS main label set (15 test WSI). The single-rater corrected dataset (correction by pathologists) has been used to assess performance. Adapted from [116].

		Lymphocyte	Macrophage	Uncategorized	Plasma Cell	Stromal	Tumor (Non-Mitotic)	Tumor Mitotic
$\text{CellViT}^{++}_{\text{HIPT-256}}$	$F_1 \pm$ SD	0.654 ± 0.029	0.082 ± 0.056	0.029 ± 0.033	0.606 ± 0.082	0.670 ± 0.009	0.774 ± 0.011	0.008 ± 0.016
	$Prec \pm$ SD	0.528 ± 0.038	0.174 ± 0.108	0.248 ± 0.228	0.658 ± 0.016	0.593 ± 0.013	0.676 ± 0.018	0.029 ± 0.057
	$Rec \pm$ SD	0.861 ± 0.008	0.054 ± 0.038	0.016 ± 0.019	0.578 ± 0.147	0.769 ± 0.007	0.905 ± 0.009	0.005 ± 0.009
$\text{CellViT}^{++}_{\text{UNI}}$	$F_1 \pm$ SD	0.670 ± 0.022	0.239 ± 0.054	0.567 ± 0.073	0.570 ± 0.103	0.689 ± 0.010	0.801 ± 0.004	0.134 ± 0.049
	$Prec \pm$ SD	0.545 ± 0.030	0.325 ± 0.060	0.837 ± 0.074	0.718 ± 0.059	0.598 ± 0.028	0.714 ± 0.013	0.302 ± 0.102
	$Rec \pm$ SD	0.871 ± 0.017	0.212 ± 0.089	0.446 ± 0.122	0.481 ± 0.121	0.816 ± 0.023	0.913 ± 0.011	0.100 ± 0.053
$\text{CellViT}^{++}_{\text{SAM-H}}$	$F_1 \pm$ SD	0.661 ± 0.016	0.102 ± 0.034	0.037 ± 0.027	0.544 ± 0.087	0.691 ± 0.006	0.800 ± 0.008	0.092 ± 0.092
	$Prec \pm$ SD	0.531 ± 0.021	0.336 ± 0.062	0.467 ± 0.241	0.675 ± 0.049	0.601 ± 0.015	0.719 ± 0.014	0.213 ± 0.198
	$Rec \pm$ SD	0.878 ± 0.012	0.061 ± 0.023	0.020 ± 0.015	0.463 ± 0.110	0.814 ± 0.012	0.902 ± 0.003	0.059 ± 0.060

Table A.31: Performance of the single-cell type classifier trained on the automatically generated SegPath cell dataset, compared to NuCLS expert level annotations, with final evaluation on the NuCLS corrected single-rater test set for lymphocytes. Adapted from [116].

			Lymphocyte			Other		
Dataset	Network	$mF_1 \pm$ SD	$F_1 \pm$ SD	$Prec \pm$ SD	$Rec \pm$ SD	$F_1 \pm$ SD	$Prec \pm$ SD	$Rec \pm$ SD
NuCLS Base	SAM-H	0.769 ± 0.012	0.693 ± 0.020	0.595 ± 0.047	0.839 ± 0.052	0.846 ± 0.006	0.793 ± 0.021	0.908 ± 0.029
	HIPT-256	0.742 ± 0.011	0.652 ± 0.020	0.555 ± 0.037	0.795 ± 0.047	0.832 ± 0.005	0.764 ± 0.018	0.916 ± 0.021
	UNI	0.756 ± 0.004	0.673 ± 0.006	0.590 ± 0.018	0.787 ± 0.035	0.839 ± 0.003	0.767 ± 0.011	0.928 ± 0.011
SegPath	SAM-H	0.743	0.651	0.625	0.679	0.836	0.751	0.943
	HIPT-256	0.713	0.603	0.490	0.783	0.824	0.770	0.887
	UNI	0.738	0.642	0.532	0.809	0.834	0.779	0.897

Table A.32: Performance of the single-cell type classifier trained on the automatically generated SegPath cell dataset, compared to NuCLS expert level annotations, with final evaluation on the NuCLS corrected single-rater test set for plasma cells. Adapted from [116].

			Plasma Cell			Other		
Dataset	Encoder	$mF_1 \pm$ SD	$F_1 \pm$ SD	$Prec \pm$ SD	$Rec \pm$ SD	$F_1 \pm$ SD	$Prec \pm$ SD	$Rec \pm$ SD
NuCLS Base	HIPT-256	0.654 ± 0.075	0.486 ± 0.134	0.641 ± 0.112	0.432 ± 0.195	0.822 ± 0.017	0.715 ± 0.026	0.968 ± 0.013
	UNI	0.745 ± 0.042	0.651 ± 0.075	0.740 ± 0.073	0.594 ± 0.117	0.838 ± 0.009	0.738 ± 0.015	0.971 ± 0.005
	SAM-H	0.682 ± 0.033	0.524 ± 0.060	0.644 ± 0.041	0.449 ± 0.090	0.839 ± 0.006	0.742 ± 0.012	0.967 ± 0.005
SegPath	HIPT-256	0.582	0.354	0.668	0.240	0.811	0.692	0.980
	UNI	0.723	0.606	0.612	0.600	0.841	0.747	0.961
	SAM-H	0.740	0.632	0.653	0.613	0.848	0.760	0.960

Table A.33: Summary of cell annotations in the PanopTILs dataset split by cell type and data split. Adapted from [116].

	TILs	Stromal	Epithelial	Miscellaneous
Train	197,617	237,483	338,251	41,535
Test	14,237	7,986	16,704	5,946

Table A.34: PanopTILs reference results of our CellViT^{++} models, split across cell types. Adapted from [116].

		CellViT$^{++}_{\text{HIPT-256}}$	CellViT$^{++}_{\text{UNI}}$	CellViT$^{++}_{\text{SAM-H}}$
TILs	F_1 ± SD	0.792 ± 0.005	0.800 ± 0.006	0.801 ± 0.006
	$Prec$ ± SD	0.843 ± 0.012	0.845 ± 0.008	0.846 ± 0.010
	Rec ± SD	0.747 ± 0.019	0.760 ± 0.017	0.760 ± 0.016
Epithelial	F_1 ± SD	0.785 ± 0.009	0.787 ± 0.005	0.800 ± 0.003
	$Prec$ ± SD	0.858 ± 0.011	0.827 ± 0.007	0.868 ± 0.008
	Rec ± SD	0.723 ± 0.021	0.750 ± 0.007	0.741 ± 0.010
Stromal	F_1 ± SD	0.632 ± 0.003	0.634 ± 0.006	0.643 ± 0.006
	$Prec$ ± SD	0.563 ± 0.009	0.549 ± 0.017	0.584 ± 0.017
	Rec ± SD	0.721 ± 0.012	0.751 ± 0.018	0.716 ± 0.017
Other	F_1 ± SD	0.278 ± 0.023	0.247 ± 0.026	0.242 ± 0.023
	$Prec$ ± SD	0.545 ± 0.033	0.603 ± 0.036	0.560 ± 0.024
	Rec ± SD	0.189 ± 0.024	0.157 ± 0.021	0.155 ± 0.020

Table A.35: MIDOG++ precision values. Adapted from [116].

Models		CellViT$^{++}_{\text{SAM-H}}$	CellViT$^{++}_{\text{SAM-H}}$	CellViT$^{++}_{\text{SAM-H}}$
Organs	Origin	No Additional Cells	20 Additional Cells	200 Additional Cells
Breast Cancer	Human	0.73 ± 0.03	0.79 ± 0.01	0.77 ± 0.04
Neuroendocrine Tumor		0.47 ± 0.14	0.53 ± 0.03	0.54 ± 0.07
Melanoma		0.74 ± 0.21	0.75 ± 0.04	0.83 ± 0.03
Cutaneous Mast Cell Tumor	Canine	0.65 ± 0.14	0.73 ± 0.05	0.73 ± 0.04
Lung Cancer		0.36 ± 0.07	0.41 ± 0.04	0.42 ± 0.06
Lymphoma		0.61 ± 0.09	0.60 ± 0.05	0.59 ± 0.04
Soft Tissue Sarcoma		0.71 ± 0.03	0.75 ± 0.02	0.71 ± 0.03

Table A.36: MIDOG++ recall values. Adapted from [116].

Models		CellViT$^{++}_{\text{SAM-H}}$	CellViT$^{++}_{\text{SAM-H}}$	CellViT$^{++}_{\text{SAM-H}}$
Organs	Origin	No Additional Cells	20 Additional Cells	200 Additional Cells
Breast Cancer	Human	0.38 ± 0.05	0.42 ± 0.02	0.49 ± 0.03
Neuroendocrine Tumor		0.33 ± 0.08	0.39 ± 0.02	0.48 ± 0.06
Melanoma		0.56 ± 0.10	0.60 ± 0.03	0.62 ± 0.03
Cutaneous Mast Cell Tumor	Canine	0.63 ± 0.07	0.61 ± 0.03	0.68 ± 0.03
Lung Cancer		0.34 ± 0.06	0.41 ± 0.03	0.44 ± 0.02
Lymphoma		0.39 ± 0.05	0.44 ± 0.03	0.57 ± 0.05
Soft Tissue Sarcoma		0.43 ± 0.06	0.42 ± 0.01	0.48 ± 0.04

Table A.37: Overview of augmentation techniques used in the experiments. This table lists the augmentation techniques along with their corresponding Albumentations function names, the probability of application, and relevant parameters for each technique. Adapted from [116].

Augmentation Technique	Albumentations Function Name	Probability (p)	Parameters
Random Rotate 90	RandomRotate90	0.5	None
Horizontal Flip	HorizontalFlip	0.5	None
Vertical Flip	VerticalFlip	0.5	None
Downscale	Downscale	0.15	scale_max = 0.5, scale_min = 0.5
Blur	Blur	0.2	blur_limit = 10
Gaussian Noise	GaussNoise	0.25	var_limit = 50
Color Jitter	ColorJitter	0.2	brightness = 0.25, contrast = 0.25, saturation = 0.1, hue = 0.05
Superpixels	Superpixels	0.1	p_replace = 0.1, n_segments = 200, max_size = h/2
Zoom Blur	ZoomBlur	0.1	max_factor = 1.05
Random Sized Crop	RandomSizedCrop	0.1	min_max_height = (h/2, h), height = h, width = w

Table A.38: List of classical machine learning models evaluated using the PyCaret automated machine learning (AutoML) framework for the Lizard dataset. CatBoost classifier (marked bold) was the best performing model among all tested. Adapted from [116].

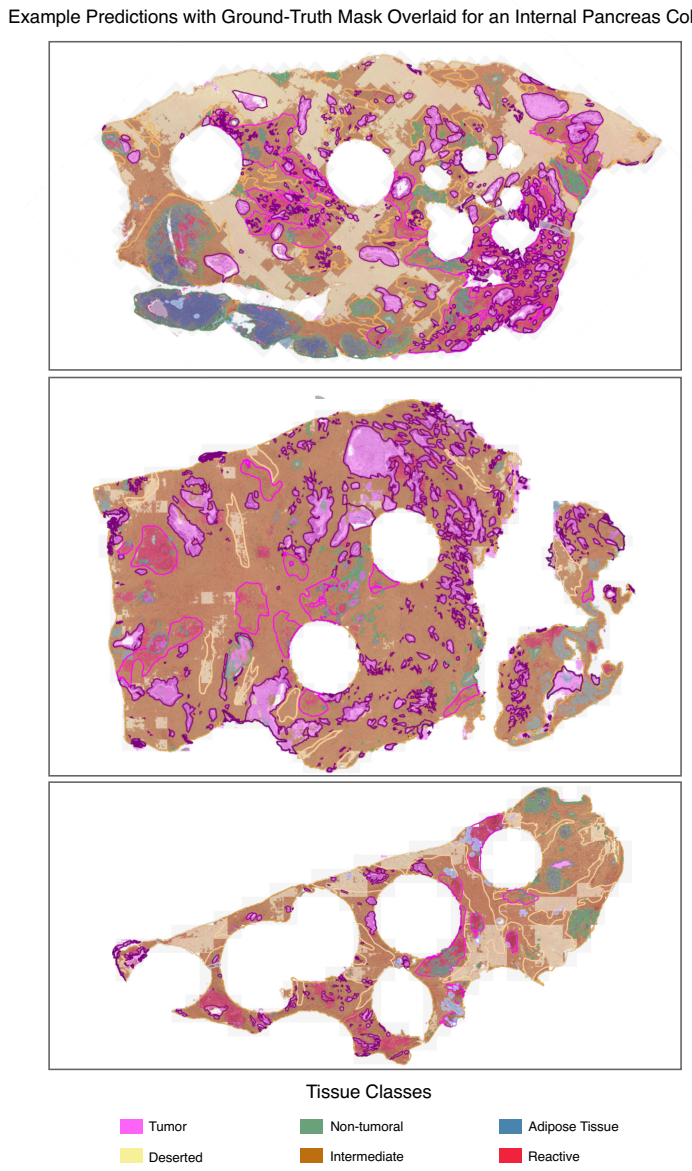
	Model
1	Logistic Regression
2	Linear Discriminant Analysis
3	Ridge Classifier
4	Ada Boost Classifier
5	Gradient Boosting Classifier
6	**CatBoost Classifier**
7	Extreme Gradient Boosting
8	Light Gradient Boosting Machine
9	Quadratic Discriminant Analysis
10	Random Forest Classifier
11	Decision Tree Classifier
12	Extra Trees Classifier
13	K Neighbors Classifier
14	Naive Bayes
15	SVM - Linear Kernel

Table A.39: Important Resources and identifiers used for the CellViT++ experiments. Comparison methods just include methods trained or evaluated by us.

Resource	Source	Identifier	Reference
CellViT^{++}			-
Albumentations 1.3.0	Pip	albumentations.ai	Buslaev et al. [24]
CatBoost 1.2.5	Pip	github.com/catboost/catboost	Dorogush et al. [66]
cuCIM 24.04.00	Rapidsai (conda channel)	github.com/rapidsai/cucim	-
CuPY 13.2.0	Conda-forge (conda channel)	cupy.dev	Okuta et al. [209]
GeoJSON 3.0.0	Pip	python-geojson.readthedocs.io/en/latest	-
huggingface-hub 0.22.2	Pip	huggingface.co	-
Numba 0.59.0	Pip	numba.pydata.org	Lam et al. [155]
NumPy 1.23.5	Pip	numpy.org	Harris et al. [105]
OpenSlide 4.0.0	Conda-forge (conda channel)	openslide.org	Goode et al. [91]
openslide-python 1.3.1	Pip	openslide.org/api/python	-
opencv-python-headless 4.5.4.58	Pip	opencv.org	Bradski [20]
pandarallel 1.6.5	Pip	github.com/nalepae/pandarallel	-
pandas 1.4.3	Pip	pandas.pydata.org	The pandas development team [260]
PathoPatch 1.0.4b0	Pip	github.com/TIO-IKIM/PathoPatcher	Hörst et al. [117]
Pillow 10.3.0	Conda-forge (conda channel)	pillow.readthedocs.io	-
PyCaret 3.3.2	Pip	pycaret.org	Ali [2]
Python 3.10.14	Conda-forge (conda channel)	www.python.org	-
Ray 2.9.3	Pip	www.ray.io	Moritz et al. [199]
scikit-base 0.7.8	Pip	scikit-learn.org	Pedregosa et al. [220]
scikit-image 0.19.8	Pip	scikit-image.org	Van der Walt et al. [264]
scikit-learn 1.3.0	Pip	scikit-learn.org	Pedregosa et al. [220]
scipy 1.8.1	Pip	scipy.org	Virtanen et al. [269]
timm 1.0.8	Pip	timm.fast.ai	Wightman [283]
torch 2.2.1	Pip	pytorch.org	Paszke et al. [214]
torchmetrics 0.11.4	Pip	lightning.ai/docs/torchmetrics	Detlefsen et al. [62]
torchvision 0.17.1	Pip	pytorch.org	-
ujson 5.8.0	Pip	pypi.org/project/ujson	-
WandB 0.15.4	Pip	wandb.ai	-
Wsidicom 0.20.4	Pip	github.com/imi-bigpicture/wsidicom	-
Wsidicomizer 0.13.2	Pip	github.com/imi-bigpicture/wsidicomizer	-
XGBoost 2.1.1	Pip	xgboost.readthedocs.io	Chen et al. [47]
Comparison Methods			-
SoftCTM	GitHub	github.com/lely475/SoftCTM (commit: 8918beafd7d5a36695d1bbdb5bb8d6139376a4dc)	Schoenpflug et al. [241]
HoVer-Net	GitHub	github.com/vqdang/hover_net (commit:67e2ce5e3f1a64a2ece77ad1c24233653a9e0901)	Graham et al. [95]
Cerberus	GitHub	github.com/TissueImageAnalytics/cerberus (commit: 5bcecbb071bebd5911250034c94f3568f23f50bb)	Graham et al. [97]
TIAToolBox	GitHub	github.com/TissueImageAnalytics/tiatoolbox (commit: c180566bbe7ec04a9b91924748acf2d03f6302d9)	Pocock et al. [222]
PointNu-Net	GitHub	github.com/Kaiseem/PointNu-Net (commit: 747f5019df5f611e81a823e5318a2fa0b60e2571)	Yao et al. [295]
Nucleiio	GitHub	github.com/huangzhii/nuclei.io (commit: 78d52270eaeb05bc26f9b134231431a04a837b22)	Huang et al. [121]
CGIS-CPF			-
Foundation Models			-
HIPT	GitHub	github.com/mahmoodlab/HIPT (commit: 7336ee7d4fc70a348358cab31984cba093a77983)	Chen et al. [44]
UNI	Hugging Face	huggingface.co/MahmoodLab/UNI (commit: ba5018a94088b378720cd07995efe65a79c6b952)	Chen et al. [45]
Virchow	Hugging Face	huggingface.co/paige-ai/Virchow (commit: b80411ffe80f1d3070879e512ffb0152d7997377)	Vorontsov et al. [270]
Virchow 2	Hugging Face	huggingface.co/paige-ai/Virchow2 (commit: a8536e8a8dd3cd0b200aa44be674ef95d2ad1598)	Zimmermann et al. [303]
Segment Anything	GitHub	github.com/facebookresearch/segment-anything	-
Datasets			-
Ocelot	Zenodo	zenodo.org/records/8417503	Ryu et al. [235]
Midog++	Figshare	doi.org/10.6084/m9.figshare.c.6615571.v1	Aubreville et al. [6]
CoNSeP	HoVer-Net	warwick.ac.uk/fac/cross_fac/tia/data/hovernet	Graham et al. [95]
Lizard	-	warwick.ac.uk/fac/cross_fac/tia/data/lizard	Graham et al. [96]
NuCLS	-	sites.google.com/view/nucls/home	Amgad et al. [5]
PanopTILs	-	sites.google.com/view/panoptils	Liu et al. [174]
SegPath	-	dakomura.github.io/SegPath/ zenodo.org/record/7412529 zenodo.org/record/7412500	Komura et al. [146]

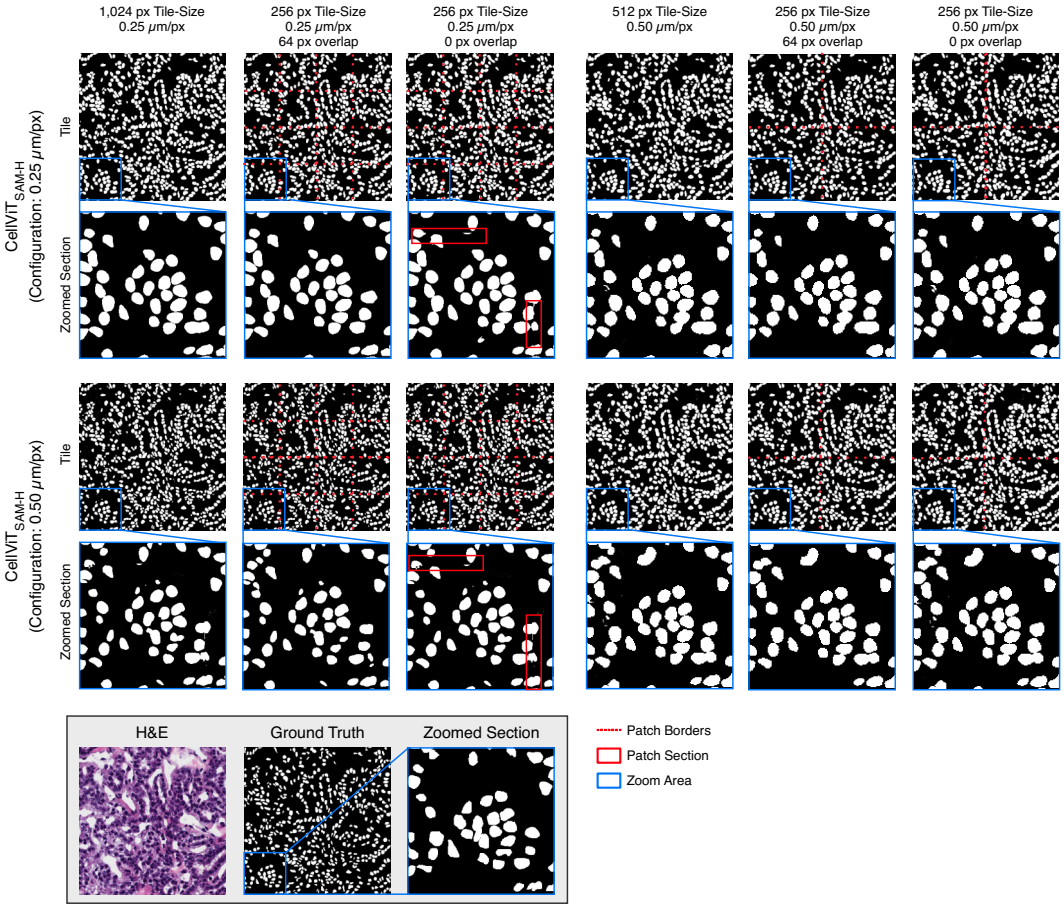
Supplementary Figures

Whole Tissue Segmentation

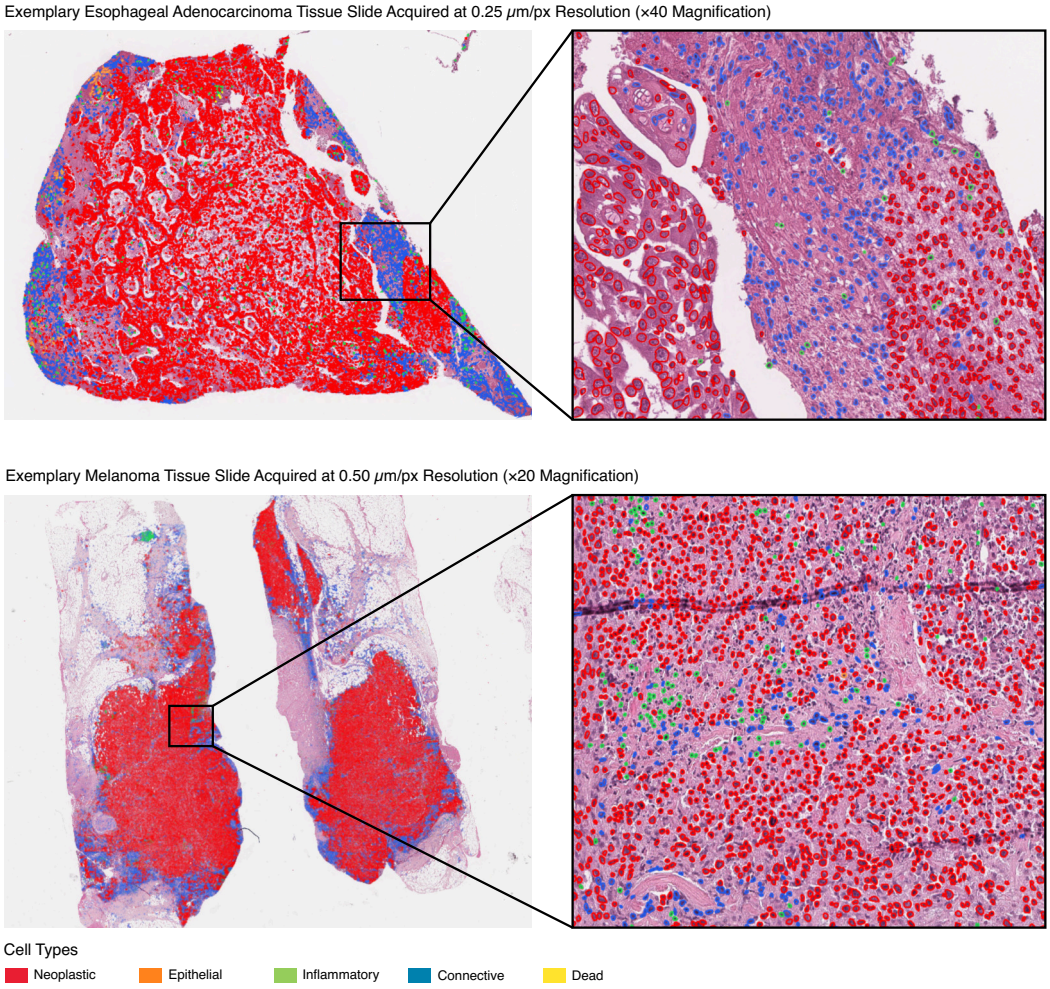


A.1: Exemplary samples from our internal pancreatic cancer test dataset for the MAF experiments. Provided are the ground truth region outlines (marked as solid lines) together with model predictions (region overlay).

CellViT: A Novel Approach to Cell Segmentation

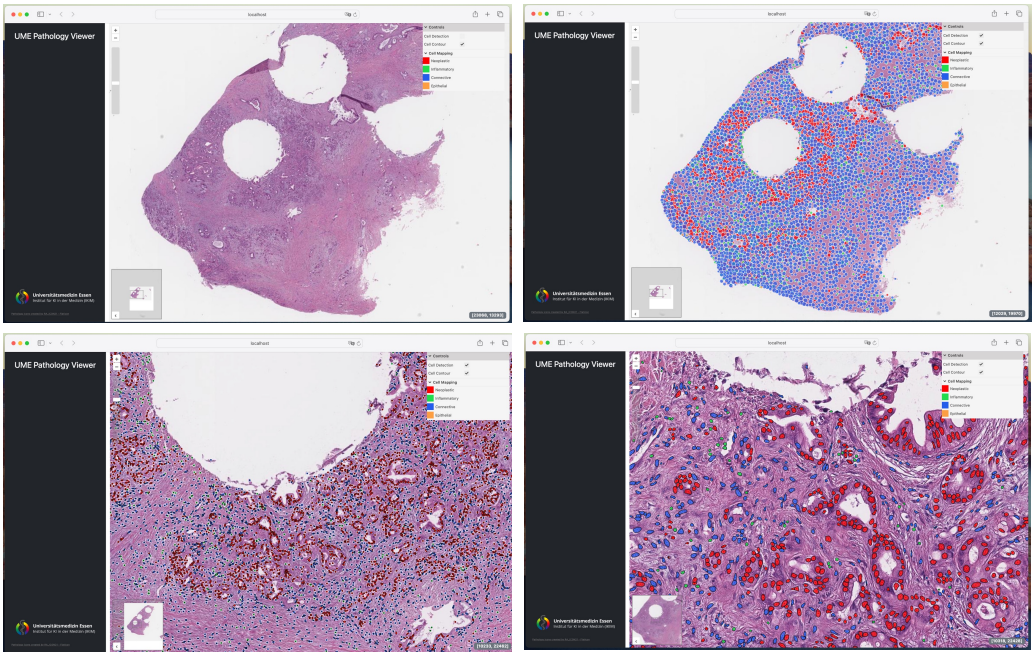


A.2: Example of one MoNuSeg tissue sample with ground truth binary masks and predictions of the CellViT$_{\text{SAM-H}}$ model for different input sizes and magnifications. Adapted from [119].

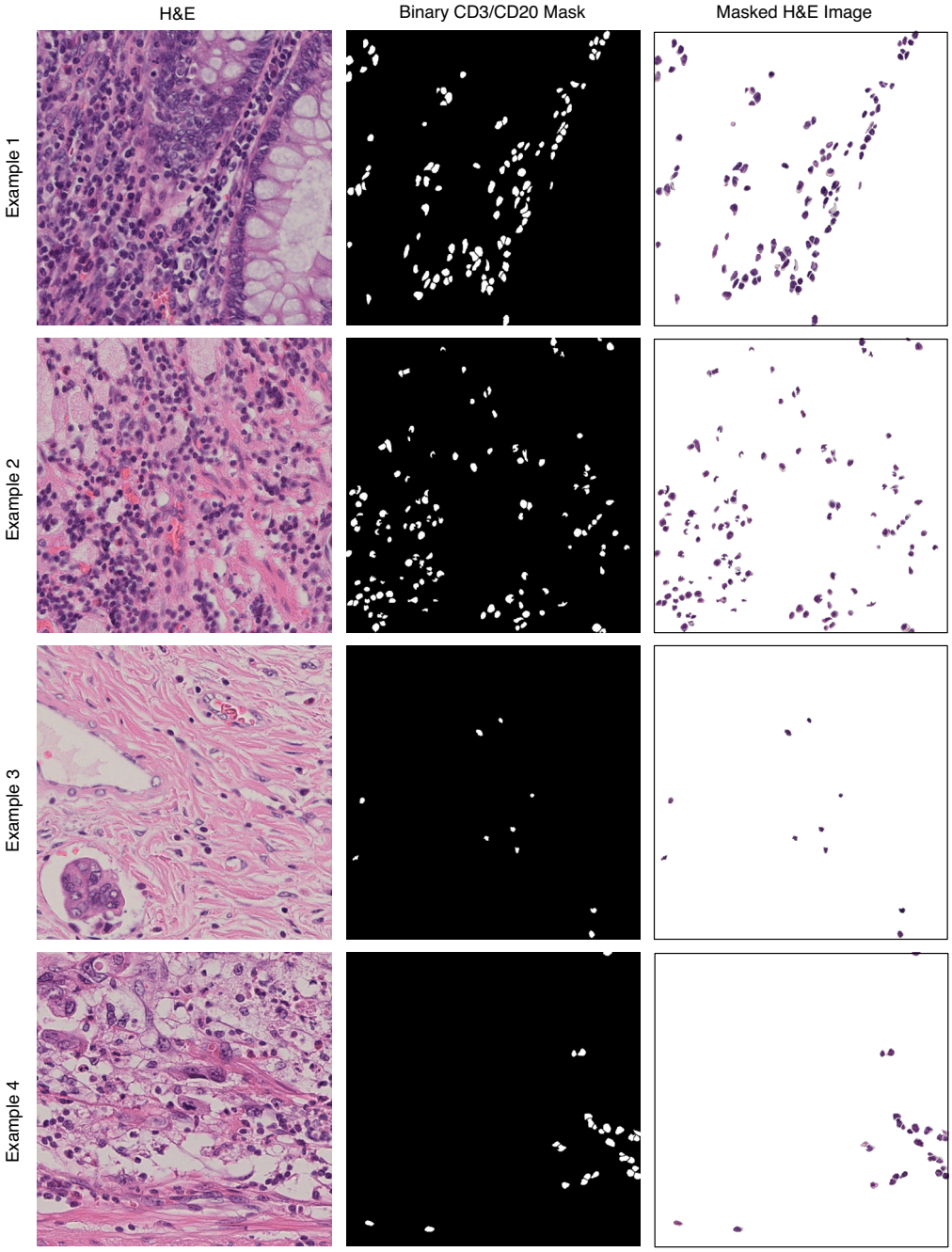


A.3: Exemplary WSI files with corresponding cell polygons imported into QuPath to show the interoperability of our inference pipeline. For each of the files, approximately $150{,}000$ nuclei have been detected, which can be imported into QuPath without any performance problems regarding fast file loading and zooming on a standard laptop. The first WSI file was acquired at a magnification of $\times 40$ with $0.25\ \mu\mathrm{m/px}$, the second at $\times 20$ with $0.50\ \mu\mathrm{m/px}$. Results have been derived with the $\text{CellViT}_{\text{SAM-H}}$ model. Adapted from [119].

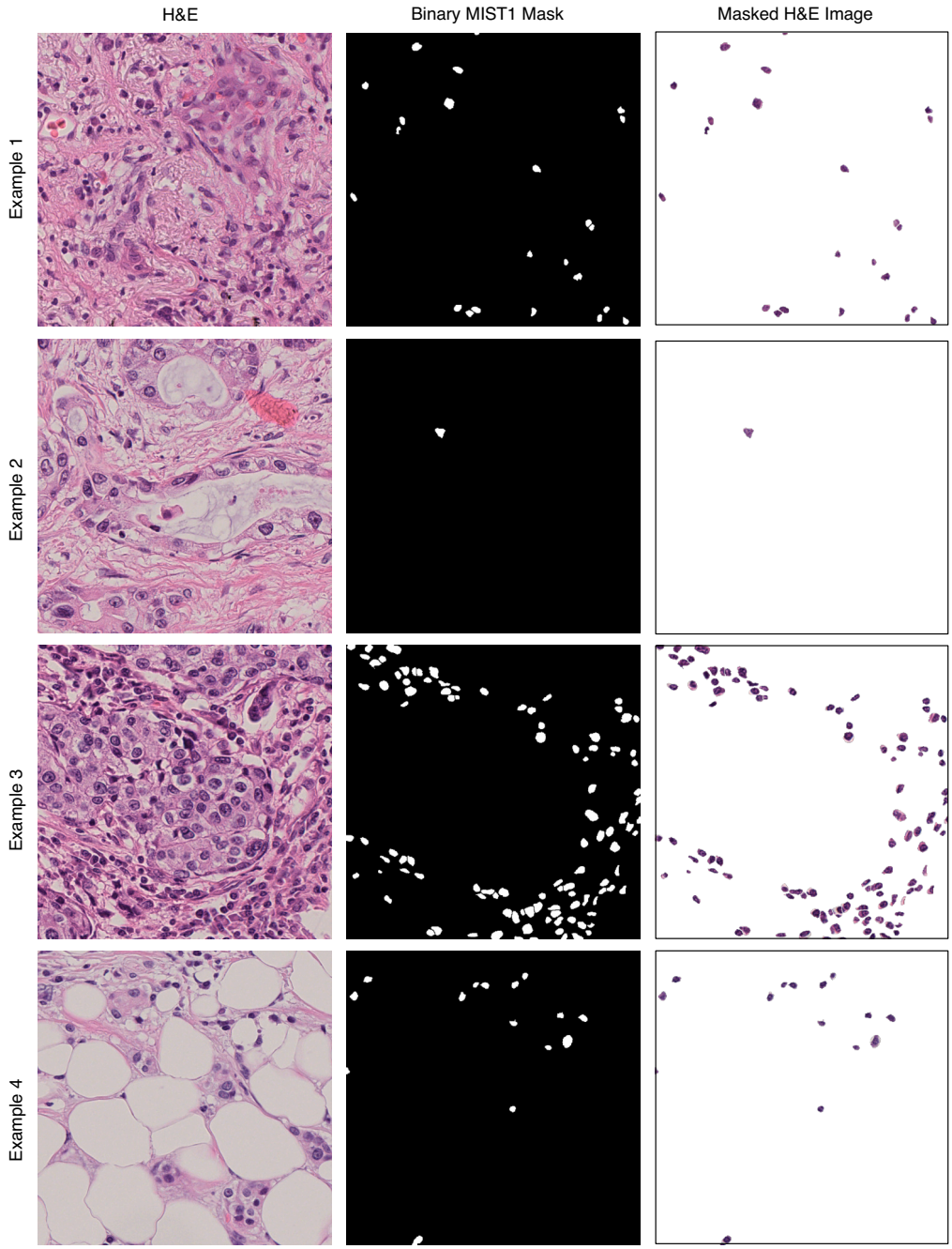
CellViT++: Enhancing Cellular Analysis Capabilities



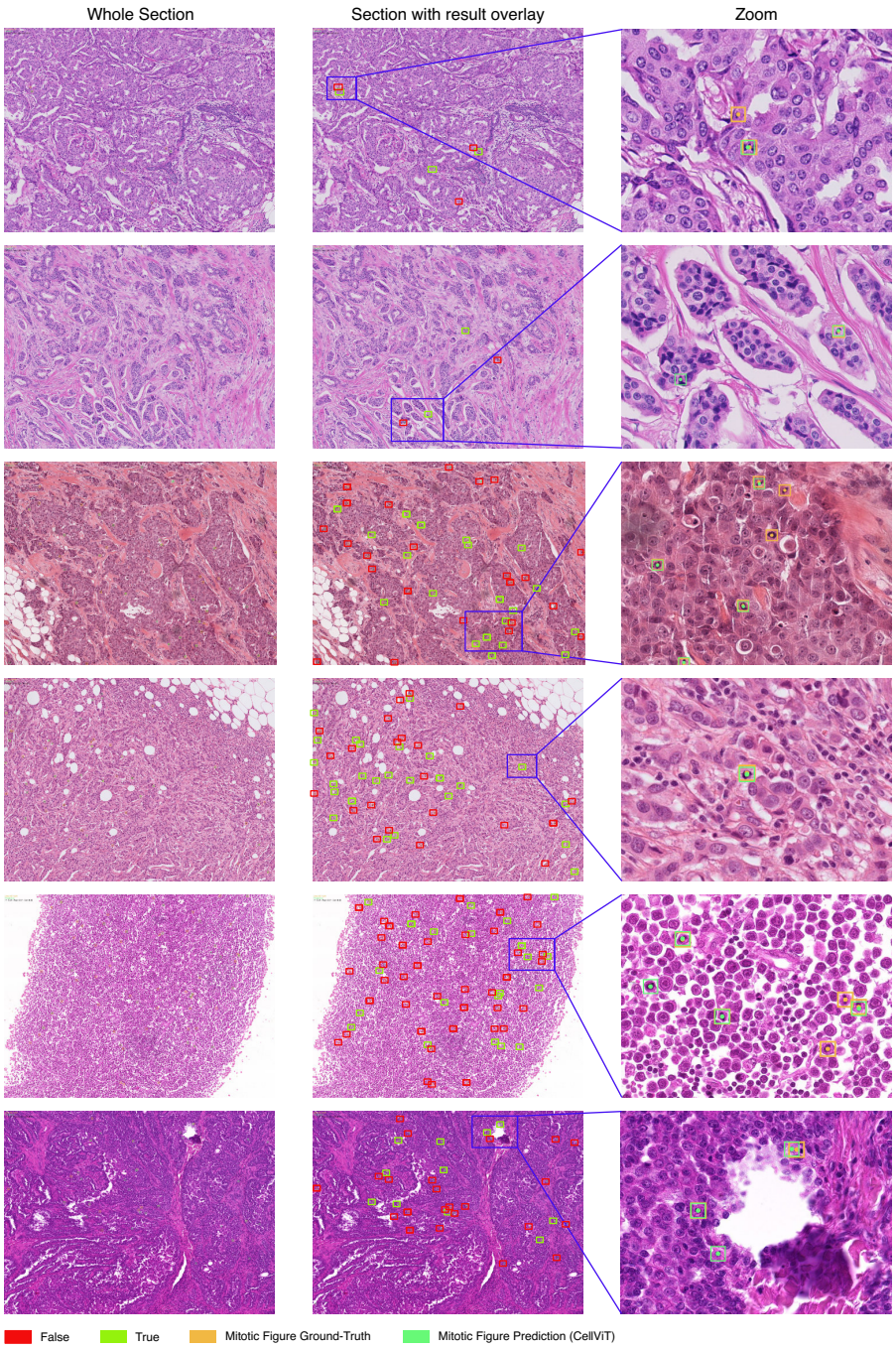
A.4: Web-based WSI viewer demonstrating visualization capabilities without local software installation using modern web technologies. (Upper left) General overview of the WSI. (Upper right) Overview with cell detections, clustered to depict spatial distribution. (Lower left) Zoomed-in view of a selected region with detailed cell detections. (Lower right) High-resolution view with cell contour overlays. Adapted from [116].



A.5: Representative tissue sections from the SegPath dataset with CD3/CD20 staining masks.



A.6: Representative tissue sections from the SegPath dataset with MIST1 staining masks.



A.7: Representative tissue sections from the MIDOG++ dataset, illustrating the challenge of mitotic figure detection as a "needle in a haystack" problem. Each WSI section is annotated with ground truth (GT) mitotic figures and corresponding predictions generated by the CellViT$^{++}_{\text{SAM-H}}$ model. Adapted from [116].

Zur Person:

_____ _____
Vorname Name geboren am/ in

_____
Matrikelnummer

Belehrung:

Die Abgabe einer eidesstattlichen Versicherung ist eine nach §§ 156, 161 Strafgesetzbuch (StGB) strafbewehrte Bestätigung der Richtigkeit einer Erklärung. Die Abgabe einer falschen oder unvollständigen Versicherung an Eides statt ist strafbar.

Wer vorsätzlich eine falsche Versicherung an Eides statt abgibt, kann mit einer Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft werden, § 156 StGB. Die fahrlässige Abgabe einer falschen Versicherung an Eides statt kann mit einer Freiheitsstrafe bis zu einem Jahr oder Geldstrafe bestraft werden, § 161 StGB.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

_____ _____
Ort, Datum Unterschrift

Eidesstattliche Versicherung

In Kenntnis der Bedeutung einer eidesstattlichen Versicherung und der Strafbarkeit der Abgabe einer falschen eidesstattlichen Versicherung versichere ich hiermit an Eides statt, dass ich die vorliegende Dissertation mit dem Titel

_____
_____
_____

selbstständig und ohne unzulässige fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht.

_____ _____
Ort, Datum Unterschrift