Università degli Studi di Napoli Federico II
Ph.D. Program in
Information and Communication Technology for Health
XXXVI Cycle

Thesis for the Degree of Doctor of Philosophy

Distilled Nuclei Segmentation For Graph-Based Whole Slide Images Analysis and Retrieval

by
Cristian Tommasino

Advisor: Prof. Antonio Maria Rinaldi

Co-advisors: Prof. Stefania Staibano and Prof. Francesco Merolla

Scuola Politecnica e delle Scienze di Base
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

If we knew what it was we were doing,
it would not be called research,
would it?

- Albert Einstein

Distilled Nuclei Segmentation For Graph-Based Whole Slide Images Analysis and Retrieval

Ph.D. Thesis presented
for the fulfillment of the Degree of Doctor of Philosophy
in Information and Communication Technology for Health
by

Cristian Tommasino

October 2023

Approved as to style and content by

_____

Prof. Antonio Maria Rinaldi, Advisor

_____

Prof. Stefania Staibano and Prof. Francesco Merolla, Co-advisors

Università degli Studi di Napoli Federico II
Ph.D. Program in Information and Communication Technology for Health
XXXVI cycle - Chairman: Prof. Daniele Riccio

http://icth.dieti.unina.it

Candidate's declaration

I hereby declare that this thesis submitted to obtain the academic degree of Philosophiæ Doctor (Ph.D.) in Information and Communication Technology for Health is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.
Parts of this dissertation have been published in international journals and/or conference articles (see list of the author's publications at the end of the thesis).

Napoli, December 13, 2023

_____

Cristian Tommasino

Abstract

The integration of technological advancements, particularly in data processing and machine learning, has profoundly impacted the trajectory of medicine and healthcare evolution. Pathology, an essential pillar of medical diagnosis, has not been exempt from this technological metamorphosis. The conventional methodology of microscope-based histological tissue analysis has undergone a significant transition, culminating in the birth of digital pathology. This digitization has not only enhanced the operational efficiency of pathologists but has also facilitated the inception of an interdisciplinary domain termed computational pathology. This discipline employs computational methodologies to analyze and model histopathological imagery meticulously. The overarching objective of computational pathology is to architect a robust digital diagnostic infrastructure functioning as a Computer-Aided Diagnosis (CAD) system. This paradigmatic shift harbors the potential to radically transform the methodologies employed in the diagnosis and therapeutic interventions of diseases, with a particular emphasis on oncological conditions. The salience of computational pathology is accentuated by its prospective capacity to engender transformative alterations in the diagnostic and therapeutic modalities of oncological diseases. Given the accelerated advancements in deep learning paradigms and computer vision algorithms, in conjunction with the seamless integration of data derived from digital pathology, computational pathology is poised at the precipice of a vast paradigmatic evolution. In this regard, we present two primary contributions: the first focuses on enhancing Picture Archiving and Communication Systems (PACSs), and the second aims to reduce the time complexity analysis of Whole Slide Images (WSIs). Our first contribution concerns employing deep features in a Content-Based Image Retrieval (CBIR) system that employs deep features to improve retrieval in PACSs. This contrasts with prevalent methods typically using

metadata to retrieve WSIs or their segments. Our approach uses visual features to improve retrieval, enabling query by example. Our second contribution encompasses two key improvements: one concerns nuclei instance segmentation and classification, and the other focuses on cell-graph representation and classification. Specifically, we introduce Fast-HoVerNet, a distilled version of HoVerNet that is one of the most used networks for nuclei instance segmentation and classification, which offers less inference time while maintaining comparable segmentation and classification capabilities. Employing Fast-HoVerNet, we introduce a novel method to extract cell features in cell-graph representation, which is faster than existing ones, and unifies the process using a single network for both cell detection and representation. Finally, we prove the effectiveness of our cell graph representation using it for cancer subtype classification. Experimental results demonstrate that our approach achieves results comparable to the current state-of-the-art ones but with reduced time complexity.

Keywords: Computational Pathology, Cell-Graph Representation, Graph Neural Network, Content-Based Image Retrieval, Cell-Graph Classification

Sintesi in lingua italiana

L'integrazione di avanzamenti tecnologici, in particolare nei settori dell'elaborazione dei dati e dell'apprendimento automatico, ha profondamente influenzato l'evoluzione della medicina e dell'assistenza sanitaria. La patologia, pilastro essenziale della diagnosi medica, non è stata esente da questa metamorfosi tecnologica. La metodologia convenzionale di analisi istologica dei tessuti basata su microscopio ha subito una significativa transizione, culminando nell'emergenza della patologia digitale. Questa digitalizzazione non solo ha migliorato l'efficienza operativa dei patologi, ma ha anche facilitato la nascita di un dominio interdisciplinare chiamato patologia computazionale. Questa disciplina si propone di utilizzare metodologie computazionali per analizzare e modellare meticolosamente le immagini istopatologiche. L'obiettivo principale della patologia computazionale è di creare un'infrastruttura diagnostica digitale robusta che funzioni come un sistema di Diagnosi Assistita dal Computer (CAD). Questo cambiamento di paradigma ha il potenziale di trasformare radicalmente le metodologie utilizzate nella diagnosi e negli interventi terapeutici delle malattie, con un particolare focus sulle condizioni oncologiche.

L'importanza della patologia computazionale è accentuata dalla sua capacità potenziale di generare trasformazioni nella diagnosi e nelle modalità terapeutiche delle malattie oncologiche. Data la rapida evoluzione nell'apprendimento profondo e negli algoritmi di visione artificiale, in congiunzione con l'integrazione senza soluzione di continuità dei dati derivati dalla patologia digitale, la patologia computazionale è all'avanguardia di una grande evoluzione paradigmatica.

A questo proposito, presentiamo due principali contributi: il primo si concentra sul miglioramento dei Sistemi di Archiviazione e Comunicazione delle Immagini (PACSs), mentre il secondo mira ad accelerare il processo di analisi per Whole Slide Images (WSIs).

Come primo contributo, introduciamo un sistema di recupero di immagini basato sul contenuto che utilizza deep feature per miglioare il recupero della immagini nei PACSs. Questo si contrappone ai metodi che tipicamente utilizzano metadati per recuperare WSIs o le loro parti. Il nostro approccio fa uso elementi visivi per migliorare l'accuratezza, permettendo una query basata su esempi.

Il nostro secondo contributo comprende due miglioramenti chiave: uno riguarda la segmentazione e classificazione dell'istanza dei nuclei e l'altro al rappresentazione e classificazione del cell-graph. In particolare, introduciamo Fast-HoVerNet, una versione efficiente di HoVerNet, la quale è ampiamente utilizzata per la segmentazione e classificazione di nucli, che offre tempi di inferenza più rapidi mantenendo capacità comparabili di segmentazione e classificazione. Utilizzando Fast-HoVerNet, suggeriamo un nuovo metodo per estrarre feature cellulari per la rappresentazione ael cell-grpah, che è più veloce e consolida il processo utilizzando una singola rete sia per la rilevazione delle celule che per la loro rappresentazione. Infine, proponiamo di utilizzare la nostra rappresentazione per la classificazione dei sottotipi di cancro.

I risultati sperimentali confermano che i nostri approcci raggiungiono risultati paragonabili allo stato dell'arte per ciascun task trattato, ma con una complessità temporale ridotta.

Parole chiave: Computational Pathology, Cell-Graph Representation, Graph Neural Network, Content-Based Image Retrieval, Cell-Graph Classification

Acknowledgements

I want to express my deepest gratitude to my advisors, Professors Antonio Maria Rinaldi for his invaluable guidance that was essential to my research success, and Francesco Merolla of the Department of Medicine and Health Sciences "V. Tiberio" at the University of Molise, as well as Stefania Staibano of the Department of Advanced Biomedical Sciences at the University "Federico II" of Naples for their support.

I am also grateful to Professor Francesco Ciompi for welcoming me into the Diagnostic Images Analysis Group (DIAG) at Radboud University Medical Center. This opportunity to work and live there was a pivotal experience in my journey.

Thanks to Cristiano Russo for his support, particularly in the final month. I'm equally thankful to Andrea Mancuso, Adriano Masono, and the entire "Ricerca Operativa" group for their warm hospitality.

I want to acknowledge all the new friends I've made during my Ph.D. journey and my travel companions Eleonora, Antonio, Enrica, Luca, and Enzo. Their company made the long train waits much more enjoyable.

Lastly, I thank Professor Daniele Riccio for excellently coordinating the Ph.D. Program.

Finally, my most profound appreciation goes to my parents and Maria Giovanna. Their support, understanding, infinite patience, and encouragement were my pillars, especially when needed.

Contents

List of Acronyms

The following acronyms are used throughout the thesis.

ADH	Atypical Ductal Hyperplasia
AI	Artificial Intelligence
ANN	Artificial Neural Network
BACH	BreAst Cancer Histology
BRACS	BReAst Carcinoma Subtyping
BRCA	BReast CAncer
CAD	Computer-Aided Diagnosis
CBIR	Content Based Image Retrieval
CCG	Cluster-Centroid-Graph
CG	Cell-Graph
CNN	Convolutional Neural Network
CoNSeP	Colorectal Nuclear Segmentation and Phenotypes
CPATH	Computational Pathology

CRC Colorectal Cancer

CSN Color Stain Normalization

DCAN Deep Contour-Aware Network

DCIS Ductal Carcinoma in Situ

DICOM Digital Imaging and Communications in Medicine

DL Deep Learning

DNA DeoxyriboNucleic Acid

DNN Deep Neural Network

DP Digital Pathology

FDA Food and Drug Administration

FEA Flat Epithelial Atypia

GCN Graph Convolutional Network

GIN Graph Isomorphism Network

GNN Graph Neural Network

HACT HierArchical Cell-to-Tissue

HACT-Net HierArchical Cell-to-Tissue Network

H&E Hematoxylin and Eosin

HER2 Human Epidermal growth factor Receptor 2

HV Horizontal and Vertical

IC Invasive Carciname

JK	Jumping Knowledge
KD	Knowledge Distillation
KNN	k-Nearest Neighbors
LSTM	Long Short Term Memory
ML	Machine Learning
MLP	Multi Layer Perceptron
MPNN	Message Passing Neural Network
N	Normal Tissue
NC	Nuclear Classication
NMF	Non-negative Matrix Factorization
NP	Nuclear Pixel
OD	Optical Density
PACS	Picture Archiving and Communication System
PB	Pathological Benign
PG	Patch-Graph
PNA	Principal Neighbourd Aggregator
ReLU	Rectified Linear Unit
RNA	RiboNucleic Acid
RNN	Recurrent Neural Network
RQ	Research Question

SOTA	State-of-the-art
SVD	Single Value Decomposition
TG	Tissue-Graph
TCGA	The Cancer Genome Atlas
TRoI	Tissue Region Of Interest
UDH	Usual Ductal Hyperplasia
WSI	Whole Slide Imaging
WSI	Whole Slide Image
XAI	eXplainable Artificial Intelligence

List of Figures

List of Tables

Chapter 1

Introduction

Medical diagnostics has always relied on pathology as an essential discipline. It has continuously adapted to meet the changing needs of modern medicine, achieving significant milestones such as the adoption of microscopy in the 19th century and the rise of molecular pathology in the 21st century. Nowadays, we are catching the need for two significant developments: Digital Pathology (DP) and Computational Pathology (CPATH). DP improves diagnostic accuracy and reach by leveraging digital technology to evolve traditional microscopy. Meanwhile, CPATH represents a deeper shift by using artificial intelligence to mine subvisual data from histopathology images, enabling predictions and insights beyond human perception. This chapter provides a comprehensive exploration of CPATH, starting from the basics of glass slide images and moving on to applying Artificial Intelligence (AI) techniques.

1.1 From Staining to Computational Analysis

The pathology field, which investigates diseases' causes and impacts, has experienced changes throughout its history. However, the fundamental goal of obtaining a tissue diagnosis by interpreting macroscopic and microscopic morphology has remained steadfast. The introduction of microscopy in the mid-19th century [82] marked a significant shift in the field, leading to a histology-based understanding of diseases, known as histopathology, which transformed medicine, as shown from the timeline in Figure 1.1. Im-

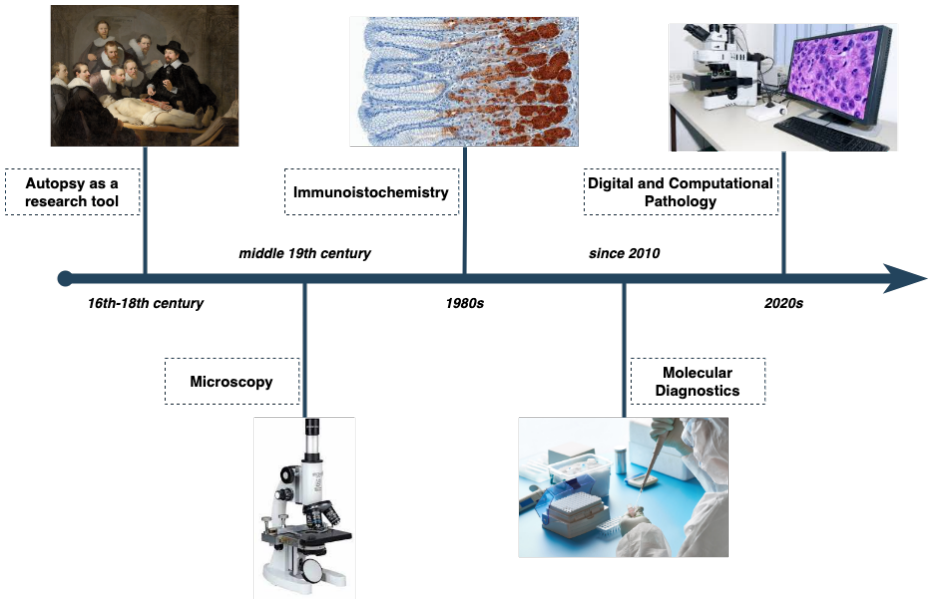


Figure 1.1. Pathology timeline

munohistochemistry emerged in the 1980s, utilizing antibodies to identify specific antigens in tissue samples. Furthermore, the 2010s saw the emergence of molecular pathology, which exceeds the scope of the microscope to analyze nucleic acids using various methods [25, 80]. Nowadays, DP is directing in another era of transformation in the field. It revolutionizes primary diagnosis work, consultations, education, and multidisciplinary conferences. Furthermore, DP lays the groundwork for CPATH, a big-data approach that employs AI to extract data from images [51, 1]. In the following subsection, we will outline the complete process, from tissue staining to the complexities of computational pathology.

1.1.1 From tissue to digital Hematoxylin and Eosin slide

The staining process begins with removing the tissue via biopsy or surgery to obtain a digital Hematoxylin and Eosin (H&E) slide from the tissue. Once removed, the tissue must undergo a series of steps to become a stained raw slide, as described in [21, 28]. The first four steps preserve the

tissue structure, while the last step stains the tissue slices. After obtaining a raw tissue slide, it is scanned to create a digital copy. The first step is fixation, where chemicals preserve and protect the tissue structure from irreversible protein cross-linking. Common fixatives include Neutral Buffered Formalin or Paraffin-formalin. The second step is dehydration, where the sample is treated with ethanol to remove water, followed by xylene to remove the ethanol residue. During the embedding stage, the sample is placed in paraffin wax or plastic resin to facilitate the extraction of cellular structures, which generally inhibits antibody penetration. H&E, the sample is sectioned by mounting it on a microtome and cutting it into thin slices with an optimal thickness of around 4-5 micrometers. H&E staining is crucial in microbiology and histology processes for visualizing cells and cellular components under a microscope [14]. This process involves applying hematoxylin, a basic dye that bleaches acidic structures to blue. The structures that attract this dye, called basophilic structures, include DeoxyriboNucleic Acid (DNA) in cell nuclei, RiboNucleic Acid (RNA) in ribosomes, and rough endoplasmic reticulum structures. Eosin, an acid dye, is then applied as a counterstain, which targets the basic structures and stains them pink. Eosinophilic structures that attract eosin are called eosinophilic structures; the cytoplasm is an example of an eosinophilic structure [28]. The staining process involves immersing the tissue sections in solutions of the stains, followed by a series of washes to remove excess stains. The stained sections are dehydrated, cleared, and mounted with a coverslip for microscopic examination. Finally, the stained tissue sections are examined under a light microscope, providing invaluable information for medical diagnosis and scientific study.

1.1.2 Digitalization Process

The process of digitizing slides encloses a four-step paradigm: image acquisition through scanning, image storage, image editing, annotation, and finally, image display. The scanning of slides is executed using dedicated slide scanner devices, with notable models from Leica Biosystems and Philips receiving Food and Drug Administration (FDA) clearance for clinical diagnosis. To show the main difference between each company in terms of slide format, we reported in Table 1.1 a list of major companies with their scanner models and supported slide formats. A Whole Slide

Imaging (WSI) scanner is intricately composed of four major components: a light source, a slide stage, an objective lens, and a digital camera. Appropriately selecting focal points across the slide is imperative to generate in-focus Whole Slide Images (WSIs). The individual tiles or linear scans captured are subsequently stitched together to formulate the comprehensive image. However, the digitalization process is not devoid of challenges. Variations in color and other discrepancies introduced during scanning necessitate careful consideration, especially given the diversity in scanner models and vendors. Addressing these variations is crucial for developing and deploying Computer-Aided Diagnosis (CAD) tools.

The slide scanning put in outputs a virtual microscope called WSI [58, 59]; some examples are reported in Figure 1.2. WSI technology, first described by Ferreira et al. in 1997 [22], digitizes whole tissue slides into high-resolution images. WSI systems consist of four main elements: a light source, a microscope with multiple lenses, a digital camera, and a system for repositioning the camera view along the sample. These systems capture high-resolution images (in the range of gigapixels) and can scan with different magnitudes, most commonly x20 or x40. WSI scanners can use one or several imaging modes, including bright-field, fluorescent, and multispectral imaging. Each method highlights different anatomical structures or physiological events in the tissue and uses various light sources. To display a two-dimensional image on a computer, a multi-resolution pyramid method of organization is used instead of a single-frame organization due to limitations when dealing with very high-resolution images. This method involves replicating the image in multiple resolutions, dividing each level into two-dimensional blocks of pixels of the same size called tiles. Only the tiles of a pyramid level in the viewing area will be loaded into memory. However, not all software supports pyramid organization.

1.1.3 Digital Pathology

In pathology, a paradigm shift is underway, transitioning from traditional reliance on conventional microscopes and glass slides to digital transformation. This digital shift, however, has encountered a slower pace than initially anticipated, with challenges such as high initial costs, lack of standardization, regulatory constraints, and other hurdles impeding rapid adoption. DP has been divided with the field of radiology, which experi-

Company	Scanner Model	Slide Format
Leica Biosystems	Aperio AT2/CS2/GT450	TIFF(SVS)
Hamamatsu	Nanooomer SQ/S60/S360/S210	JPEG
F. Hoffmann-La Roche AG	Ventana DP200/iScan HT/iScan Coreo	BIF, TIFF, JPG2000, DICOM
Huron Digital Pathology	TissueScope IQ/LE/LE120	BigTIFF, DICOM compliant
Philips	Ultra-Fast Scanner	iSyntax Philps proprietary file
3DHistech	Pannoramic Series	MEXS, JPG, JPEF
Mikroscan Technologies	SL5	TIFF
Olympus	SL5	JPEG, vsi, TIFF
Akoya Biosciences	Sakura VisionTek	BigTIFF, TIFF, JPG2000
Meyer Instruments	Easyscan PRO 6	SVS, MDS, JPEG, JPEG2000
Kfbio	KF-PRO	JPEG, JPEG2000, BMP, TIFF
Motic	Easyscan PRO	JPEG, JPEG2000, Aperio compatible
Precipoint	PreciPoint O8	GTIF
Zeiss	Zeiss Axio	Not specified
Objective Imaging Glissando	SVS, BigTIFF	
Microvisioneer	manual WSI	Not specificed

Table 1.1. A list of major companies with their WSI scanner models and supported formats

enced digitization several decades ago. However, distinct differences are evident between the two domains [54, 36, 61, 16]. Microscopic images in pathology inherently have larger dimensions than radiology images, necessitating augmented resources. While radiology has streamlined its workflow by eliminating films, pathology continues to produce glass slides, thereby inheriting the additional task of scanning and managing digital data. The adoption of a universal image standard, Digital Imaging and Communications in Medicine (DICOM), in radiology contrasts the prevailing proprietary standards in pathology, which remain under development [34, 15]. DP is characterized as the practice of pathology employing digital imaging technology, encompassing the acquisition, management, sharing, and interpretation of pathology information, including slides and data [56]. It extends beyond the confines of microscopy, incorporating macroscopic and molecular information. The origins of DP are rooted in virtual microscopy and telepathology, with WSI serving as a significant catalyst for its advancement [20, 57]. The advantages of DP over traditional methodologies are manifold, including expedited case delivery, the capability for simultaneous case viewing by multiple colleagues, direct annotation functionalities on digital slides, enhanced access to patients' historical data, reliable storage mechanisms, rapid image transmission for several applications, and augmented presentations for multidisciplinary tumor boards. DP manifests its utility across many clinical and non-clinical scenarios, demonstrating high concordance with traditional glass slides for primary diagnosis. It showcases enhanced efficiency and speed for case reviews, consultations, and second opinions, facilitating telepathology, remote work, and intraoperative consultations. The interactivity of WSI enriches the functionality of multidisciplinary tumor boards. Moreover, DP has been an invaluable asset in educational settings for years, unveiling new quality assurance and control avenues.

1.1.4 Computational Pathology

CPATH represents a transformative intersection of pathology, computer science, and AI that aims to enhance the analysis of medical histopathology images, primarily for diagnosing and treating diseases, especially cancer [37, 18]. This interdisciplinary field focuses on developing computational approaches to analyze these images, with the overarching goal of

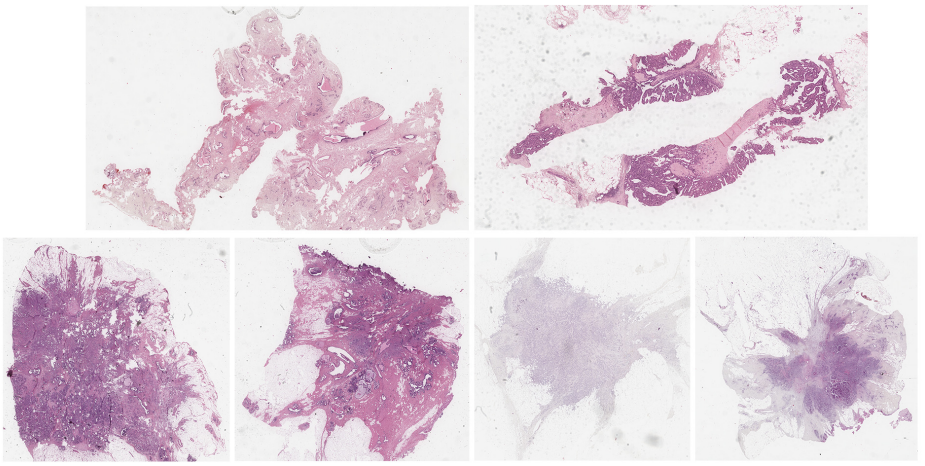


Figure 1.2. WSIs examples

creating a digital diagnostic infrastructure to assist clinical pathology. Two primary approaches have been identified in the realm of CPATH as reported in [37]. The Data-Centric Approach emphasizes collecting and compiling pathology data, addressing the lack of labeled WSIs data, and capturing a predefined pathology ontology. In contrast, the Model-Centric Approach, favored by computer scientists and engineers, designs algorithmic solutions based on available pathology data. AI, particularly deep learning techniques, plays a pivotal role in CPATH. AI-based computational pathology has shown immense promise in increasing the accuracy and availability of high-quality healthcare.Artificial Neural Networks (ANNs), especially Convolutional Neural Networks (CNNs), have been basic tools in analyzing digital pathology images. Early machine learning methods aimed to replicate the pathologist's approach. However, as Machine Learning (ML) and Deep Learning (DL) evolved, they began performing tasks beyond a pathologist's capabilities. Histopathologic image analysis can be categorized into several areas of segmentation and classification. Segmentation performs as accurate recognition of the borders of specific tissue elements, such as epithelia, glands, stroma, cells, or nuclei. Classification grouping images into categories, such as tumor vs. non-tumor, tumor subtypes, or tumor grades. There is more information

in histopathology images than what is visible to the human eye. Machine learning and deep learning can extract these "subvisual" features for classification or predicting molecular findings from standard H&E images. Some histomorphologic features correlate with molecular features, allowing for potential predictions of molecular signatures from WSIs. Advanced deep learning-based approaches can infer molecular features and therapy responses directly from tissue biomarkers [48, 49, 9].

1.2 Motivation

Cancer is one of the leading causes of death globally. Its complexity lies in the diversity of cancer types and their subtypes, which can vary significantly among individuals. Proper classification of cancer subtypes is critical for accurate diagnosis and selection of the most effective treatment. However, classification based on traditional methods can be error-prone and time-consuming. Graph Neural Networks (GNNs) have shown significant potential in analyzing data structured as graphs. This makes the approaches particularly suitable for analyzing biomedical data, which often have complex and interconnected relationships. In parallel, digital pathology, which involves the analysis of digitized images of tissue specimens WSIs, is increasingly gaining ground as an essential tool in cancer diagnosis and research. However, the massiveness and complexity of WSIs pose significant information extraction and interpretation challenges. This is where the concept of deep features comes into play. Using deep learning techniques to extract relevant features from WSIs can significantly improve the effectiveness of pathological search and diagnosis. Combining GNNs for cancer subtype prediction and deep features for WSIs retrieval could lead to a breakthrough in cancer diagnosis and treatment.

1.2.1 Research Questions

- **RQ1**: Are pre-trained CNNs suitable as a feature extractor in Tissue Region Of Interests (TRoIs) Content Based Image Retrieval (CBIR) systems?
- **RQ2**: What are the effects of Color Stain Normalization (CSN) in a TRoIs CBIR systems?

- **RQ3**: How can we obtain a CNN for Nuclei Instance Segmentation and Classification faster than State-of-the-art (SOTA) ones?

- **RQ4**: How can we obtain a unified framework for Cell-Graph (CG) building?

- **RQ5**: Is our CG representation suitable for breast cancer subtype classification?

1.3 Our Contribution

We focused on histological patch image retrieval and classification of TRoI based on cell nuclei classification. Concerning retrieval, we designed a CBIR framework for Picture Archiving and Communication System (PACS) in pathology, where the main characteristic is the employment of pre-trained CNNs as feature extractors. In particular, we used this framework to conduct two studies. First, we analyzed SOTA CNNs to identify the main blocks in it, and then we experimentally evaluated each output taken from each CNN block to understand what is the best way to extract features in this domain. Later, we experimentally estimated the effects of color stain normalization in our framework to see if they effectively improved the results. Meanwhile, we proposed different contributions concerning TRoI classification. Since this task requires CG building, we first focused on nuclei instance segmentation and classification, where we identified the SOTA a CNN, HoVerNet [27], which has good performance in terms of classification and segmentation. Still, it is too slow for real applications. Therefore, we proposed Fast-HoVerNet, a faster version obtained using Knowledge Distillation (KD), which reached similar performance in segmentation and classification but is three times faster. Afterward, we worked on nuclei feature extraction, proposing novel techniques to extract nuclei features. In this task, common approaches use pre-trained CNNs, which produces high dimensional features and often can be redundant for vertical tasks, so we proposed to use Fast-HoverNet to extract nuclei features. We proved the effectiveness of our graph representation approach in cancer subtype classification, where we achieved results comparable with SOTA.

1.4 Thesis Organization

Chapter 2 provides the background, initially focusing on the content-based retrieval approach in digital pathology, then exploring nuclei instance segmentation and classification, and subsequently delving into the application of GNNs in computational pathology. Chapter 3 presents our proposed framework, emphasizing its architecture and highlighting the used technologies and methodologies. Chapter 4 shows our experimental results, together with a comparative analysis with the SOTA. Chapter 5 summarizes our research and points out our findings, discussing our work's strengths and weaknesses.

Chapter 2

Background and Related Work

This chapter introduces the background knowledge reporting the most relevant study about CPATH related to image retrieval and image analysis using CNNs and GNNs. Furthermore, we reported the most relevant deep-learning techniques involved in this work. In particular, we explored in-depth feature extraction techniques, instance and semantic segmentation methods used for nuclei extraction, and graph neural networks in CPATH applications.

2.1 Image Retrieval in Pathology

In this section, we analyzed different works on histopathology image retrieval, mainly focusing on feature representation. In [7], the authors presented an approach to represent histopathology knowledge for CBIR systems. It was accomplished by a semantic mapper based on support vector machine classifiers. This mapper allows for a new semantic feature space in which a metric measures image similarity. They used ray histograms, color histograms, Tamura texture histograms, and Sobel histograms and computed other meta-features on the histograms. The authors in [8] presented a framework to build histology image representations that combine visual and semantic features using the NMFA and NSE algorithms. Their method learns the relationships between both data modalities and uses that model to project semantic information back to the visual space, building the fused representation. Ultimately, such representation is used

in an image search system that matches potential results using a similarity measure. In the same way, in [91], the authors introduced an image retrieval framework for histopathological image analysis. Mainly, they focused on hashing-based retrieval methods and investigated a kernelized and supervised hashing approach for real-time image retrieval. Instead, in [62], the authors proposed a CBIR algorithm based on a hierarchical annular histogram with a refinement schema based on dual-similarity relevance feedback. Jimenez et al. [41] proposed a multimodal case-based retrieval approach for histopathology cases based on visual features obtained with deep learning with an automatic description of pathology reports. Furthermore, they used a strategy fusing visual features from WSIs and text embeddings of pathology reports. The deep features for WSIs are generated with a CNN trained to classify cancer gradings. Moreover, authors in [92] proposed a complete size-scalable CBIR framework for a large-scale database of WSIs using the binarization method and hashing technique to feature identification and similarity measurement for the images represented in multiple binary codes. The primary operand of the proposed method is from the ranking step, which varies with the number of proposal regions. Also, in [46], authors produced compact features for image retrieval, reduced deep features, and deep barcodes derived from deep features of a pre-trained network. They used VGG16, VGG19, and AlexNet as deep feature extractors. In [33], a deep learning-based reverse image search tool for histopathology images is presented called Similar Medical Images Like Yours (SMILY). It allows pathologists to perform queries by example. They divided their application into two stages. The first one concerned the database creation from WSIs, tailing the images into patches, and extracting the depth features. The latter concerns the query process, where a patch is selected from query WSI, and the nearest neighbor search is performed. Likewise, in [69], authors proposed a DP system with a WSIs viewer to retrieve visually similar local areas in the same image and other images from an extensive database and open-access literature. The system evolves a DenseNet121 trained on breast cancer histopathological images and deep features extraction from patches. In [89], a retrieval and classification system for histological images based on local energetic information, local structural information, local geometric information, and local patterns of the textures using Riesz Transform and Monogenic local

binary patterns (M-LBP). In [88], the authors use deep metric learning for the histopathological image retrieval task. They constructed the network with a mixed attention mechanism involving spatial and channel attention and trained with the multi-similarity loss under the supervision of category information. Yottixel [43] is an image search engine for DP. It is based on a combination of supervised and unsupervised algorithms. In particular, it uses the VGG network, all Inception, DenseNet, and in-house trained CNNs. They used deep features to characterize patches extracted from WSIs.

2.2 Nuclei Instance Segmentation and Classification

This section reports the main work on nuclei instance segmentation and classification. This task is one of the most crucial in pathology because nuclei's characteristics are useful for analyzing tissue images. This task consists of two subtasks: nuclei instance segmentation and nuclei classification. The first subtask detects and segments nuclei instances, or rather, it classifies each pixel of them as a nucleus. The second subtask classifies each nucleus using nuclei type as reported in the training dataset. Over the years, many solutions have been proposed to solve this task. Some scholars proposed to adapt some architecture for segmentation designed for general tasks, while others proposed ad-hoc architectures. This section focuses only on the architecture specifically designed for this task. We devived it in tools that allow us to perform nuclei instance segmentation, classification, and approach based on Deep Neural Network (DNN).

2.2.1 Pathology tools

CellProfiler [10] is a sophisticated open-source image analysis software developed to identify and quantify cell phenotypes, addressing the challenges in analyzing large volumes of high-resolution cellular images. Its modular architecture characterizes the software, allowing for tailored applications and flexibility in diverse research projects. It is equipped with advanced algorithms capable of accurately identifying a variety of cell types and features, and it adeptly addresses illumination correction, a critical as-

pect of image analysis. It has an intuitive user interface, is user-friendly, and is designed for high-throughput, efficient processing of large-scale experiments, making it a valuable asset in biological research. As an accessible open-source platform available across various operating systems, it fosters collaborative research and is supported by a robust community and extensive documentation. QuPath [5] is an open-source bioimage analysis software designed specifically for DP and whole slide image analysis. It aims to provide a user-friendly, extensible solution to meet the growing needs in this field. Furthermore, it offers a comprehensive set of tools for tumor identification and high-throughput biomarker evaluation, along with powerful batch-processing and scripting functionality. It features a cross-platform, multithreaded, tile-based whole slide image viewer with extensive annotation and visualization tools. The software employs a hierarchical, object-based data model, enabling efficient representation and interaction with large numbers of image objects across gigapixel images.

2.2.2 Deep Neural Network based approach

Here, we describe all architecture specifically designed for CPATH. The architecture of U-Net [67] is designed for biomedical image segmentation and is characterized by a U-shaped structure comprising a contracting path and an expansive path. The contracting path adheres to the conventional architecture of a convolutional network, involving repeated applications of two 3×3 convolutions, each succeeded by a Rectified Linear Unit (ReLU) and a 2×2 max pooling operation with stride 2 for downsampling. With each downsampling step, the number of feature channels doubles, enabling the contracting path to capture the context in the input image. Symmetrically juxtaposed to the contracting path is the expansive path. Each step in this path encompasses an upsampling of the feature map, followed by a 2×2 convolution, termed "up-convolution," which halves the number of feature channels. Subsequently, there is a concatenation with the correspondingly cropped feature map from the contracting path and two 3×3 convolutions, each followed by a ReLU. This expansive path is instrumental in augmenting the resolution of the output and facilitates precise localization by amalgamating high-resolution features from the contracting path with the upsampled output. In the terminal layer of the architecture, a 1×1 convolution is employed to map each 64-component feature vector to

the desired number of classes, culminating in a network comprising 23 convolutional layers. A distinctive feature of this architecture is its seamless tiling of the output segmentation map, achieved by judiciously selecting the input tile size to ensure that all 2×2 max-pooling operations are applied to a layer with an even x- and y-size. The implementation of stochastic gradient descent is pertinent to the training of the U-Net. The architecture necessitates the incorporation of data augmentation, with a particular emphasis on elastic deformations, which is indispensable for training the network with a limited number of annotated images. This architectural design exhibits a harmonious balance between localization and context, rendering it particularly apt for tasks in biomedical image segmentation. In [11], the authors proposed a Deep Contour-Aware Network (DCAN) for accurate gland segmentation in histology images. This network is designed to address the challenges of gland segmentation by harnessing multi-level contextual features in an end-to-end manner. DCAN operates under a multi-task learning framework, allowing it to segment gland structures simultaneously and depict clear contours to separate clustered objects. The network architecture comprises both downsampling and upsampling paths, with the former extracting high-level abstractions and the latter predicting pixel-wise score masks. By integrating complementary information from gland objects and their contours, the DCAN efficiently segments and differentiates individual gland structures, even in cases where they are closely clustered. Additionally, the authors employ transfer learning to enhance its performance, initializing the network with pre-trained weights from a model trained on a larger dataset. The authors in [55] formulate the segmentation problem as a regression task of the distance map, aiming to address the challenge of segmenting touching or overlapping nuclei, a recurrent issue in the field of biology. The architecture employed in this study is a DNN that maps its final output to $R^{n \times p \times k}$. This mapping is achieved by modifying the final $1x1$ convolutional layer, enabling it to map each pixel convolutional layer k channels. The global loss is defined, incorporating model parameters and a weight decay hyperparameter, λ, and is optimized using Adam optimization. A crucial aspect of the architecture is the introduction of a scale parameter, n_{feat}, which serves to scale the complexity of the DNN. This parameter is specifically used for the U-Net, as other architectures in the study utilize pre-trained networks

with fixed architectures. The authors formulated the segmentation problem as a regression task of the distance map, which aims to enhance the segmentation of nuclei that are in close proximity or overlapping, which is a recurrent and challenging issue in biological image analysis. By regressing the distance map, the architecture can more effectively discern the boundaries between adjacent nuclei, thereby improving the segmentation accuracy. HoVerNet [27], a specialized CNN, was designed to optimize the segmentation and classification of nuclei in histology images. The architecture integrates a pre-activated residual network with 50 layers for feature extraction, employing different residual units at mixed down-sampling levels to modulate spatial resolution within the network. The network achieves nuclei instance segmentation and classification through three distinct up-sampling branches: the Nuclear Pixel (NP) branch for distinguishing nuclei pixels from the background, the Horizontal and Vertical (HV) branch for predicting horizontal and vertical distances of nuclear pixels to their centers of mass, and the Nuclear Classication (NC) branch for determining the type of each nucleus. This multi-branch approach is pivotal for separating touching or clustered nuclei and ensuring accurate classification. In particular, horizontal and vertical distance maps are central to the segmentation process, representing the spatial relationship of each pixel to the nucleus's center of mass, thereby enhancing accuracy, especially in high-density areas. These maps are instrumental in the post-processing stage for refining segmentation results and associating pixels with specific nuclei instances while also addressing variability in nuclei appearance by providing consistent spatial representation across different nuclei types. The post-processing algorithm calculates the gradient using the Sobel operator, utilizing the horizontal and vertical distance maps from the HV branch. This highlights areas of significant pixel value differences, indicative of boundaries between neighboring nuclei, guiding the meticulous separation of nuclei. The algorithm also incorporates the watershed algorithm, using markers derived from threshold-applied energy landscapes, such as the distance map, in a marker-controlled watershed to segment identified regions. This methodology is essential for the precise separation and segmentation of individual nuclei instances, underscoring the accuracy of the post-processing stage.

2.3 Graph Neural Networks in Pathology

In this section, we introduce several works related to GNNs in pathology for WSIs or TRoIs classification. In particular, classification concerning cancer type or subtype. As reported by Ahmedt-Aristizabal et al. in [3], different approaches to represent a WSIs or TRoIs as a graph are presented. In particular, they recognized five main representations: CG, Cluster-Centroid-Graph (CCG), Tissue-Graph (TG), Patch-Graph (PG), and hierarchy. In the following section, we described these graph representation techniques and the main works that use them in the classification task.

2.3.1 Cell-Graph

A CG is a computational representation that captures the morphology and topology of cells within a tissue or sample. In constructing a CG, individual cells are represented as nodes, and the interactions or relationships between these cells are represented as edges. The edges can be determined using various criteria, such as proximity or specific cellular interactions. For each cell, features like shape, texture, and spatial location can be extracted and associated with the corresponding node. Algorithms, such as the k-Nearest Neighbors (KNN), can be used to recognize the graph topology, connecting cells that are close to each other and ensuring that distant cells remain disconnected. While CG effectively captures cellular interactions and structures, they might not always consider the broader tissue distribution information. The following describes the most recent GNNs that works on CG. The CGC-Net[93] transforms each large histology image into a graph. In this graph, each node represents a nucleus from the original image, and edges between nodes denote cellular interactions based on node similarity. The network uses nuclear appearance features and the spatial location of nodes to enhance its performance. The authors also introduce Adaptive GraphSage, a graph convolution technique that merges multi-level features in a data-driven manner. To manage redundancy in the graph, they propose a sampling technique that eliminates nodes in areas with dense nuclear activity. The images used in the study are divided into three classes based on the degree of gland differentiation: normal, low grade, and high grade. The dataset used for evaluation consists of images

with dimensions 4548×7520 at $20\times$ magnification. The results indicate that the proposed method achieves an image accuracy of 96.28 ± 1.03, highlighting its effectiveness in colorectal cancer grading by incorporating nuclear and graph-level features. Authors in [74] explored using Graph Convolutional Networks (GCNs) to visualize histopathology images. The study emphasizes the importance of nuclei and their relationships in cancer tissue by introducing an attention-based architecture and occlusion-based visualization. Two datasets were used: the BreAst Cancer Histology (BACH) dataset and the Prostate Cancer Gleason Grade Dataset. The primary objective was to differentiate specific cancer types, achieving an accuracy of 0.94 for BACH binary classification and 0.97 for the Gleason dataset using three distinct models. The results are in line with current state-of-the-art benchmarks. Guillaume Jaume et al. [40] presented the CGE XPLAINER, a model designed to provide compact and interpretable explanations from DP images. This model processes the images by using GNN, specifically the Graph Isomorphism Network (GIN) [86]. By mapping DP images to CGs, where cells and their interactions are represented as nodes and edges, the model shifts from traditional pixel-level analysis to a more biologically relevant entity/relationship-oriented representation. Using the BReAst Carcinoma Subtyping (BRACS) dataset comprising 2080 TRoI from breast carcinoma images, the CGE XPLAINER demonstrated its ability to simplify these graphs effectively while retaining crucial diagnostic details. The results underscore the model's potential to provide consistent and meaningful explanations, marking a significant advancement in the interpretability realm of DP. In [73], the authors delve into the application of Message Passing Neural Networks (MPNNs), a subset of GCNs. These networks aggregate information from neighboring nodes in a graph through iterative message passing. The GCN, a spectral-based GNN, is particularly highlighted. Among the various GNN architectures explored, GCN with Jumping Knowledge (JK) [87] and Graph-SAGE [29] were the top performers, achieving a classification accuracy of 94.8%. This work underscores the potential of GNN in classifying intestinal glands, offering a promising avenue for further research in the domain.

2.3.2 Cluster-Centroid-Graph

A CCG is a computational representation used in the context of image analysis, particularly in histopathology. In this representation, features from images, often extracted using CNN, are clustered using techniques like K-means clustering. Each cluster's centroid, representing the central or most representative point of a cluster, is treated as a node in the graph. Edges in the graph are then established based on correlations or intrinsic similarities between these centroids. The resulting graph captures relationships between different feature clusters, providing a higher-level understanding of the underlying data. This graph can then be used as input to GCNs or other models for further analysis or classification tasks. Authors in [70] investigated a method for cervical cell classification using a GCN. The proposed approach utilizes GCN to explore image-level potential relationships to enhance classification performance. A CNN, specifically DenseNet-121, is employed to represent each cervical cell image. These images' CNN features are clustered, and a graph structure is constructed based on the intrinsic clustering correlation. GCN is then applied to propagate the underlying correlation of nodes in this graph. Regarding results, the proposed method achieved a classification accuracy of $95.35 \pm 0.42\%$ on the SIPaKMeD Dataset. On the Motic Dataset, the technique had accuracies of $94.90 \pm 0.25\%$ and $94.86 \pm 0.34\%$ on the two subsets, respectively.

2.3.3 Tissue-Graph

A TG is a computational representation that captures tissue section properties and spatial distribution within a biological sample. In this representation, distinct tissue regions or structures are denoted as nodes, while the relationships or interactions between these regions are depicted as edges. The edges can be established based on various criteria, such as the proximity of tissue regions or specific functional interactions between them. The TG operates higher than representations like the CGs, focusing on the broader tissue structures and their interrelationships rather than individual cells. This approach provides insights into tissues' overall organization and context, which can be crucial for understanding pathological changes or disease states. The following describes the most recent architecture that works on TG. Wenqi Lu et al. [52] introduced the Slide Graph, a graph

convolutional neural network model. The model's workflow comprises four main steps. First, it uses the HoVerNet to perform simultaneous nuclei instance segmentation and classification. Next, it employs agglomerative clustering to group spatially neighboring nuclei into clusters. The third step involves constructing a global planar graph representation of the WSI, where each cluster is treated as a node. Finally, the model uses a graph GCN with graph GIN layers to predict WSI-level labels. For evaluation, they utilized a dataset from The Cancer Genome Atlas (TCGA) BReast CAncer (BRCA) containing 709 WSIs. Specifically, for Human Epidermal growth factor Receptor 2 (HER2) status differentiation, the dataset has 608 HER2 negative and 101 HER2 positive images. The results indicate that the Slide Graph model could predict HER2 and PR status. Additionally, it is more computationally efficient than traditional patch-based models. The document also provides insights into the model's ability to visualize node features, showing clear differences between HER2+ and HER2- WSIs. Authors in [63] focused on staging Colorectal Cancer (CRC) using WSIs. The authors introduce a Graph Attention Multi-instance Learning (Graph Attention MIL) model incorporating texture features to capture the spatial structure between patches and predict the TNM staging of CRC. The model has two main components. The first is texture feature extraction, which aims to learn a feature representation for a given image patch, focusing on texture-based features. It uses a texture encoding network and a cluster embedding network. The second component is Graph Attention Multi-Instance Learning. After clustering the patches into different tissue types, the texture features from the patches are used to predict the tumor stage. The model uses a GCN to learn the relationship between features from each WSI. Multiple graphs are created, with each graph representing different tissue features. An attention mechanism is applied to these instances to predict the tumor stage. They used a dataset from the Molecular and Cellular Oncology (MCO) study, which contains data from over 1,500 Australian individuals who underwent curative resection for CRC from 1994 to 2010. The dataset was split into training, validation, and testing sets. Each WSI has been annotated with an image-level label representing the tumor stage by expert pathologists. Over a million patches were extracted from the dataset, covering approximately 82% of the tissue area. The baseline method, which trained on reduced-resolution images,

achieved an accuracy of 53.6%. This indicates that down-sampled WSIs are ineffective for staging due to the loss of resolution. The paper suggests that their proposed framework improves performance over existing methods, indicating potential for future research in graph-based learning for TNM staging.

2.3.4 Patch-Graph

A PG is a computational representation derived from DP, particularly from WSIs of tissues. In this representation, the tissue image is divided into fixed-sized patches, and each patch is represented as a node in the graph. The edges between nodes can be established based on various criteria, such as spatial proximity or feature similarity. Instead of directly analyzing the entire high-resolution image, the PG allows for a more structured and efficient analysis by focusing on these smaller, meaningful image segments. This representation is particularly useful in capturing local tissue structures and their relationships, facilitating tasks like tissue classification or disease detection. The following reports the most recent works that use PG. Authors in [88] introduced a system designed to segment and classify breast cancer TRoI images. This system integrates hierarchical processing for both segmentation and classification tasks. It consists of two main modules: a segmentation module and a GCN module. The segmentation module obtains segmentation masks of image patches from the TRoI image, which are combined to produce the segmentation result of the entire TRoI image. On the other hand, the GCN module captures spatial and semantic relationships among image patches by constructing a graph using the segmentation masks. The features learned by the GCN module are then used to classify the TRoI image. The system architecture employs an encoding and decoding structure for semantic segmentation, where encoding extracts deeper semantic information and decoding reconstructs image features. Mohammed Adnan et al. in [2] delved into the representation learning of WSIs in DP. Their method involves sampling relevant patches using a color-based technique and then employing graph neural networks to understand the relationships among these patches, aggregating the image information into a singular vector representation. They proposed a framework that models a WSI as a fully connected graph, with each instance treated as a node of the graph, enabling the learning of relations

among nodes end-to-end manner. The research emphasizes learning the adjacency matrix, which defines the connections within nodes. Additionally, the paper introduces attention via graph pooling to discern patches of higher relevance automatically. The model is designed to process the entire WSI at its highest magnification level, requiring only a single label for it without any patch-level annotations. For the dataset, the researchers utilized 1,026 lung cancer WSIs from TCGA dataset, focusing on classifying two prevalent subtypes of lung cancer: Lung Adenocarcinoma and Lung Squamous Cell Carcinoma. The results revealed that the proposed method achieved an accuracy of 88.8% and an AUC of 0.89 on lung cancer subtype classification by leveraging features from a pre-trained DenseNet model, showcasing its potential in differentiating between the two lung cancer subtypes.

2.3.5 Hierarchical Representation

A hierarchical representation in DP is a multi-level computational approach that captures the detailed cellular structures and broader tissue distribution properties. It integrates both low-level and high-level information from a tissue sample. At the lower level, it might use a CGs to represent individual cells and their interactions. At a higher level, it might employ a TG to capture larger tissue sections' properties and spatial distribution. The hierarchical nature of this representation ensures that both the minute details of cellular interactions and the overarching tissue structures are considered. This comprehensive view is essential for understanding the intricate relationships within biological tissues and can be crucial for tasks like disease diagnosis or treatment response prediction. Pati et al.[60] proposed using hierarchical graph representations to enhance tissue specimens' diagnosis, prognosis, and therapy response predictions. The authors introduce a novel approach that employs a multi-level hierarchical entity-graph representation of tissue specimens. This representation, termed HierArchical Cell-to-Tissue (HACT), captures information at multiple levels, ranging from individual cells to broader tissue regions. The HACT graph is designed to represent both the cell microenvironment and the broader tissue microenvironment. To process this graph, they propose a hierarchical graph neural network called HierArchical Cell-to-Tissue Network (HACT-Net). This network is designed to classify the HACT

representations, effectively mapping the tissue structure to its functionality. The methodology mirrors how pathologists analyze tissues, moving from detailed to broader perspectives. To evaluate their approach, the authors introduce the BRACS dataset, which consists of a large collection of H&E stained breast tumor regions of interest. Using this dataset, they benchmark their methodology against pathologists' assessments and other computer-aided diagnostic methods. The results reveal that their proposed method offers superior classification outcomes compared to other methods. Notably, their results are on par with individual pathologists, showcasing the potential of their method in the realm of cancer diagnosis and prognosis. The paper underscores the significance of capturing cellular and tissue-level insights to understand tissue structures and their implications in cancer care comprehensively. Authors in [90] proposed a graph convolutional neural network, MS-GWNN, designed for histopathological image classification of breast cancer. This model captures multi-scale contextual features in cancerous tissue by leveraging the localization property of spectral graph wavelet for multi-scale analysis. The MS-GWNN is trained end-to-end and maps pathological images into the graph domain to capture multi-level tissue structural information. The architecture comprises graph construction, node classification based on GWNN, and graph classification using feature aggregation. The model was tested on two public datasets: the ICIAR 2018 BACH grand challenge dataset and the BreakHis dataset. The BACH dataset consists of 400 histopathological images, with 320 used for training and 80 for testing. The BreakHis dataset has 7909 samples, with the study focusing on samples at 40× magnification. Regarding results, the MS-GWNN achieved 93.75% accuracy on the BACH dataset and 99.67% on the BreakHis dataset. The model's high accuracy, especially on the BreakHis dataset, underscores its potential in breast cancer diagnosis based on histopathological images.

2.4 Proposed Improvements

Here, we describe the difference between our and existing methods.

CBIR in Patholgy

Concerning the CBIR in pathology, we propose a comparative study over multiple CNNs to identify the best approach to extract patch features. Furthermore, we conducted a study to estimate the effects of CSN.

Nuclei Instance Segmentation and Classification

After analyzing the literature, we identified the need to have a faster nuclei instance segmentation and classification network than existing ones. Therefore, we propose Fast-HoVerNet that achieved SOTA results, and it is three times faster. Furthermore, we also used KD in this task to reach our purpose.

Cell-Graph Representation and Classification

For CG classification, we propose a novel approach to extract cell features. Our method obtains cell feature shapes smaller than existing methods in CG representation and also unifies the process using the same network to extract nuclei instances and their feature. Furthermore, we prove the effectiveness of our approach in a classification task where our results are comparable with SOTA.

Chapter 3

Pathological Images Analysis

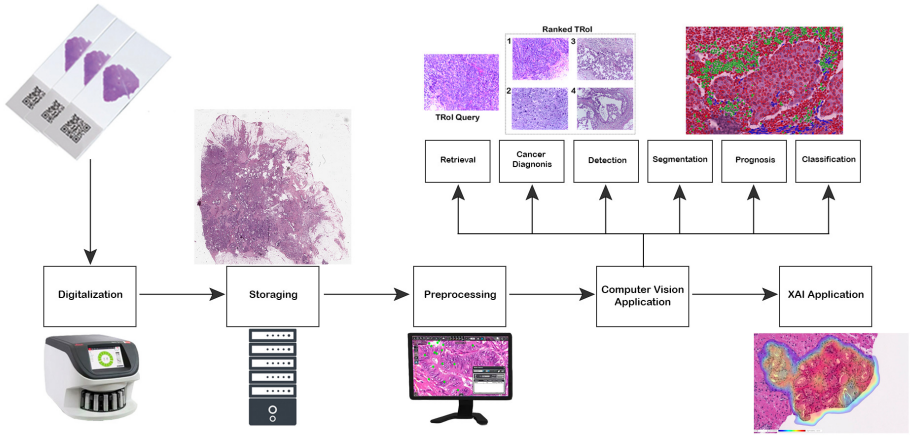


Figure 3.1. Computational Pathology framework

This chapter introduces our CPATH framework, highlighting our contributions. The high-level framework, illustrated in Figure 3.1, reports the main module mandatory in our framework. The first is digitalization, which transforms raw glass slides into WSIs, and then they are stored. Afterward, the WSIs are preprocessed using techniques like CSN. At this stage, the data are ready for computer vision application. These applications mainly concern retrieval, cancer diagnosis, detection, segmentation,

prognosis tasks, and classification. Lastly, some eXplainable Artificial Intelligence (XAI) applications can be used to make the results of AI applications useful for pathologists.

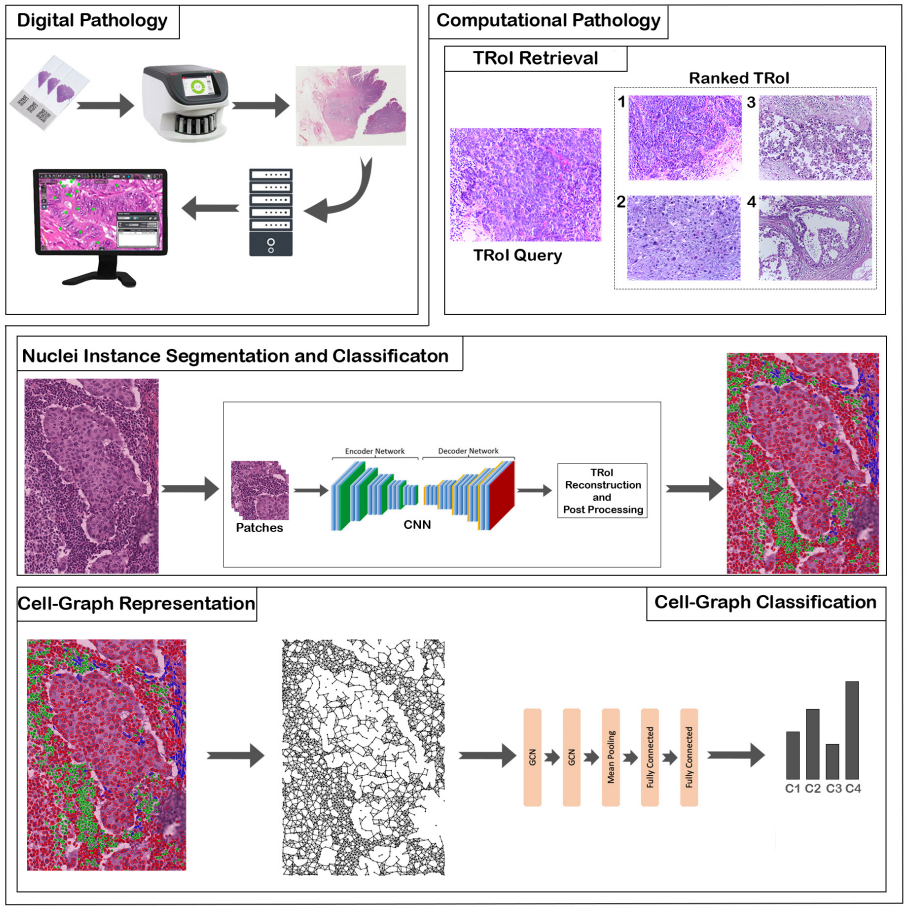


Figure 3.2. Visual summary of modules implemented by ours. In particular, the tasks are grouped in the Computational Pathology block.

In particular, we explored the following topics:

- H&E TRoIs retrieval: we propose a CBIR for TRoIs using deep feature obtained from pre-trained CNNs;

- Instance nuclei segmentation and classification: we propose Fast-HoVerNet a CNN faster than HoVerNet, which is one of the most used networks for this task;
- H&E WSIs classification using CG representation:
  - CG Representation: we propose a novel method to extract nuclei features in CG representation;
  - We will prove that our representation model obtained results comparable with SOTA approaches but having a feature length smaller than common approaches;

Figure 3.2 summarizes all tasks that we explored and implemented in this manuscript.

3.1 H&E TRoI Retrieval

PACSs have become the de facto standard in digital pathology. Usually, the component for retrieval is based on metadata and generally uses textual or structured information. To improve PACS's usability and precision, we propose using the CBIR module, which allows retrieval of TRoI as a portion of the searched WSI. In this way, pathologists can search for similar cases in PACS to make decisions or analyze them. Our CBIR system is depicted in Figure 3.3; it mainly consists of two blocks. The first one extracts and stores features from TRoIs in the features repository. The second block computes the similarity between query features and stored features. In particular, our system contains three main modules: *Image Preprocessor*, *Features Extractor*, *Similarity Performer*, and *Results Ranker*.

The *Similarity Performer* computes the similarity between a visual query and all stored TRoIs features. The similarity measure used the cosine similarity, defined in Equation 3.1. Where A and B are two vectors with the same dimension, n, they contain the image features.

$$sim(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.1}$$

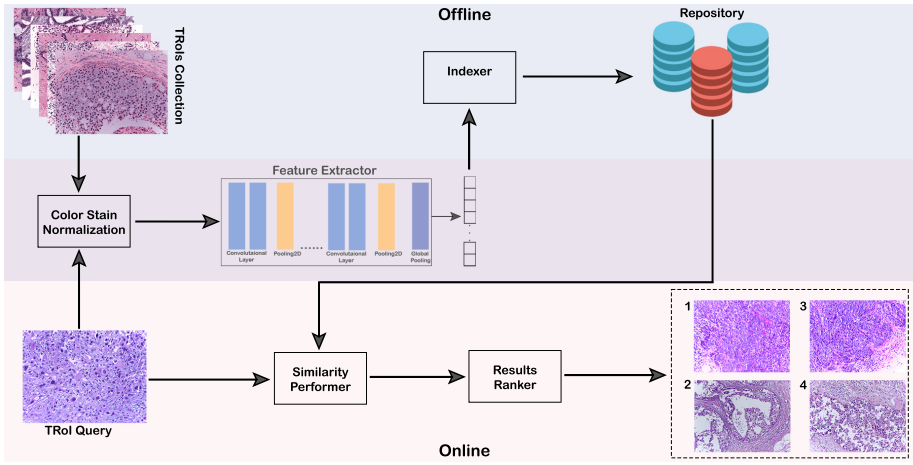


Figure 3.3. Our CBIR Framework

The *Features Extractor* implements methods to extract features. It is used both in the online process and offline processes. In particular, we used CNNs pre-trained on ImageNet to extract features. In the following, we detailed each CNN supported by our system.

Image Preprocessor implements image processing techniques like reshape, and it is mainly focused on CSN that mitigates color and intensity variations arising from differing stain protocols and concentrations. We implemented various techniques that we describe in the following subsections.

3.1.1 Color Stain Normalization

Usually, histological images undergo staining with dyes such as H&E to emphasize distinct tissue elements. But here is the hitch: staining consistency can fluctuate across different labs and even within batches from the same lab. Such inconsistencies render these images with varied color tones and intensities, complicating subsequent analyses and interpretations. To tackle this, CSN techniques endeavor to harmonize the appearance of these images. They adjust the color and intensity profiles to align with a specified reference image, ensuring that the structural intricacies remain untouched while the colors match. The methods adopted for CSN

span a spectrum, from straightforward linear color channel adjustments to intricate ones rooted in statistical or machine learning frameworks. These techniques enhance ensuing tasks like nuclei segmentation and classification by stabilizing image input. However, CSN isn't without its hurdles. Choosing the right reference images, maintaining inherent biological diversity, and managing multi-stained images are among the challenges. Yet, given its significance in delivering precise and repeatable histological image analysis, CSN remains a cornerstone in the discipline. Despite some CSN methods using deep learning in recent years, we focused on the deterministic ones. One of the most used CSN methods was proposed by Macenko et al. [53], which involves the decomposition of the color image into its constituent stains using Single Value Decomposition (SVD). In particular, the Macenko method first converts the RGB image of the tissue into an Optical Density (OD) space. This transformation is applied using the Equation 3.2 where I is the intensity, and I_0 is the maximum possible intensity, usually 255 for 8-bit imaging. Afterward, non-informative pixels are excluded. Typically, this consists of filtering out OD values belonging between thresholds.

$$OD = -\log \frac{I}{I_0} \tag{3.2}$$

The main idea of the Macenko method is to represent the image OD values as a linear combination of the OD values of the individual stains. By assuming that most histological images are primarily stained using two stains (like Hematoxylin and Eosin), the problem boils down to finding two vectors (representing the stains) that best account for the variation in the OD values across the image. SVD is used for this purpose. Once the matrix of OD values is formed, its columns are ordered based on their importance using SVD. The first two columns then correspond to the stain vectors. Then, Using the stain vectors obtained from SVD, the image can be separated into its constituent stains. Each pixel's OD value is expressed as a linear combination of the two stain vectors, giving the concentration of the stains at that pixel. To perform stain normalization, stain concentration values of the source image are scaled such that their mean and standard deviation match those of a target or reference image. Finally, the normalized stain concentrations are combined using the stain vectors to produce the normalized OD image. This is then converted back to the

RGB space to get the final color-normalized image. Again, Reinhard et al. introduced a CSN technique that aligns the statistical attributes of images in the Lab color space [65]. Initially, the source and reference images are transformed from their native RGB representation into the Lab domain. The Lab color space is favored due to its perceptual uniformity, which ensures visually consistent differences across its range. In this color space, 'L' denotes lightness, 'a' captures the spectrum from green to red, and 'b' covers the range from blue to yellow. The mean and standard deviation for the source and target images in each L, a, and b channels are computed for normalization. The source channel is then adjusted by subtracting its mean, followed by a rescaling based on the ratio of the standard deviations of the target to the source. Subsequently, the mean of the target is added to the adjusted source channel. The source image's color distribution closely matches the target image across the Lab channels through this process. After these adjustments, the image is converted back from Lab to RGB. The strength of the Reinhard approach is its simplicity. Rather than relying on intricate procedures like matrix factorization or deep learning, it aligns the source and target images based solely on their statistical attributes, yielding visually harmonized results. Vahadane et al. introduced a novel approach to stain normalization that hinges on sparse Non-negative Matrix Factorization (NMF) for efficient stain separation and subsequent color normalization [81]. Their procedure commences with a transition from the RGB domain to OD space, mirroring the foundational steps of the Macenko method. This transformation is pivotal for counteracting the inconsistencies introduced by fluctuating illumination during imaging. In this method, the extracted OD values undergo factorization to yield two distinct matrices, indicative of stain concentration and color, respectively. A crucial attribute of NMF is its guarantee of non-negative matrix values, an essential characteristic considering the physical impossibility of negative stain concentrations and colors. The sparsity inherent in the factorization is particularly advantageous as it mirrors the histopathological reality where typically a limited set of stains, often just Hematoxylin and Eosin, dominate the slide. Upon obtaining these matrices, the methodology entails deconstructing the image into its foundational stains. This allows each pixel's OD value to be interpreted as a nuanced blend of various stain colors, modulated by their corresponding concentrations. The

normalization phase involves adjusting the source image's concentration values to align with those of a designated target or reference image. Color values can be fine-tuned to emulate a predetermined target or align with a specific staining device's color signature if necessary. Lastly, the process involves merging the recalibrated stain concentration and color matrices and reverting from the OD domain to the familiar RGB domain. This results in the generation of a harmonized, stain-normalized image.

3.1.2 Convolutation Neural Netowrks

This section describes the deep neural network architectures we implemented in the *Feature Extractor*.

VGG-Net

VGGNet [71] was designed by the Visual Geometry Group at the University of Oxford. This convolutional neural network's core attribute is its reliance on uniformly small 3×3 convolutional layers stacked in increasing depth, a departure from using larger filter sizes. Filters detect patterns, textures, and other features in the input image. Between convolutional layers, max-pooling layers reduce the spatial dimensions, effectively summarizing the features identified in the preceding convolutional layers. VGG16 and VGG19 are the most notable variants. VGG16 comprises 13 convolutional layers and three dense, fully connected layers, making it 16 layers deep. VGG19, as the name suggests, has an additional three convolutional layers, totaling 19 layers. The ReLU activation function is applied as a convolutional operation, introducing non-linearity and enhancing the network's capability to learn complex patterns. Once the data passes through these convolutional and pooling layers, it reaches the three fully connected layers. The first two layers have 4096 neurons each, while the final layer has 1000 neurons, each corresponding to a class in the ImageNet dataset [19], ready for classification via a softmax function.

Residual Network

Residual Network (ResNet) [31, 32] designed by He et al. addressed some challenges like vanishing and exploding gradients as the number of

layers increased. ResNet addressed these issues, enabling the training of networks with depths as high as 152 layers, significantly deeper than previous architectures. Furthermore, it introduced residual blocks. Instead of directly learning the desired underlying mapping, these blocks attempt to learn the residual or the difference between the input and desired output. Mathematically, if the desired mapping is $H(x)$, the network learns the residual $F(x) = H(x) - x$, and the output becomes $H(x) = F(x) + x$. To facilitate the learning of residuals, ResNet introduces skip connections. These connections allow the output of one layer to be added to the output of another layer situated a few hops away, bypassing some layers in between. This design enables the flow of information through the network and mitigates the vanishing gradient problem. A bottleneck design was employed for the deeper ResNet architectures, like ResNet-50 and beyond. Each residual block in these networks consists of three layers: a 1×1, a 3×3, and another 1×1 convolution, which reduces the number of features temporarily, processes them, and then expands them, making computations more efficient.

Incenption V3

The Inception deep convolutional architecture was called GoogLeNet [75] and represents the Inception v1. Afterward, Inception v2 was introduced in [75], adding batch normalization. Later, in [77], Inception v3 was proposed with additional factorization concepts. In brief, the basic idea is factorizing convolution to reduce the number of connections and parameters without decreasing the network efficiency. This architecture employs four main kinds of modules: the first one (module 1) uses convolutional layers and implements small factorization convolutions; the second and third (modules 2 and 3) implement factorization into asymmetric convolutions; and the last one implements efficient grid size reduction. The final architecture consists of three Inception Module 1, one Grid Size Reduction Module, four Inception Module 2, one Grid Size Reduction Module, and two Inception Module 3.

Inception Residual Network

Inception-ResNet [76] was inspired by ResNet and Inception. There are two versions of this network architecture, namely v1 and v2. The architecture mainly employs six modules: Stem, Inception-resnet-A, Reduction-A, Inception-resnet-B, Reduction-B, and Inception-resnet-C. Inception-ResNet modules are similar to inception modules, adding the residual connection. Reduction modules are like inception modules, and the Stem module performs convolutions and spatial pooling. The final network configuration in InceptionResNetV2 consists of one Stem Module, one Inception-A block, ten Inception-ResNet-A blocks, one Reduction-A block, twenty Inception-ResNet-B blocks, one Reduction-B block, ten Inception-ResNet-C blocks, and a final convolution block.

Xception

Xception (Extreme Inception) [13] was inspired by Inception V3 and ResNet. It mainly consists of three main flow entries, middle, and exits repeated respectively one, eight, and one time. Each flow uses convolution with a receptive field of 3×3, spatial pooling, and separable convolution introduced in this architecture in inception-like. Furthermore, it has all residual connections.

Dense Convolutional Network

Dense Convolutional Network (DenseNet) [39] introduced a direct connection between any two layers with the same feature-map size. DenseNet mainly employs two components: dense block and transition layers. The architecture switches between them to build a deep network. In a dense block, each layer receives collective knowledge from preceding layers. Practically, each layer gets additional input from all preceding layers and gives its feature maps to all following layers. A transition layer controls the complexity of the model, reducing the number of channels by using 1x1 convolutional layers and halving the height and width of the average pooling layer.

NASNet

The authors in [94] proposed an architectural block of CNN using a deep reinforcement learning method. They specified the general architecture arranged as some normal cells followed by a reduction cell. They used a Recurrent Neural Network (RNN) to predict some network characteristics, such as the number of normal cells and the architecture of cells. A normal cell is a convolutional block that gives back a feature map of the same dimension, while a reduction cell is a convolutional block that gives back a feature map where the feature map height and width are reduced by a factor of two. They trained the first version of CNN on the CAFIR dataset and adapted it on ImageNet.

MobileNet

MobileNet [38] is a CNN architecture inspired by InceptionNet to work on mobile devices, with the primary goal of reducing the number of parameters and computations and preserving the performance as much as possible. The authors introduced a filter architecture called Depthwise Separable Convolution that split the calculation into two steps: (i) it applies a single convolutional filter for each input channel, and (ii) it uses pointwise convolution to create a linear combination of the output. In our study, we chose MobileNetV2 [68], an improvement of MobileNet in which the authors added an inverted residual connection and highlighted the importance of linear bottlenecks. This version has two main blocks, one with and one without residual connection. The final architecture consists of sixteen blocks.

EfficientNet

EfficientNet [78] is a family of models that are optimized to have few parameters and be faster. This model is scalable in depth, width, and resolution. The authors developed a baseline network using a multi-objective NAS [12]. The main layers are Mobile inverted Bottleneck Convolution (MBConv) and Squeeze Excite (SE). EfficientNetV2 [79] has been improved using progressive learning and replacing some MBConv layers with Fused-MB Conv [85]. It uses NAS to search for the best combination of fused and regular MB Conv Layers. In our work, we used EfficentNetV2L,

where L stands for large. This architecture is scaled up in-depth, width, and resolution. It consists of seven wider blocks for feature learning, followed by batch normalization, activation, and top layers.

3.2 Fast-HoVerNet for nuclei instance segmentation and classification.

This section introduces Fast-HoVerNet, our proposed nuclei instance segmentation and classification network. We obtained it using knowledge distillation techniques to mimic HoVerNet [27], one of this task's most used networks. In particular, we used a teacher-student architecture through an offline approach. HoVerNet is the teacher model in this setup, while a U-Net network, with a backend based on Mix Vision Transformer (MixViT) [84] backbone, is the student. We distilled HoVerNet branches (nuclei predictions (NP), horizontal and vertical outputs (HV), and type prediction (TP)) in a student with one decoder. As a distillation strategy, we used the logits predicted by HoVerNet in each branch so that our students could output the same kind of prediction maps using only one unit, as shown in Figure 3.4. An input image is given to the teacher (yellow component)

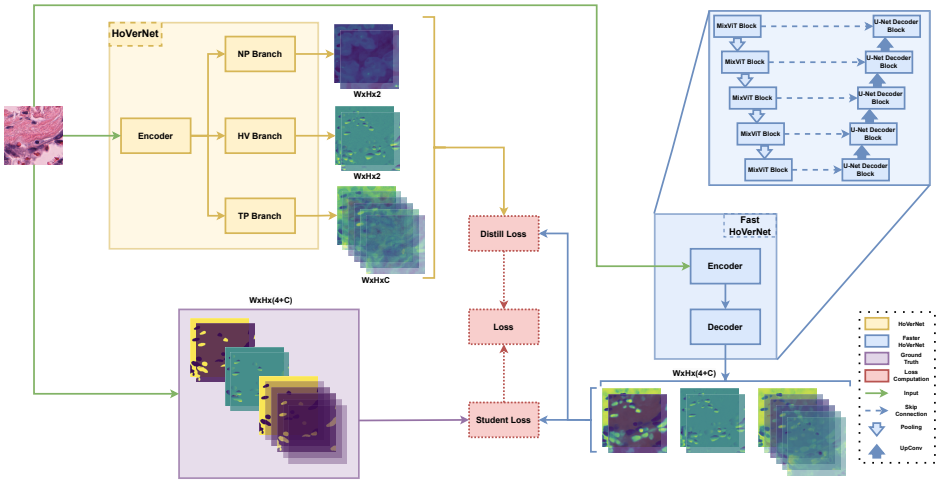


Figure 3.4. HoVerNet distillation framework.

and the student (blue part). Then, the distillation loss is computed between the teacher output and the student output, while the student loss is calculated between the generated ground truth (violet component) and student output. Lastly, both losses are combined to have the final loss (red components). Therefore, we aim to train a student to replace only the HoVerNet backbone, not its post-processing steps. In fact, we used the same post-processing of HoVerNet as described in [27]. Before describing our custom loss function used to distill HoVerNet, we briefly introduce KD.

Knowledge Distillation

Knowledge distillation, introduced by Hinton et al. [35], enables the training of a smaller network, referred to as a *student*, based on a more complex network known as a *teacher*. The core idea behind knowledge distillation is to use the teacher's predictions to guide and improve the student's learning process using a loss function that merges two hyper-parameters: 1) temperature (T) to smooth the predictions, allowing the student to learn without being influenced by any biases from the teacher, and 2) alpha (α), to combine the student loss, computed between the student's predictions and the ground truth, with the distillation loss. The distillation loss is calculated as the Kullback-Leibler divergence between the scaled softmax logits of the student and teacher, multiplied by $1/T^2$.

In recent years, significant advancements have been made in knowledge distillation techniques. Gou et al.[26] consider the teacher-student framework the most common architecture for knowledge distillation. Different types of knowledge, such as response-based, feature-based, or relation-based, can be distilled from the teacher to the student. Moreover, distillation techniques can be classified as offline, online, or self-distillation. Various algorithms have been proposed to facilitate knowledge distillation, catering to different requirements and scenarios. These include single-teacher, multi-teacher, adversarial, graph-based, attention-based, data-free, quantized, lifelong, and network adapter search-based algorithms. Each algorithm brings unique capabilities and advantages to the distillation process, allowing for efficient knowledge transfer from teacher to student.

3.2.1 Loss Fanction

To distill HoVerNet into a single-branch U-Net, we propose a custom loss function that draws inspiration from HoVerNet's loss function and the knowledge distillation theory [35]. Our loss function is a linear combination of two losses: the *student loss* between the student and the ground truth and the *distillation loss* between the student and the teacher. These two losses are combined using the α parameter, as shown in Equation 3.3. The student loss and distillation loss consist of a linear combination of six loss functions each. There are two loss functions for HV channels, two for HV channels, and two for NC channels, as illustrated in Equations 3.4 and 3.5, respectively.

Equation 3.3 represents the total loss, which is a combination of the student loss ($\mathscr{L}_{student}$) and the distillation loss ($\mathscr{L}_{distill}$). The α parameter is used to balance these two losses.

$$\mathscr{L} = \alpha \cdot \mathscr{L}_{student} + (1 - \alpha) \cdot \mathscr{L}_{distill} \tag{3.3}$$

Equations 3.4 and 3.5 break down student and distillation loss into their components. Each loss is a combination of losses from the HV, NP, and NC branches of the network, $\lambda_{a..f}$ are the weights assigned to each component in student loss, and $\lambda_{g..n}$ are the weights assigned to each component in distillation loss.

$$\mathscr{L}_{student} = \underbrace{\lambda_a \mathscr{L}_a + \lambda_b \mathscr{L}_b}_{\textbf{HoVer Branch}} + \underbrace{\lambda_c \mathscr{L}_c + \lambda_d \mathscr{L}_d}_{\textbf{NP Branch}} + \underbrace{\lambda_e \mathscr{L}_e + \lambda_f \mathscr{L}_f}_{\textbf{TP Branch}} \tag{3.4}$$

$$\mathscr{L}_{distill} = \underbrace{\lambda_g \mathscr{L}_g + \lambda_h \mathscr{L}_h}_{\textbf{HoVer Branch}} + \underbrace{\lambda_i \mathscr{L}_i + \lambda_l \mathscr{L}_l}_{\textbf{NP Branch}} + \underbrace{\lambda_m \mathscr{L}_m + \lambda_n \mathscr{L}_n}_{\textbf{TP Branch}} \tag{3.5}$$

Equation 3.6 represents the mean square error (MSE) computed for the horizontal and vertical maps. This measures the difference between the predicted (p_i) and actual values (q_i).

$$\mathscr{L}_a = \mathscr{L}_g = \frac{1}{n} \sum_{i=1}^{n} (p_i - q_i)^2 \tag{3.6}$$

Equation 3.7 represents the mean square gradient error (MSGE), which is the MSE computed on the gradients of the vertical and horizontal maps where the nuclei exist, as defined in [27].

$$\mathscr{L}_b = \mathscr{L}_h = \frac{1}{N}\sum_{i\in M}(\Delta_x(p_i) - \Delta_x(q_i))^2 + \frac{1}{M}\sum_{i\in M}(\Delta_y(p_i) - \Delta_y(q_i))^2 \quad (3.7)$$

Equation 3.8 represents the weighted cross-entropy loss, which measures the dissimilarity between the predicted and actual probability distributions. Where w_k is the weight for $k-th$ class computed as in Equation 3.9, where N is the total number of nuclei, and n_k is the total number of nuclei for $k-th$ class, and c is the number of classes.

$$\mathscr{L}_c = \mathscr{L}_e = \mathscr{L}_i = \mathscr{L}_m = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} w_k p_{i,k}\log q_{i,k} \quad (3.8)$$

$$\omega_k = \sqrt[2]{\frac{N}{n_k}}\cdot c \qquad w_k = \frac{\omega_k}{\sum_{i=1}^{c}\omega_i}\cdot c \quad (3.9)$$

Equation 3.10 represents the dice loss, a measure of the overlap between the predicted and actual binary masks.

$$\mathscr{L}_d = \mathscr{L}_f = \frac{1}{K}\sum_{k=1}^{K}\left[1 - \frac{2\times\sum_{i=1}^{N}(p_{i,k}\times q_{i,k}) + \epsilon}{\sum_{i=1}^{N} p_{i,k} + \sum_{i=1}^{N} q_{i,k} + \epsilon}\right] \quad (3.10)$$

Finally, Equation 3.11 represents the Kullback-Leibler (KL) divergence, which measures the difference between the predicted and actual probability distributions, where T is a temperature coefficient used to smooth teacher errors as in [35].

$$\mathscr{L}_l = \mathscr{L}_n = \frac{1}{N}\sum_{i=1}^{N} SoftMax(p_i/T)\cdot\log\frac{SoftMax(q_i/T)}{SoftMax(p_i/T)} \quad (3.11)$$

3.3 Cell Graph Representation and Classification

This section introduces our CG representation approach and the network used for cancer subtype classification.

3.3.1 Cell Graph Definition

We denote a CG as $G := (V, E, F)$, where V is the set of nodes that represents nuclei cells, E is the set of edges that represents the interaction between two cells, and F is the set of cells features. Each node $v \in V$ is represented by a feature vector $f(v) \in \mathbb{R}^d$, and $F \in \mathbb{R}^{|V| \times d}$, where $|\cdot|$ is a set cardinality. Furthermore, we denote an edge between two cells $(u, v) \in V$ as e_{uv}. The topology of the CG is described by a symmetric adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$, where $A_{u,v} = 1$ there is an edge between u and v.

3.3.2 Cell Graph Representation

As shown in Section 2.3, a common approach to extracting cell features consists of detecting and segmenting each nucleus using a specialized CNN like HoVerNet and then cropping each one. Consequently, these approaches use a pre-trained CNN, like ResNet, to obtain a one-dimensional feature for each cropped nucleus. To unify the process and reduce both computational time and the space to store each image, we propose to employ the same network used for nuclei detection as a feature extractor. First, we did not use HoVerNet but Fast-HoVerNet, which we introduced and described in Section 3.2. We made this decision because, in the experimental results, our version reached similar results in classification and instance segmentation, but it is three times faster. Therefore, the feature extraction process consists of using the feature map obtained from the second last layer of Fast-HoVerNet, denoted with Φ, which has shape $H \times W \times 16$, and the bounding boxes obtained from post-processing, denoted with b_i with $i \in 1, ..., N$, where N is the number of nuclei extracted from images. Here, we define an operation Θ that uses a feature map and bounding boxes to obtain a cropped feature map for each nucleus, denoted with ϕ_i, which has shape $w_i \times h_i \times 16$ where w_i and h_i are respectively the

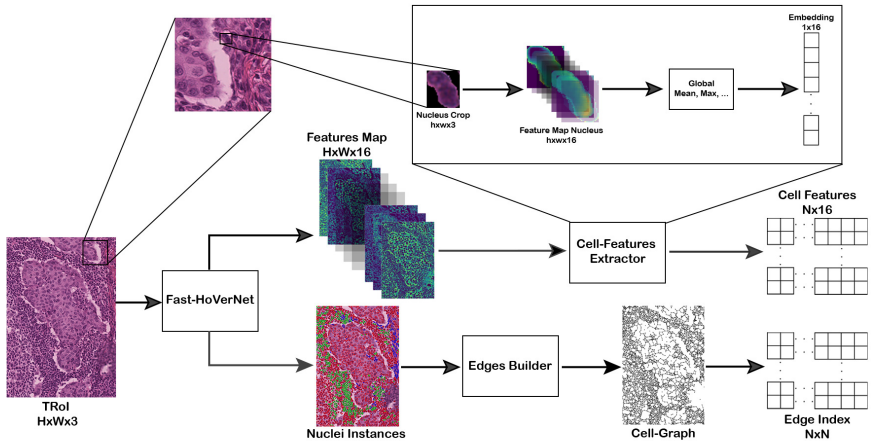


Figure 3.5. Our Cell-Graph building method

weight and height of bounding box b_i, as shown in Equation 3.12.

$$\Theta(\Phi, b_i) \rightarrow \phi_i \; i \in 1, ..., N \tag{3.12}$$

Therefore, ϕ_i represents the morphological feature of each nucleus. Furthermore, as shown in other works, we used spatial features as normalized centroids on TRoI, denoted as c_i. At this point, each nucleus is represented with a 3-D tensor with different shapes in width and height. To obtain a 1-D tensor, we defined a cell feature, denoted as f_i, as a concatenation of normalized centroids and statistical characterizations computed for each channel to obtain a tensor of 1×16. Equation 3.13 reports the final feature formulation, where $S(f_i)_j$ is a statistical function like mean, standard deviation, max, and sum. Therefore, f_i has shape $(2 + 16 \cdot K) \times 1$ where K is the number of statical functions used to characterize each f_i.

$$f_i = CONCAT([c_i, S(f_i)_0, ..., S(f_i)_j] \; j \in K \tag{3.13}$$

Therefore, for each TRoI, we have a CG where $f_i \in F$.

Based on the assertion that close cells have stronger interaction made by Francis et al. [23], and accordingly with Pati et al. [60], we realized the initial graph topology using a KNN algorithm and that we pruned remov-

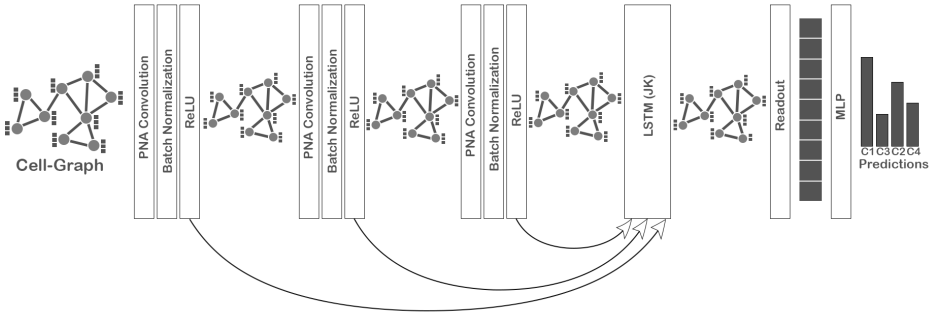


Figure 3.6. Our Graph Neural Network architecture

ing edges longer than a threshold distance d_{min}. The distance between two cells is computed using Euclidean distance between centroids. Therefore, as also defined in [60], for each cell v there is an edge $e_e u$ if:

$$u \in \{w | dist(v,w) <= d_k \wedge dist(v,w) < d_{min},$$
$$\forall w, v \in V, d_k = k^{th} \text{smallest distance in } dist(v,w)\}$$

We summarized our method for CG building in Figure 3.5.

3.3.3 Cell Graph Classification

A typical GCN architecture for graph classification consists of some graph convolutional layers, readout operation, and an Multi Layer Perceptron (MLP). Graph convolutional layers learn node feature; readout operation aggregates the node feature in a 1-D tensor and MLP ahead work as a classifier. Our model uses Principal Neighbourd Aggregator (PNA) [17] as GCN and mean-pooling as a readout operation. Lastly, we used a classic MLP as a classifier. Furthermore, we also use a skip connection technique called Long Short Term Memory (LSTM) attention JK [87]. To clarify our approach, we briefly describe each component in our architecture. PNA is a spatial graph convolution operator that addresses the variance in the number of neighbors each node might have. PNA provides a more comprehensive understanding of neighborhood aggregation mechanisms by incorporating various aggregation functions such as sum, mean, minimum, maximum, and standard deviation using degree scalar to

mitigate potential adverse effects from nodes having disparate neighbors. Equation 3.14 contains the scaler formula, where δ is in Equation 3.15, and α is a variable between -1 and 1.

$$S(d,\alpha) = \left(\frac{\log(d+1)}{\delta}\right)^{\alpha},\ d>0,\ \alpha \in [-1,1] \tag{3.14}$$

$$\delta = \frac{1}{|train|}\sum_{i \in train} \log(d_i+1) \tag{3.15}$$

The PNA operator is defined by $\bigoplus$ in Equation 3.16, where $\otimes$ denoted the product between matrices, and the second matrix includes the aggregators.

$$\bigoplus = \begin{bmatrix} I \\ S(D,\alpha=1) \\ S(D,\alpha=-1) \end{bmatrix} \otimes \begin{bmatrix} \mu \\ \sigma \\ max \\ min \end{bmatrix} \tag{3.16}$$

The PNA convoutaial layer is in Equation 3.17

$$X_i^{(t+1)} = U\left(X_i^{(t)}, \bigoplus M(X_i^{(t)}, E_{j\to i}, X_j^{(t)})\right) \tag{3.17}$$

To summarize all, Figure 3.6 shows our architecture. Lastly, during the training of our network, we employed weighed cross-entropy loss to mitigate unbalanced classes, where weights are computed as reported in Equation 3.18.

$$\omega_k = \sqrt[2]{\frac{N}{n_k}} \cdot c \qquad\qquad w_k = \frac{\omega_k}{\sum_{i=1}^{c}\omega_i} \cdot c \tag{3.18}$$

Chapter 4

Experimental Results

This chapter contains our experimental results for each module introduced in Chapter 3. In order to make the chapter more understandable, we first introduce all datasets used in this thesis, and later, we report the obtained experimental results divided as follows:

- TRoIs Retrieval:
  - Experimental results to identify the best CNN and its best layer to represent a TRoI;
  - Experimental results to estimate the effects of color stain normalization in our system;
- Fast-HoVerNet:
  - Experimental results to identify the best backbone and the best Knowledge distillation hyperparameters;
  - Experimental results to compare Fast-HoVerNet against the SOTA nuclei instance segmentation and classification networks;
- CG Classification:
  - Experimental results to compare SOTA GNNs using our CG representation.

4.1 Datasets

In this work, we used different well-known datasets based on the task we had to work on. As shown in Table 4.1, we used BACH [4] for CBIR, Pannuke [24] and Colorectal Nuclear Segmentation and Phenotypes (CoNSeP) [27] for nuclei instance classification and segmentation, and BRACS [6] for CG classification. In the following, we describe each dataset.

Task	**Dataset**
CBIR Retrieval	BACH [4]
Nuclei Instance segmentation and classification	Pannuke [24], CoNSeP [27]
CG Classification	BRACS [6]

Table 4.1. Addressed tasks and related datasets

4.1.1 BACH

The BACH dataset was created as part of the BACH challenge. This challenge is divided into two parts. The first one is focused on the automatic classification of H&E stained breast histology microscopy images into four distinct classes: normal, benign, in situ carcinoma, and invasive carcinoma. The second one was centered around performing pixel-wise labeling whole-slide breast histology images in the same four classes mentioned above. BACH provided two labeled training datasets. The first has microscopy images annotated image-wise by two expert pathologists from the Institute of Molecular Pathology and Immunology of the University of Porto (IPATIMUP) and the Institute for Research and Innovation in Health (i3S). The provided images are in RGB .tiff format and have a size of 2048 $\times$ 1536 pixels and a pixel scale of 0.42 μm $\times$ 0.42 μm. The second dataset contained pixel-wise annotated and non-annotated WSIs, where the annotations were designed to identify regions of interest for diagnosis at the lowest magnification setting. This means that the annotations might include non-tissue and normal tissue regions. WSIs were acquired using a Leica SCN400. These images were available in the .svs format, with a pixel scale of 0.467 μm/pixel. The dimensions of these images var-

ied, with widths ranging from 39,980 to 62,952 pixels and heights ranging from 27,972 to 44,889 pixels.

In this work, we used the first part of the dataset that contains only images; we found only the public train test, which consists of 400 images divided equally for each class. Figure 4.1 shows an example for each class.

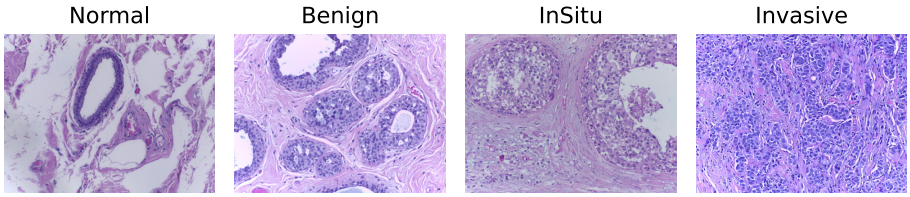


Figure 4.1. BACH dataset example

4.1.2 CoNSeP

CoNSeP dataset, proposed in [27] has 41 H&E stained image tiles with a size of $1,000 \times 1,000$ pixels at $40\times$ objective magnification. These images were obtained from 16 colorectal adenocarcinomas (CRA) WSIs. They were scanned using an Omnyx VL120 scanner at the Department of Pathology at University Hospitals Coventry and Warwickshire, UK. The dataset specifically focuses on colorectal adenocarcinoma to showcase the true tissue variation within this type of cancer, as opposed to datasets that pull from various cancer types. CoNSeP captures a variety of tissue components, including stroma, glandular, muscular, collagen, fat, and tumor regions. It also features different nuclei types like normal epithelial, tumor epithelial, inflammatory, necrotic, muscle, and fibroblast, where "type" indicates the cell from which the nucleus originates. Every nucleus in this dataset was annotated by one of two expert pathologists, and post-annotation, each sample underwent a review by both pathologists to refine the annotations, ensuring consensus. Alongside marking the nuclear boundaries, each nucleus received a label, such as normal epithelial, malignant/dysplastic epithelial, fibroblast, muscle, inflammatory, endothelial, or miscellaneous. The miscellaneous category encompasses necrotic, mitotic nuclei, or couldn't be categorized. Specific nuclei types were grouped

for experimental purposes, like the normal and malignant/dysplastic epithelial nuclei and the fibroblast, muscle, and endothelial nuclei, termed spindle-shaped nuclei. Figure 4.2 show some example of it. Furthermore, Table 4.2 shows its statistics.

Table 4.2. CoNSeP statistics

Set	Malignant	Normal	Endothelial	Miscellaneous	Fibroblas	Muscle	Inflammatory	Images
Train	371	3941	4765	901	40	4420	1117	27
Test	561	1638	2789	495	80	2711	503	14
Total	932	5579	7554	1396	120	7131	1620	41

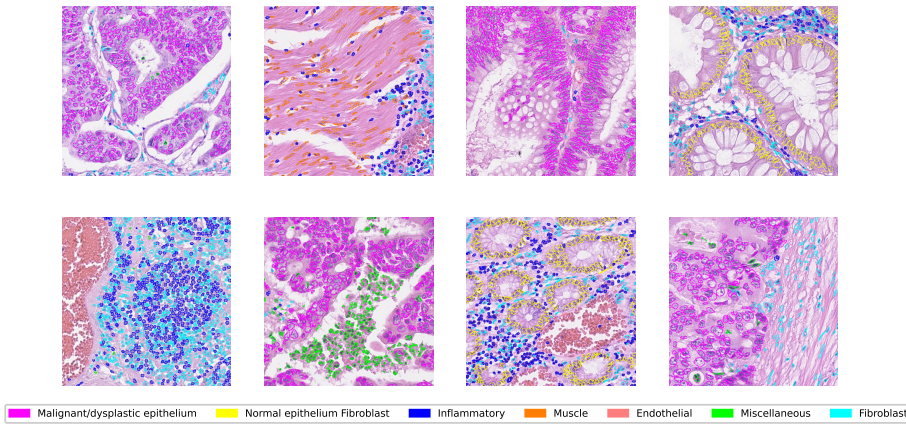


Figure 4.2. CoNSeP examples

4.1.3 Pannuke

The PanNuke dataset [24] is an extensive and diverse pan-cancer collection that has undergone clinical quality control. It was developed to address the challenges deep learning models face when applied to real-world CPATH applications. The dataset was generated by aggregating a set of publicly available nucleus classification and detection datasets. An initial dataset was created for semi-automatic ground truth generation. A fully convolutional neural network (FCNN) was trained for nucleus detection using four publicly available datasets: Kumar [47], CPM2017 [83], 15 visual fields from TCGA [50] that were labeled independently, and a

dataset of bone marrow visual fields [42]. For the final version of PanNuke, the NuClick [45] method was used to generate an accurate segmentation mask from a single point. The iterative process led to a dataset comprising 481 visual fields with 189,744 exhaustively annotated nuclei. Domain experts verified these annotations. PanNuke is the largest and most miscellaneous dataset for nucleus segmentation and classification. It has been annotated in a semi-automated manner and quality-controlled by clinical professionals. The dataset covers 19 different tissue types, making it a valuable resource for developing and testing algorithms in computational pathology. Table 4.3 shows statistics for each fold provided by authors and the whole dataset dimension. In particular, each image in the Pannuke is RGB and has shape 256×256.

Table 4.3. Pannuke statistics

Organ	Number of Images	Neoplastic	Inflammatory	Connective/Soft Tissue	Dead	Ephitelial
Adrenal_gland	437	83812	522838	985	459440	25079222
Bile-duct	420	496590	1196924	998	142847	22612337
Bladder	146	56974	384848	0	2561	7178761
Breast	2351	2030240	5090469	1896	5490123	126504089
Cervix	293	399410	450165	12062	94624	14709725
Colon	1440	2957938	4373202	62513	4350019	743100116
Esophagus	424	92026	1203046	20619	83002	22641215
HeadNeck	384	589725	680528	13752	50132	19974420
Kidney	134	60577	204595	0	51501	7672125
Liver	224	85797	506991	39	293607	12666111
Lung	184	59702	362274	185399	0	9717862
Ovarian	146	64220	547842	0	160994	7223717
Pancreatic	195	220716	952848	0	160923	10969715
Prostate	182	13028	379214	0	95418	9917707
Skin	187	811058	273426	0	185742	9607483
Stomach	146	530451	414273	25448	0	7605568
Testis	196	339941	471791	0	290862	10763905
Thyroid	226	179128	582094	780	895447	12467208
Uterus	186	123166	729549	30532	7897	16800255
Total	6078	6749258	21383232	287379	10020795	399437691

4.1.4 BRACS

The BRACS dataset [6] was created to support the development of breast cancer diagnostic methods by automatically analyzing histology images. The dataset was built through the collaboration of the National Cancer Institute—Scientific Institute for Research, Hospitalization and Healthcare (IRCCS) 'Fondazione G. Pascale,' the Institute for High-Performance Computing and Networking (ICAR) of National Research Council (CNR), and International Business Machines (IBM) Research—Zurich. The dataset

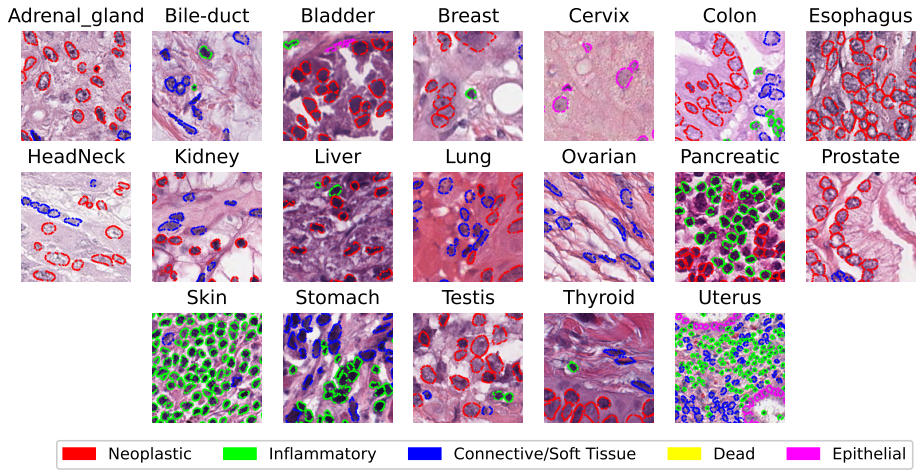


Figure 4.3. Pannuke examples

was acquired from patients between 2019 and 2020 by board-certified clinicians of the Department of Pathology at the National Cancer Institute—IRCCS Fondazione G. Pascale in Naples, Italy. The samples were generated from H&E breast tissue biopsy slides and were selected based on the patient's diagnostic reports. BRACS is provided in two versions. In the first version, only TRoIs are included. In the second version, also WSIs are in. The TRoIs have variable sizes to ensure the entire diagnostic lesion is included, preventing the loss of diagnostically relevant information. The number of extracted TRoI per WSI ranges from 0 to 119, with an average of 11. This classification included categories such as Normal Tissue (N), Pathological Benign (PB), Usual Ductal Hyperplasia (UDH), Atypical Ductal Hyperplasia (ADH), Flat Epithelial Atypia (FEA), Ductal Carcinoma in Situ (DCIS), and Invasive Carciname (IC). Figure 4.4 shows one example per class. Furthermore, Table 4.4 shows the statistics of the dataset.

- N: in normal mammary glandular tissue, there are two types of epithelial cells: the luminal layer and the basal myoepithelial layer. Additionally, there are two types of stromal cells: interlobular stroma and intralobular stroma. Unlike Pathological Benign, the epithelial

Table 4.4. BRACS dataset statistics

	Normal	Benign	UDH	ADH	FEA	DCIS	Invasive	Total
Image								
Images	512	758	471	568	783	749	550	4391
Pixels (in million)	2.8±2.7	5.7±4.5	2.4±2.9	2.2±2.0	1.2±1.1	5.0±5.0	8.2±5.4	3.9±4.3
Max/Min pixel ratio	75.3	97.9	180.1	75.3	58.3	128.6	62.4	235.6
Image split								
Train	342	586	303	405	599	562	366	3163
Validation	86	87	88	77	85	97	82	602
Test	84	85	80	86	99	90	102	626
WSI split								
Train	67	86	59	38	37	33	41	198
Validation	28	24	24	28	17	21	19	68
Test	15	16	20	17	12	16	16	59

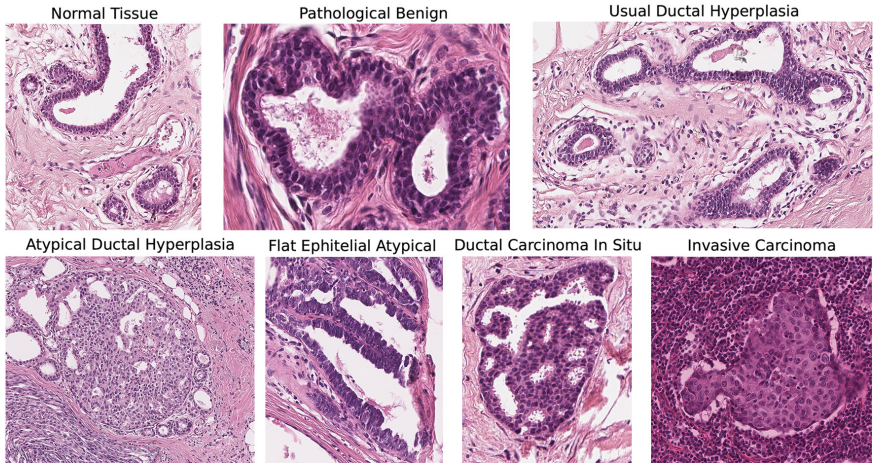


Figure 4.4. Bracs examples

component and stroma ratio are preserved.

- PB: this category included both non-proliferative lesions and proliferative lesions, except for Usual Ductal Hyperplasia, Flat Epithelial Atypia, and Atypical Ductal Hyperplasia, which were considered as three independent subtypes. Therefore, Pathological Benign includes cyst, apocrine metaplasia, ductal ectasia, squamous metaplasia, atrophy, stromal fibrosis, mastitis, sclerosing adenosis, papilloma, radial scar, and simple and complex fibroadenoma.

- UDH: it has a rate of occurrence of 20%. An increase in the epithelial layers characterizes it and is a cohesive proliferation of disorderly distributed but oriented cells. UDH can have different architectural aspects, such as solid, fenestrated, and micropapillary patterns. Even though UDH shares some architectural features with ADH and DCIS, it does not show atypia.

- FEA: it represents 3.8–10% of core needle biopsy samples. It is a proliferative lesion characterized by low-grade cytological atypia, cell monomorphism, loss of polarity and orientation concerning the basement membrane, presence of apical snout, endoluminal secretion, and frequent calcifications.

- ADH: it is a proliferation of monomorphic cells, which only partially fill the ductal spaces. Its architectural aspects include a solid pattern, a cribriform pattern, and a papillary pattern. The cytologic atypia in ADH is similar to that of low-grade DCIS. Still, the lesion spans no more than 2 mm or has insufficient architectural atypia involving only partial ducts and/or lobules.

- DCIS: it is a malignant proliferation of epithelial cells that fills the entire duct without evidence of stroma invasion. Typically, it involves multiple adjacent ductal spaces. It can have patterns like cribriform, solid, papillary, and micropapillary.

- IC: it is characterized by the invasion of tumor cells infiltrating the breast stroma with a loss of peripheral myoepithelial cells. The presence of the myoepithelial cell layer is an essential distinction from DCIS.

4.2 TRoI CBIR: Experimental results

This section shows two studies on our TRoI CBIR. The first study aims to detect the best pre-trained CNN and which layer is the best feature extractor in our pathological domain. The second study seeks to estimate the effects of color stain normalization in our framework. Before showing the experimental results, we introduce the architecture of each pre-trained CNN we chose according to the SOTA. For each one, we briefly describe

its architecture and a table to summarize the selected layers with their feature shape.

4.2.1 CNN description

VGG-Net

For VGG16, we consider the output of each block as a deep feature, applying global max/average pooling or flattened operation, as shown in Table 4.5. To summarize, we select layers with output shape and feature size after dimensional reduction where block_i_pool is the last layer of the i-th block.

Table 4.5. Deep feature size VGG16 for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
block1_pool	112x112x64	64	802816
block2_pool	56x56x128	128	401408
block3_pool	28x28x256	256	200704
block4_pool	14x14x512	512	100352
block5_pool	7x7x512	512	25088

Inception V3

We choose as deep features the outputs of each module applying global max/average pooling or flattened operation, as shown in Table 4.6. To summarize, we select layers with output shape and feature size after dimensional reduction, where mixed is the output of the i-th module.

Residual Network

We considered the output of each convolutional block as a deep feature applying global max/average pooling or flattened operation, as shown in Table 4.7. To summarize, we select layers with output shape and feature size after dimensional reduction where conv_i_block$j$_out is the j-th output of the i-th module.

Table 4.6. Deep feature size Inception V3 for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
mixed0	35x35x256	256	313600
mixed1	35x35x288	288	352800
mixed2	35x35x288	288	352800
mixed3	17x17x768	768	221952
mixed4	17x17x768	768	221952
mixed5	17x17x768	768	221952
mixed6	17x17x768	768	221952
mixed7	17x17x768	768	221952
mixed8	8x8x1280	1280	81920
mixed9	8x8x2048	2048	131072
mixed10	8x8x2048	2048	131072

Table 4.7. Deep feature size ResNet152V2 for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
conv2_block3_out	28x28x256	256	200704
conv3_block8_out	14x14x512	512	100352
conv4_block36_out	7x7x1024	1024	50176
conv5_block3_out	7x7x2048	2048	100352

Inception Residual Network

We choose the output of the Inception-A block, Reduction-A block, Reduction-B block, and final convolution applying on each one global max/average pooling or flattened operation as deep features. Table 4.8 summarizes the selected layers with output shape and feature size after dimensional reduction where mixed_5b corresponds to 1, mixed_6a to 2, mixed_7a to 3, and conv_7b to 4.

Table 4.8. Deep feature size Inception-ResNetV2 for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
mixed_5b	35x35x320	320	392000
mixed_6a	17x17x1088	1088	314432
mixed_7a	8x8x2080	2080	133120
conv_7b	8x8x1536	1536	98304

Xception

We select as deep features the output of entry flow, the eight outputs of middle flow, and the output of exit flow, applying global max/average pooling or flattened operation to each considered layer. The table 4.8 summarizes the selected layers with output shape and feature size after dimensional reduction where add_2 corresponds to the output of entry flow, add_10 to the output of the last middle flow, and block_14_sepconv2 to the output of the exit flow.

Table 4.9. Deep feature size Xception for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
add_2	19x19x728	728	262808
add_10	19x19x728	728	262808
block14_sepconv2	10x10x2048	2048	204800

Dense Convolutional Network

In our study, we choose DenseNet201, that have two hundred-one layers, considering the output of each transition layer and the last dense block after global average/max pooling or flatten operation as deep features. The table 4.8 summarizes the selected layers with output shape and feature size after dimensional reduction where pool_i corresponds to output i-th transaction layer, and con5_block32_concat is the output of the last dense block.

Table 4.10. Deep feature size DenseNet201 for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
pool2_pool	28x28x128	128	100352
pool3_pool	14x14x256	256	50176
pool4_pool	7x7x896	896	43904
conv5_block32_concat	7x7x1920	1920	94080

NASNet

We chose NasNetLarge, a version trained on ImageNet, and we considered the output of each block of normal cells and reduction cells as a deep feature after applying global max/average pooling or flatten operation. Table 4.11 summarizes the selected layers with output shape and feature size after dimensional reduction, where layers normal_concat_5, normal_concat_12, and normal_concat_12 are the last layers of a series of normal cells. At the same time, normal_concat_reduce_6 and normal_concat_reduce_12 are the previous layers of reduction cells.

Table 4.11. Deep feature size NASNetLarge for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
normal_concat_5	42x42x1008	1008	1778112
reduction_concat_reduce_6	21x21x1344	1344	592704
normal_concat_12	21x21x2016	2016	889056
reduction_concat_reduce_12	11x11x2688	2688	325248
normal_concat_18	11x11x4032	4032	487872

MobileNet

In our study, we chose MobileNetV2 [68], an improvement of MobileNet in which the authors added an inverted residual connection and highlighted the importance of linear bottlenecks. This version has two main blocks, one with and one without residual connection. The final architecture consists of sixteen blocks. We considered the output of each block and the last

one, excluding dense layers as a deep feature after applying global max/average pooling or flatten operation. Table 4.12 summarizes selected layers with output shape and feature size after dimensional reduction, where block_i_project_BN is the last layer of an i-th block without residual connection, block_i_add is the last layer of i-th block with residual link, and out_relu is the previous layer for feature extraction.

Table 4.12. Deep feature size MobileNetV2 for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
block_1_project_BN	56x56x24	24	75264
block_2_add	56x56x24	24	75264
block_3_project_BN	28x28x32	32	25088
block_4_add	28x28x32	32	25088
block_5_add	28x28x32	32	25088
block_6_project_BN	14x14x64	64	12544
block_7_add	14x14x64	64	12544
block_8_add	14x14x64	64	12544
block_9_add	14x14x64	64	12544
block_10_project_BN	14x14x96	96	18816
block_11_add	14x14x96	96	18816
block_12_add	14x14x96	96	18816
block_13_project_BN	7x7x160	160	7840
block_14_add	7x7x160	160	7840
block_15_add	7x7x160	160	7840
block_16_project_BN	7x7x320	320	15680
out_relu	7x7x1280	1280	62720

EfficientNet

We chose the output of each block and the last one, excluding dense layers as an in-depth feature after applying global max/average pooling or flatten operation. Table 4.13 summarizes the selected layers with output shape and feature size after dimensional reduction, where blockij_add is the output of the i-th block, j stands for the last sub-blocks that envelope

it in width, and top_rule is the last layers excluding top classification layers.

Table 4.13. Deep feature size EfficientNetV2L for each selected layer

Layer	Output shape	Feature size	
		Max/Avg	Flatten
block1d_add	240x240x32	32	1843200
block2g_add	120x120x64	64	921600
block3g_add	60x60x96	96	345600
block4j_add	30x30x192	192	172800
block5s_add	30x30x224	224	201600
block6y_add	15x15x384	384	86400
block7g_add	15x15x640	640	144000
top_activation	15x15x1280	1280	288000

4.2.2 Evaluation Strategy

Here, we present and discuss our experimental strategy. We conducted several experiments to identify the best CNN, particularly its best-performing layer with related dimensionality reduction techniques, and to estimate the effects of CSN. Different metrics such as precision, precision at k, recall, f-measure, precision-recall curve, Mean Average Precision, and mean Average Precision at K are presented in literature [66] and, we used precision at k (P@k) because we are interested in the first k relevant results. In detail, as reported in Equation 4.1, P@k calculates how many retrieved items on K top-ranked ones are relevant for a given query. We computed the P@k on four values of K: 5, 10, 50, and 100.

$$P@k = I_r/K \tag{4.1}$$

Moreover, we also consider the average of P@K (MAP@k) to evaluate the results from all queries. Furthermore, we used a confusion matrix to analyze the results and understand how the CBIR task works for each category available in the used datasets. In particular, rows of the confusion matrix contain the query category, and the columns are the category of retrieved images. Each value concerns precision at k of j-th category retrieved for

the i-th category queried, i is the index of the row, and j of the column. We used each image dataset of BACH dataset as a query by example, obtaining for each experiment 400 queries. The image used as the query is left out from the results set. In order to perform a robust evaluation, we focused on analyzing three crucial aspects to justify the choice made by quantifying the loss/gain of the dimensionality reduction method, the layer selection for each CNN, and, ultimately, the CNN architecture.

4.2.3 Preprocessing

The images given in input to Convolution Neural Networks, especially for a pre-trained one, must be processed to have the correct representation according to the training format. In particular, the input must be equal to the one used in the training step, and each pixel value must be normalized according to the used architecture. In particular, each input has three channels because images are RGB, and it must be 224x224 for VGG16, ResNet152V2, MobileNetV2, and DenseNet201, 229x299 for InceptionV3, InceptionResNetV2 and Xception, 331x331 for NASNetLarge, and 480x480 for EfficientNetV2. Table 4.14 summarizes the input shape and the operation computed by the preprocessing pipeline.

4.2.4 Results: deep feature extractor identification

Here, we show the results we did to investigate which is the best CNN and its layer in our TRoIs CBIR as a feature extractor. Firstly, we analyzed all results identified by each CNN and considered layer and reduction operations to find the best way to reduce the feature map from a 3-D to a 1-D array. On the other hand, we set the reduction operation and analyzed the result to recognize the best layer for each CNN to extract the deep features. Eventually, we compared all CNNs to find the best result. We quantified the loss/gain in terms of P@k in each analysis.

Dimensionality reduction methods

We computed P@5, P@10, P@50, and P@100 for each CNN layer and reduction operation to identify the best dimensionality reduction method. Our experiments show that the global average pooling obtains, on average, the best results on each layer at each precision. To quantify the

Table 4.14. Input size and preprocessing input function for each CNN

CNN	Input Shape	Preprocess Input Operations
VGG16 [72]	224x224	It converts RGB to BGR. The images are converted from RGB to BGR, and each color channel is zero-centered concerning the ImageNet dataset without scaling.
InceptionV3 [77]	299x299	The inputs pixel values are scaled between -1 and 1, sample-wise.
ResNet152V2 [32]	224x224	The inputs pixel values are scaled between -1 and 1, sample-wise.
InceptionResNetV2 [76]	299x299	The inputs pixel values are scaled between -1 and 1, sample-wise.
MobileNetV2 [68]	224x224	The input pixel values are scaled between -1 and 1, sample-wise.
DenseNet201 [39]	224x224	The input pixel values are scaled between 0 and 1, and each channel is normalized concerning the ImageNet dataset.
Xception [13]	299x299	The input pixel values are scaled between -1 and 1, sample-wise.
NasNetLarge [94]	331x331	The inputs pixel values are scaled between -1 and 1, sample-wise.
EfficientNetV2L [79]	480x480	Nothing

improvement of global average pooling rather than global max pooling or flattening, we defined the loss/gain precision at k (GLP@k) for each precision level and reduction method as reported in Equation 4.2. It computes the difference between P@k for global average pooling and P@k for flattening, where m is the reduction method for which we want to measure the gain or loss of precision of k concerning the global average pooling.

$$GLP_m@k = P_{avg}@k - P_m@k \tag{4.2}$$

Table 4.15 summarizes the GLP@k for each CNN, where we calculated each value as the average of GLP@k on each layer. A positive value means that global average pooling has a gain. Otherwise, it has a loss. Results show that the global average pooling is the most suitable for k equal to 5 and

10, while for k equal to 50 and 100, we have a slight loss for ResNet152V2, there is a slight loss. Therefore, from a global point of view and our interest in having a high precision in the first k results, the global average pooling is the best choice to maximize the precision at each level.

Table 4.15. Average Gain/Loss P@k quantification by global average pooling for each CNN

CNN	Reduction	Avg-GLP@5	Avg-GLP@10	Avg-GLP@50	Avg-GLP@100
DenseNet201	flatten	0.1555	0.1281	0.0597	0.0428
	max	0.0526	0.0431	0.0091	0.0066
EfficientNetV2L	flatten	0.1905	0.1381	0.0560	0.0338
	max	0.1978	0.1571	0.0852	0.0568
InceptionResNetV2	flatten	0.1395	0.1018	0.0438	0.0311
	max	0.0675	0.0419	0.0142	0.0091
InceptionV3	flatten	0.2195	0.1810	0.0936	0.0639
	max	0.0790	0.0617	0.0273	0.0195
MobileNetV2	flatten	0.1566	0.1195	0.0509	0.0333
	max	0.1172	0.0873	0.0345	0.0210
NASNetLarge	flatten	0.1960	0.1639	0.0904	0.0608
	max	0.0902	0.0716	0.0367	0.0233
ResNet152V2	flatten	0.1816	0.1464	0.0685	0.0435
	max	0.0359	0.0248	**-0.0052**	**-0.0081**
VGG16	flatten	0.1886	0.1487	0.0671	0.0425
	max	0.0661	0.0452	0.0107	0.0046
Xception	flatten	0.1718	0.1393	0.0734	0.0530
	max	0.0462	0.0307	0.0105	0.0068

Layer selection for each CNN

To find the best layers for each CNN, we computed the P@k for each layer, and if the best one was the same for each level of precision, we chose it. Otherwise, we estimated the loss gain P@k to find the best layer, which minimizes the loss. As summarized in Table 4.16, for DenseNet, EfficientNetV2, InceptionResNetV2, InceptionV3, ResNet152V2 and Xception, the best layers are respectively conv5_block32_concat, block6y_add, mixed_7a,

Table 4.16. The best CNNs layer for each precision level

CNN	Best Layer (P@5)	Best Layer (P@10)	Best Layer (P@50)	Best Layer (P@100)
DenseNet201	conv5_block32_concat (0.67)	conv5_block32_concat (0.626)	conv5_block32_concat (0.4839)	conv5_block32_concat (0.4123)
EfficientNetV2L	block6y_add (0.7365)	block6y_add (0.671)	block6y_add (0.4998)	block6y_add (0.4192)
InceptionResNetV2	mixed_7a (0.68)	mixed_7a (0.6325)	mixed_7a (0.495)	mixed_7a (0.4177)
InceptionV3	mixed6 (0.693)	mixed6 (0.6298)	mixed6 (0.4864)	mixed6 (0.411)
MobileNetV2	block_13_project_BN (0.6655)	block_13_project_BN (0.604)	block_14_add (0.471)	out_relu (0.4025)
NASNetLarge	Normalizational_concat_12 (0.695)	Normalizational_concat_12 (0.6295)	Normalizational_concat_12 (0.4988)	reduction_concat_reduce_12 (0.4197)
ResNet152V2	conv4_block36_out (0.6475)	conv4_block36_out (0.5923)	conv4_block36_out (0.4586)	conv4_block36_out (0.3902)
VGG16	block4_pool (0.6125)	block5_pool (0.562)	block5_pool (0.4532)	block5_pool (0.3871)
Xception	add_10 (0.671)	add_10 (0.6185)	add_10 (0.4771)	add_10 (0.4051)

Table 4.17. Gain/Loss Precision at k for CNN where the best layer is not equal for each k.

CNN	Layer	Layer1	P@5	P@10	P@50	P@100
MobileNetV2	block_14_add	block_13_project_BN	**-0.0050**	**-0.00400**	0.01255	0.007475
		out_relu	0.0270	0.00850	0.00140	**-0.003925**
NASNetLarge	Normalizational_concat_12	reduction_concat_reduce_12	0.0295	0.00900	0.00535	**-0.000275**
VGG16	block5_pool	block4_pool	**-0.0130**	0.01575	0.03430	0.024600

mixed6, conv4_block36_out and add_10 at each level of precision; for MobileNetV2, the best layer is block_13_project_BN for precision equal to 5 and 10, block14_add for precision equal to 50, and out_relu at precision equal to 100; for NasNetLarge, the best layer is normal_concat_12 for precision equal to 5, 10 and 50, and reduction_concat_reduce_12 for precision equal to 100; for VGG16, the best layer is block4_pool for precision equal to 5 and block5_pool for precision equal to 10, 50 and 100. Table 4.17 shows the gain/loss precision at k for MobileNet, NasNetLarge, and VGG16. According to the results, we chose block_14_add for MobileNetV2 because it outperforms out_relu and has a slight gain on block_13_project_BN. We chose normal_concat_12 for NASNetLarge and block5_pool for VGG16 because they have better results than the other layers.

Best Convolutional Neural Network Recognition

To find the best CNN to use as a feature extractor, we computed the P@k for each one, fixing the reduction method with global average pooling and the best layer according to the ones above recognized. Table 4.18 shows the results for each configuration, highlighting that the best CNN is EfficientNetV2. It is best for k equal to 5 and 10, but for k equal to 50 and 100, the gap with other CNN is slight, as displayed in Figure 4.5. We remark that we are interested in high precision on top K results.

We analyzed the best CNN configurations in depth using a confusion matrix to accomplish a more precise analysis. We intend to understand which dataset category is not correctly retrieved. According to Figure 4.6, it is clear that the queries using benign examples are not correctly retrieved. They are often misunderstood with in situ images and normal ones. Furthermore, normal and invasive queries are usually recognized with good precision value. In particular, all networks have good performance

Table 4.18. Average P@k for each chosen layer

Model	Layer	P@5	P@10	P@50	P@100
VGG16	block5_pool	0.5995	0.56200	0.45320	0.387075
ResNet152V2	conv4_block36_out	0.6475	0.59225	0.45855	0.390225
DenseNet201	conv5_block32_concat	0.6700	0.62600	0.48390	0.412325
InceptionV3	mixed6	0.6930	0.62975	0.48635	0.411025
Xception	add_10	0.6710	0.61850	0.47710	0.405100
InceptionResNetV2	mixed_7a	0.6800	0.63250	0.49500	0.417700
MobileNetV2	block_14_add	0.6605	0.60000	0.47100	0.398575
NASNetLarge	Normalizational_concat_12	0.6950	0.62950	0.49885	0.419425
EfficientNetV2L	**block6y_add**	**0.7365**	**0.67100**	**0.49980**	**0.419150**

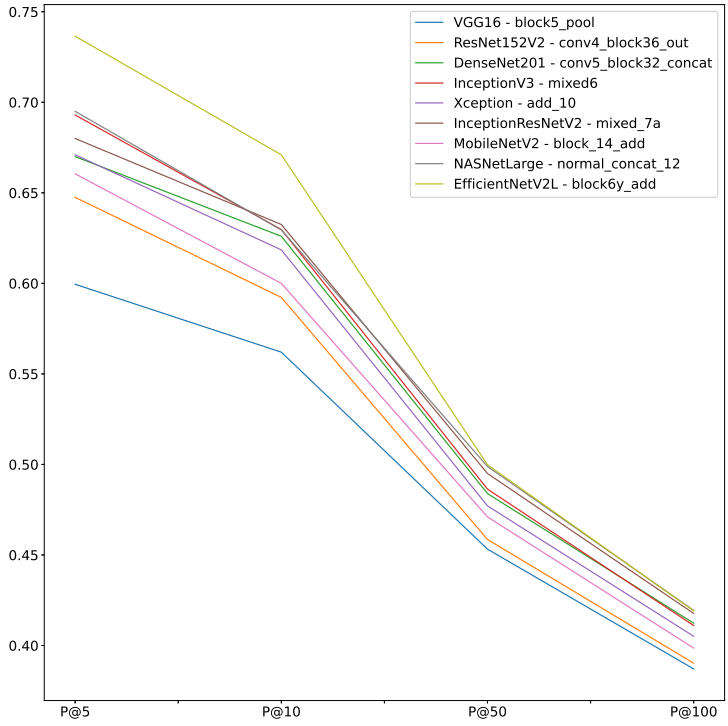


Figure 4.5. Precision comparison at 5, 10, 50, and 100 obtained from retrieval using CNN layers set according to results reported in subsection 4.2.4 and global average pooling to reduce dimensionality.

for k equal to 5 and 10, but the performance degrading for k equal to 50 and 100, especially for benign queries.

Discussion

This section provides a qualitative analysis of the worst results to understand why some queries' retrieved images are incorrect. The morphological patterns associated with breast disease at histopathology examination can be highly heterogeneous. Therefore, the diagnostic assessment considers all the morphological patterns recognized by the pathologist at the microscope during the histopathology evaluation. The pathologist reports all the characteristics observed at the histopathological examination, noting them in the report. Due to the heterogeneity mentioned above, the annotation produced to create the ground truth of a breast dataset could be oversimplified. In other words, the unsupervised and random production of patches from a WSI might occasionally provide pictures that only represent a small region and can partially represent morphological patterns other than the ground truth. This is the case, for instance, of the "benign" query in Figure 4.7a, which the model may misclassify since it can be partially superimposed on an in situ framework in the image represented by the retrieved patches. In the mentioned case, although the patch comes from a "Benign" classified case, the image refers to a borderline morphological pattern that, in our opinion, could pose a differential diagnosis issue usually ruled out by immunohistochemistry, looking for p63 expression. The same might be stated for "normal" images retrieved in response to the "in situ" query. Although belonging to TRoI annotated as "in situ," the patch seems to refer more to a typical pattern. The remaining misclassifications could also be explained by the bias of the patch's field. In the case of the "invasive" query, there could be a bias related to the type of sample, which does not seem to be a whole section but a more undersized biopsy showing a not-so-clear morphology pattern (see Figure 4.7a).

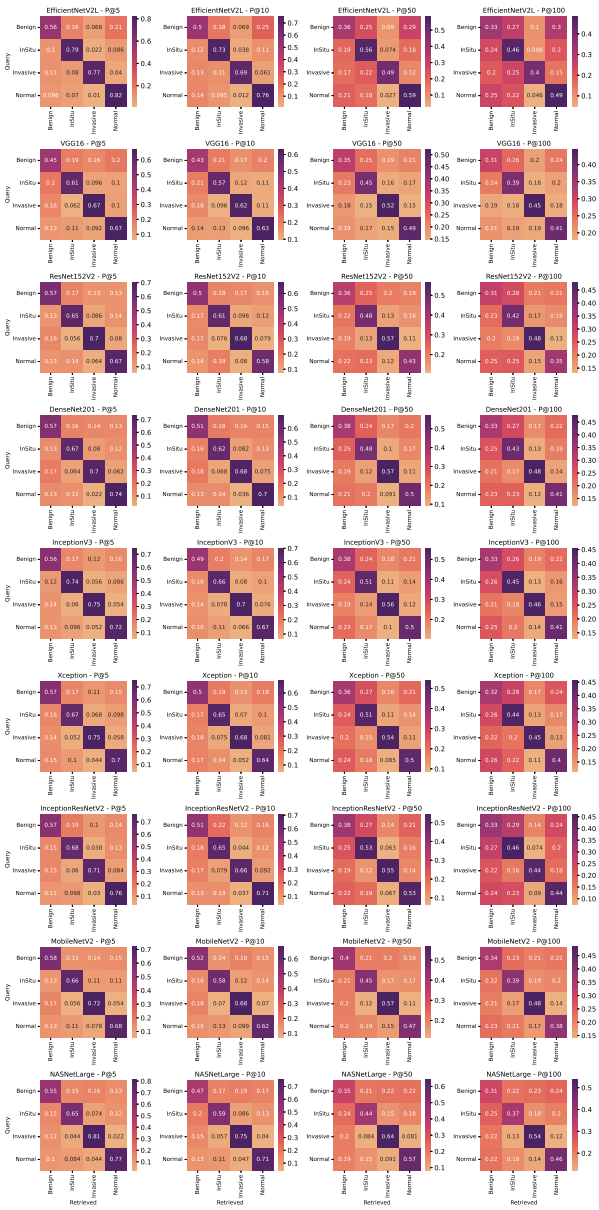
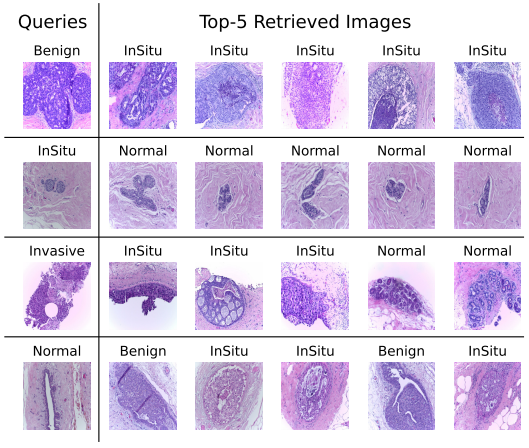
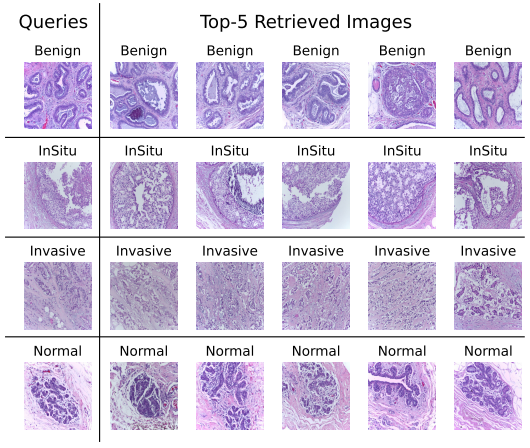


Figure 4.6. Comparison of confusion matrices obtained from retrieval using CNN layers set according to results reported in subsection 4.2.4 and global average pooling to reduce dimensionality.



(a) Examples of the wrong retrieval, in all cases, P@5 is equal to 0.



(b) Examples of the correct retrieval, in all cases, P@5 is equal to 1.

Figure 4.7. An example of totally wrong retrieval (a) and correct retrieval (b)

4.2.5 Results: Effects of color stain normalization

Here, we introduced the experimental results to estimate the effects of CSN. In the previous section, we used the average precision at K computed as the average precision of all queries [66]. The average precision in Equation 3.6 is calculated as the average precision of all queries, where $||Q||$ is the number of queries. Equation 3.5 shows the precision at K, computed as how many retrieved images on K are correct, where K is a constant limiting the number of retrieved images. We chose K equal to 5, 10, 50, and 100 in this work. Moreover, for this task, we used the BACH dataset, provided by ICIAR 2018 Grand Challenge on Breast Cancer Histology [4]. To quantify the effects of normalization on images, we used Macenko and Reinhard methods, and further, we also considered the hematoxylin channel obtained from deconvolution. Then, we compared the results with original patches having four cases: (i) the original patches, (ii) patch normalized with Macenko [53], (iii) patches normalized with Reinhard [65], and (iv) the Hematoxylin channel of the patches. In Figure 4.8, we reported an example where the row marked with *"identity"* shows the original images. Further, the figure shows one sample for each class.

Therefore, we compare the results evaluating the precision for each feature extracted as shown in Table 4.19. According to them, the results with normalized images are better than the original and Hematoxylin channel. In particular, considering the precision at five, the Macenko method achieved the best results except with features extracted by EfficientNetV2L and VGG16. Likely, precision at ten is the same as precision at 5, except for VGG16 features, where the best method is Macenko, and Mobilenet, the best, is Reinhard. Instead, the precision at 50 Reinhard method almost always achieved the best results, except for features extracted by DenseNet201 and InceptionResNetV2. The best method is Marenco. For InceptionV3, the original patches achieved the best results. Likely for precision at 100, the results are the same as precision at 50 except for features extracted from InceptionResNetV2, where the best method is Reinhard.

Finally, we evaluated the mean results, shown in Figure 4.9, as the average for each extractor and each precision. According to them, the results with k equal to five and ten, Macenko and Reinhard's methods are better than the original patches. In particular, Macenko is the best, but for k equal to 50 and 100, Reinhard is the best, and the Macenko method

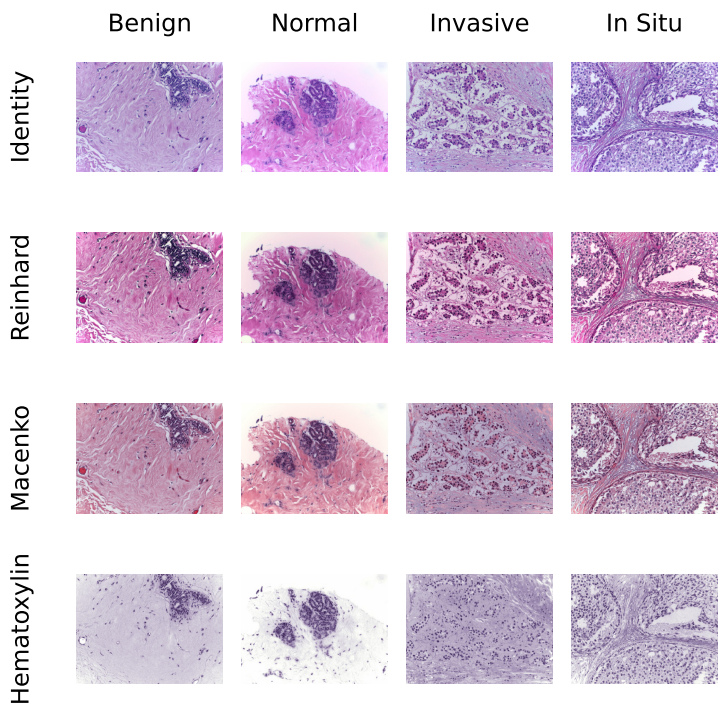


Figure 4.8. An example of one image for each class of BACH dataset using CSN

has about the same results as the original patches. Furthermore, features extracted by the Hematoxylin channel never have worse precision.

Discussion

Table 4.19 and Figure 4.9 present a comprehensive comparison of the precision of various CNN architectures when applied to different normalization techniques. The DenseNet201 architecture, when normalized with the Macenko method, consistently outperforms other normalization techniques across all P@K values. This suggests that the combination of DenseNet201 and Macenko normalization might be particularly effective for the given dataset. EfficientNetV2L and InceptionResNetV2 architectures show a

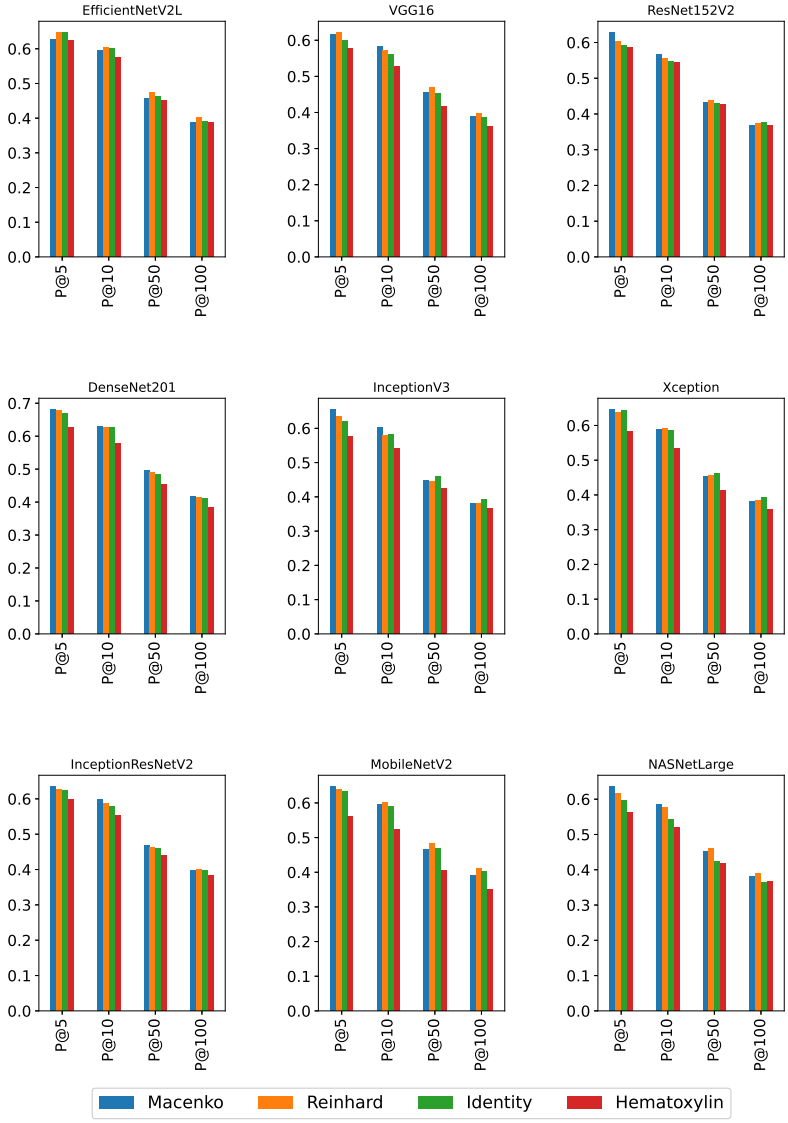


Figure 4.9. P@5, P@10, P@50, and P@100 for each features extractor and color stain normalization

Table 4.19. Comparison results between original patch normalized with Macenko, patches normalized with Reinhard, and the Hematoxylin channel used for each feature

CNN	Normalization	P@5	P@10	P@50	P@100
DenseNet201	Hematoxylin	0.6275	0.57900	0.45375	0.384500
	Identity	0.6700	0.62600	0.48390	0.412325
	Macenko	**0.6810**	**0.62850**	**0.49585**	**0.415950**
	Reinhard	0.6780	0.62675	0.48995	0.414125
EfficientNetV2L	Hematoxylin	0.6255	0.57600	0.45125	0.386875
	Identity	0.6460	0.60075	0.46350	0.390400
	Macenko	0.6280	0.59625	0.45615	0.389000
	Reinhard	**0.6470**	**0.60325**	**0.47495**	**0.403325**
InceptionResNetV2	Hematoxylin	0.5980	0.55250	0.44050	0.382350
	Identity	0.6250	0.57850	0.45850	0.396400
	Macenko	**0.6350**	**0.59875**	**0.46965**	0.398475
	Reinhard	0.6280	0.58650	0.46355	**0.400050**
InceptionV3	Hematoxylin	0.5765	0.54275	0.42465	0.366350
	Identity	0.6195	0.58350	**0.45830**	**0.392650**
	Macenko	**0.6550**	**0.60325**	0.44700	0.381075
	Reinhard	0.6340	0.57800	0.44610	0.381650
MobileNetV2	Hematoxylin	0.5600	0.52375	0.40555	0.350600
	Identity	0.6335	0.59150	0.46960	0.402500
	Macenko	**0.6475**	0.59675	0.46580	0.390450
	Reinhard	0.6390	**0.60275**	**0.48355**	**0.410700**
NASNetLarge	Hematoxylin	0.5640	0.52025	0.41890	0.366250
	Identity	0.5965	0.54200	0.42445	0.364825
	Macenko	**0.6360**	**0.58450**	0.45140	0.380000
	Reinhard	0.6170	0.57775	**0.46155**	**0.389525**
ResNet152V2	Hematoxylin	0.5870	0.54450	0.42645	0.368125
	Identity	0.5925	0.54850	0.42985	0.375250
	Macenko	**0.6280**	**0.56650**	0.43215	0.367600
	Reinhard	0.6030	0.55625	**0.43710**	**0.373525**
VGG16	Hematoxylin	0.5770	0.52800	0.41810	0.360450
	Identity	0.5995	0.56200	0.45320	0.387075
	Macenko	0.6170	**0.58225**	0.45665	0.390200
	Reinhard	**0.6215**	0.57200	**0.46810**	**0.396475**
Xception	Hematoxylin	0.5835	0.53525	0.41420	0.359325
	Identity	0.6435	0.58725	0.46195	0.393825
	Macenko	**0.6460**	0.59000	0.45285	0.380100
	Reinhard	0.6390	**0.59150**	**0.45555**	**0.385025**

mixed performance. For EfficientNetV2L, the Reinhard normalization technique seems to be the most effective, especially at higher P@K values. In contrast, for InceptionResNetV2, Macenko normalization performs best for P@5, P@10, and P@50, but Reinhard takes a slight lead at P@100. MobileNetV2 with Reinhard normalization achieves the highest precision at P@100, suggesting that this combination might be more suitable for larger retrieval tasks. For several architectures like InceptionV3, NASNet-Large, ResNet152V2, VGG16, and Xception, the Macenko normalization method either leads or is competitive with the highest precision values across different P@K metrics. The Macenko normalization method is the most effective across multiple architectures, especially at lower P@K values. This indicates its potential robustness and effectiveness in enhancing the features of the images for the CNNs. Reinhard normalization also shows competitive results, particularly at higher P@K values, suggesting its utility in larger retrieval tasks. The Identity and Hematoxylin normalization methods generally lag behind Macenko and Reinhard in most architectures, indicating that they might not be as effective for the given task. As expected, the precision values tend to decrease as K increases across all architectures and normalization methods. This is typical behavior in retrieval tasks, as retrieving more items (higher K) usually dilutes the precision. The differences in precision between normalization methods tend to be more pronounced at lower P@K values and converge as K increases. This suggests that the normalization method might be more critical for tasks requiring high precision at lower retrieval sizes.

4.3 Fast-HoVerNet: Experimental Results

This section presents the experimental strategy performed to obtain our best configuration of Fast-HoVerNet. As described in Section 3.2, we designed Fast-HoVerNet to mimic HoVerNet outputs. Therefore, its architecture is a U-Net with an RGB image as input and ten output channels, where two channels are for nuclei prediction, two channels are for horizontal and vertical maps, and six channels are for nuclei-type classification. To identify the best encoder for our model, we analyze the inference time, Multiply-Accumulate Operations (MACS), and the number of parameters of different configurations, and then we choose the encoders with less inference time and low MACS and number of parameters. Subsequently, we performed hyperparameters tuning, where our hyperparameters are α and T for each chosen encoder. Yet, we compared HoVerNet and other SOTA models on the Pannuke dataset. Lastly, we used an external dataset to compare HoVerNet and Fast-HoVerNet. We use the Adam optimizer with an initial learning rate equal to 10^{-4} and betas similar to $(0.99, 0.9999)$ to train our network. We also used a reduced on-plateau learning scheduler with a patience of 5, a scale factor of 10^{-1}, a delta of 10^{-4}, a minimum learning rate of 10^{-6}, and an early stopping with the patience of 10. Furthermore, we used a depth of encoder and decoder of 5 and initialized the encoders with ImageNet weights.

4.3.1 Inference Time Analysis

Firstly, we proved that all configurations of the student model are faster than HoVerNet. In addition, we also analyzed the inference time to demonstrate that our solution is faster than HoVerNet. Table 4.20 contrasts the U-Net and fast HoVerNet architectures, evaluating them based on computational complexity, parameter count, and inference time. The U-Net models exhibit lower MACS, indicating reduced computational complexity, and display fewer inference times than the HoVerNet. Despite HoVerNet's parameter count falling within the U-Net range, its computational complexity is higher owing to its elevated MACS value. Given these advantages of U-Net, it was selected for use with MixViT and ResNet backbone in a teacher-student setting.

Table 4.20. Comparison in terms of number of parameters, Macs, and inference time among U-Net with different encoders and HoverNet Fast version

Network	MACS	N° Params	Infer-Time(s)
U-Net (MixViT-B0)	3.01 GMac	5.55 M	0.021135
U-Net (MixViT-B1)	4.96 GMac	16.43 M	0.028674
U-Net (MixViT-B2)	6.96 GMac	27.48 M	0.050582
U-Net (MixViT-B3)	10.63 GMac	47.35 M	0.065965
U-Net (ResNet34)	7.93 GMac	24.44 M	0.030133
U-Net (ResNet50)	10.77 GMac	32.52 M	0.047324
U-Net (ResNet101)	15.64 GMac	51.51 M	0.074544
HoVerNet (Fast Version)	149.73 GMac	37.64 M	0.835522

4.3.2 Metrics

To evaluate the efficacy of our framework, we utilized the metrics as recommended in the studies of Graham et al. [27] and Gamper et al. [24]. Specifically, we employed the Panoptic Quality (PQ) metric, initially proposed by Kirillov et al. [44] and represented in Equation 4.3, in conjunction with the F-score, introduced by Graham et al. [27] and illustrated in Equation 4.4.

$$\mathscr{PQ} = \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{Detection\ Quality(DQ)} \times \underbrace{\frac{\sum_{(x,y)\in TP} IoU(x,y)}{|TP|}}_{Segementation\ Quality(SQ)} \tag{4.3}$$

$$F_c^t = \frac{2TP_c}{2TP_c + \alpha_0 FP_c + \alpha_1 FN_c + \alpha_2 FP_d + \alpha_3 FN_d} \tag{4.4}$$

The PQ is a comprehensive measure that combines the Segmentation Quality and Detection Quality, as depicted in Equation 4.3. The variable x represents the ground truth in this equation, while y denotes the predicted values. The Segmentation Quality quantifies the similarity between each detected instance and its corresponding ground truth. We

employed the PQ metric to evaluate multiclass and binary tasks, providing results for each specific nuclear type. Additionally, we utilized the F_c^t score (Equation 4.4) to assess nuclear classification for each type (t). This score is particularly useful as it accounts for over-predictions, offering a more comprehensive evaluation. It was computed using two primary sets: the paired nucleus (correctly detected) and the unpaired nucleus (over-predicted). For each type, we defined the following sets: correctly classified and detected instances (TP_c), correctly detected instances that are classified as t but actually belong to a different class (FP_c), correctly detected instances that belong to class t but are classified differently (FN_c), overpredicted instances that are classified as t but belong to a different class (FP_d), and overpredicted instances that belong to class t but are classified differently (FN_d). To assess the effectiveness of our results, we evaluated several performance metrics, including binary panoptic quality (PQ_B), multiclass panoptic quality (PQ_M), panoptic quality for specific classes ($PQ^{N,I,C,D,E}$), F-score detection (F_d), and F-score for individual classes ($F^{N,I,C,D,E}$). The classes are denoted as Neoplastic (N), Inflammatory (I), Connective/Soft Tissue (C), Dead Cells (D), and Epithelial (E). To ensure robustness, we conducted the training using a 3-fold cross-validation (3-CV) approach, following the recommended guidelines of the Pannuke authors [24]. All reported results represent the mean across the test outcomes.

4.3.3 Hyperparameters tuning and backbone selection

In order to choose the most suitable hyperparameters and backbone, we conducted an exhaustive experimental design. In particular, for each encoder, we considered α values 0, 0.5, and 1 for T values 1,3 and 5. Subsequently, we analyzed all metrics listed above. Finally, to select the best configuration, considering all metrics, we chose one that achieved the best results most times. In the following sections, we intend with α equals 0 the distillation loss, with α equals 1 the student loss, aka we did not apply KD, and with α equals 0.5 a combination of distillation and student loss equally contributed. In the following, we first report the binary results, i.e. background/foreground classification, and results averaged on each class in terms of PQ and F-score.

Average Results

First, we compared multiclass PQ and binary PQ. Table 4.21 shows the mPQ and bPQ for each encoder, T, and α value. Each result is expressed as *average* $\pm$ *standard deviation*. Analyzing these metrics, we also assert that using only HoVerNet prediction, the results are higher.

Table 4.21. Multiclass and binary PQ for each backbone and each α and T value

Encoder	α	T	mPQ	bPQ
MixViT-B0	0	1	**0.4338 ± 0.0077**	**0.6320 ± 0.0032**
		3	0.4118 ± 0.0286	0.6022 ± 0.0422
		5	0.4265 ± 0.0061	0.6286 ± 0.0028
	0.5	1	0.4136 ± 0.0075	0.6001 ± 0.0075
		3	0.4164 ± 0.0074	0.6052 ± 0.0102
		5	0.4190 ± 0.0080	0.6082 ± 0.0073
	1	-	0.3299 ± 0.0308	0.4861 ± 0.0473
MixViT-B1	0	1	**0.4388 ± 0.0093**	**0.6379 ± 0.0033**
		3	0.4326 ± 0.0133	0.6367 ± 0.0014
		5	0.4332 ± 0.0099	0.6359 ± 0.0013
	0.5	1	0.4322 ± 0.0101	0.6126 ± 0.0005
		3	0.4371 ± 0.0038	0.6218 ± 0.0020
		5	0.4407 ± 0.0073	0.6225 ± 0.0076
	1	-	0.3794 ± 0.0415	0.5417 ± 0.0541
MixViT-B2	0	1	0.4434 ± 0.0062	0.6416 ± 0.0021
		3	0.4431 ± 0.0081	**0.6434 ± 0.0028**
		5	0.4438 ± 0.0081	0.6420 ± 0.0026
	0.5	1	0.4475 ± 0.0052	0.6286 ± 0.0021
		3	**0.4482 ± 0.0066**	0.6317 ± 0.0031
		5	0.4437 ± 0.0022	0.6261 ± 0.0012
	1	-	0.3872 ± 0.0073	0.5555 ± 0.0068
MixViT-B3	**0**	1	0.4447 ± 0.0072	**0.6462 ± 0.0010**
		3	0.4452 ± 0.0029	0.6392 ± 0.0105
		5	**0.4483 ± 0.0056**	0.6437 ± 0.0018
	0.5	1	0.4412 ± 0.0022	0.6257 ± 0.0030

Continued on next page

Table 4.21. Multiclass and binary PQ for each backbone and each α and T value

Encoder	α	T	mPQ	bPQ
		3	0.4415 ± 0.0159	0.6201 ± 0.0212
		5	0.4064 ± 0.0463	0.5743 ± 0.0679
	1	-	0.3328 ± 0.0352	0.4879 ± 0.0496
		1	0.4168 ± 0.0062	0.6260 ± 0.0017
	0	3	$\mathbf{0.4226 \pm 0.0056}$	$\mathbf{0.6277 \pm 0.0034}$
		5	0.4081 ± 0.0200	0.6112 ± 0.0265
ResNet34		1	0.4000 ± 0.0091	0.5883 ± 0.0043
	0.5	3	0.4173 ± 0.0085	0.6064 ± 0.0026
		5	0.3959 ± 0.0410	0.5778 ± 0.0534
	1	-	0.3488 ± 0.0169	0.5182 ± 0.0206
		1	$\mathbf{0.4279 \pm 0.0083}$	$\mathbf{0.6270 \pm 0.0031}$
	0	3	0.4272 ± 0.0075	0.6257 ± 0.0032
		5	0.4211 ± 0.0051	0.6214 ± 0.0025
ResNet50		1	0.4126 ± 0.0043	0.5949 ± 0.0038
	0.5	3	0.4094 ± 0.0141	0.5909 ± 0.0128
		5	0.4081 ± 0.0019	0.5917 ± 0.0015
	1	-	0.3484 ± 0.0175	0.5031 ± 0.0190
		1	0.4176 ± 0.0055	0.6234 ± 0.0017
	0	3	0.4201 ± 0.0099	0.6200 ± 0.0084
		5	0.4113 ± 0.0089	0.6199 ± 0.0053
ResNet101		1	0.3731 ± 0.0439	0.5431 ± 0.0611
	0.5	3	0.3059 ± 0.0408	0.4464 ± 0.0541
		5	0.3462 ± 0.0702	0.4959 ± 0.0983
	1	-	0.3003 ± 0.0805	0.4512 ± 0.1169

Multiclass Results

To analyze the results for each class, we calculated the PQ for neoplastic, inflammatory, connective/soft tissue, dead, and epithelial. Table 4.22 shows each encoder's results and α and T values.

Table 4.22. PQ for each class, for each backbone, and each α and T value

Encoder	α	T	PQ^N	PQ^I	PQ^C	PQ^D	PQ^E
MixViT-B0	0	1	0.5284	0.3996	0.3612	0.1393	0.4323
		3	0.4913	0.3892	0.3373	0.0653	0.4230
		5	0.5299	0.3940	0.3576	0.0000	0.4287
	0.5	1	0.4701	0.3865	0.3503	0.0931	0.4361
		3	0.4894	0.3831	0.3517	0.0794	0.4273
		5	0.4918	0.3764	0.3611	0.0210	0.4429
	1	-	0.3601	0.3425	0.2713	0.0545	0.3462
MixViT-B1	0	1	0.5335	0.4025	0.3659	0.1682	0.4456
		3	0.5398	0.3912	0.3620	0.0499	0.4329
		5	0.5401	0.4034	0.3613	0.0000	0.4353
	0.5	1	0.4982	0.3983	0.3602	0.0568	0.4725
		3	0.5103	0.3886	0.3753	0.0000	0.4756
		5	0.5168	0.3924	0.3748	0.0000	0.4759
	1	-	0.4126	0.3645	0.3320	0.0081	0.3931
MixViT-B2	0	1	0.5410	0.4081	0.3671	0.1489	0.4492
		3	0.5465	0.4107	0.3679	0.1326	0.4353
		5	0.5449	0.4175	0.3716	0.0001	0.4423
	0.5	1	0.5239	0.4009	0.3794	0.0762	0.4779
		3	0.5225	0.3973	0.3848	0.0391	0.4759
		5	0.5137	0.3986	0.3802	0.0532	0.4821
	1	-	0.4621	0.3714	0.3051	0.0094	0.4040
MixViT-B3	0	1	0.5397	0.4132	0.3744	0.1426	0.4429
		3	0.5405	0.4196	0.3656	0.1486	0.4422
		5	0.5449	0.4232	0.3717	0.0976	0.4444
	0.5	1	0.5133	0.4019	0.3756	0.0947	0.4422
		3	0.5240	0.4108	0.3685	0.0000	0.4450
		5	0.4830	0.3787	0.3315	0.0014	0.4288
	1	-	0.4019	0.3286	0.2534	0.0000	0.2816
ResNet34	0	1	0.5079	0.3920	0.3494	0.1349	0.3581
		3	0.5216	0.3926	0.3559	0.0581	0.4005
		5	0.5048	0.3772	0.3425	0.0000	0.3751
	0.5	1	0.4626	0.3643	0.3396	0.0156	0.4180
		3	0.5003	0.3797	0.3512	0.0000	0.4060
		5	0.4886	0.3636	0.3214	0.0061	0.4001
	1	-	0.4204	0.3185	0.2868	0.0012	0.3271
ResNet50	0	1	0.5176	0.4037	0.3511	0.1459	0.4001
		3	0.5234	0.4047	0.3517	0.0039	0.3885

Continued on next page

Table 4.22. PQ for each class, for each backbone, and each α and T value

Encoder	α	T	PQ^N	PQ^I	PQ^C	PQ^D	PQ^E
		5	0.5176	0.3933	0.3502	0.0000	0.3723
		1	0.4879	0.3901	0.3313	0.0250	0.4200
	0.5	3	0.4830	0.3814	0.3327	0.0000	0.3956
		5	0.4828	0.3667	0.3301	0.0000	0.3837
	1	-	0.3904	0.3330	0.2828	0.0175	0.2874
		1	0.5122	0.3696	0.3603	0.1286	0.4066
	0	3	0.5132	0.4055	0.3497	0.0000	0.4016
		5	0.5199	0.3767	0.3401	0.0000	0.3777
ResNet101		1	0.4199	0.3667	0.3128	0.0358	0.4020
	0.5	3	0.3484	0.3229	0.2347	0.0000	0.2700
		5	0.3991	0.3494	0.2624	0.0000	0.3172
	1	-	0.3775	0.3413	0.2132	0.0000	0.2489

Furthermore, we computed PQ for neoplastic, inflammatory, connective/soft tissue, dead, and epithelial to analyze the behavior for each nuclei type. Table 4.23 shows results grouped by encoder, *alpha* and T.

Table 4.23. F-score for each class, for each backbone, and each α and T value

Encoder	α	T	F_d	F^N	F^I	F^C	F^D	F^E
MixViT-B0	0	1	0.7927	0.6019	0.4894	0.4429	0.1742	0.4947
		3	0.7700	0.5827	0.4950	0.4352	0.1065	0.4923
		5	0.7921	0.5987	0.4893	0.4507	0.0000	0.4990
	0.5	1	0.7801	0.6146	0.5172	0.4549	0.2105	0.5802
		3	0.7817	0.6202	0.5201	0.4590	0.1876	0.5656
		5	0.7824	0.6252	0.5206	0.4689	0.0448	0.5825
	1	-	0.7004	0.5402	0.5065	0.4048	0.1582	0.5388
MixViT-B1	0	1	0.7954	0.6066	0.4961	0.4483	0.2040	0.4989
		3	0.7947	0.6032	0.4926	0.4512	0.0587	0.4985
		5	0.7937	0.6045	0.4994	0.4512	0.0000	0.5022
	0.5	1	0.7824	0.6287	0.5309	0.4629	0.1267	0.5841
		3	0.7884	0.6347	0.5311	0.4767	0.0000	0.5873
		5	0.7885	0.6410	0.5263	0.4820	0.0000	0.6190

Continued on next page

Table 4.23. F-score for each class, for each backbone, and each α and T value

Encoder	α	T	F_d	F^N	F^I	F^C	F^D	F^E
	1	-	0.7305	0.5768	0.5081	0.4568	0.0283	0.5622
MixViT-B2	0	1	0.7969	0.6096	0.4958	0.4497	0.1812	0.4988
		3	0.7987	0.6078	0.4979	0.4536	0.1839	0.5230
		5	0.7967	0.6101	0.4992	0.4586	0.0005	0.5304
	0.5	1	0.7946	0.6441	0.5345	0.4831	0.1779	0.6211
		3	0.7933	0.6420	0.5334	0.4848	0.0996	0.6201
		5	0.7903	0.6381	0.5327	0.4829	0.1149	0.6103
	1	-	0.7488	0.6118	0.5099	0.4281	0.0142	0.5625
MixViT-B3	0	1	0.7994	0.6086	0.4953	0.4515	0.1934	0.5244
		3	0.7934	0.6086	0.5029	0.4530	0.2101	0.5294
		5	0.7972	0.6100	0.4999	0.4582	0.1591	0.5333
	0.5	1	0.7895	0.6367	0.5354	0.4778	0.1890	0.5963
		3	0.7808	0.6405	0.5320	0.4726	0.0000	0.6060
		5	0.7372	0.6150	0.4886	0.4433	0.0068	0.5907
	1	-	0.6830	0.5733	0.5088	0.3653	0.0000	0.4518
ResNet101	0	1	0.7865	0.5816	0.4519	0.4256	0.2018	0.4401
		3	0.7851	0.5850	0.4800	0.4311	0.0000	0.4688
		5	0.7881	0.5803	0.4568	0.4187	0.0000	0.4486
	0.5	1	0.7363	0.5586	0.4977	0.4182	0.1080	0.5219
		3	0.6411	0.5060	0.4857	0.3625	0.0000	0.4041
		5	0.6701	0.5472	0.4569	0.3690	0.0000	0.4332
	1	1	0.6660	0.5460	0.4814	0.2990	0.0000	0.3335
ResNet34	0	1	0.7874	0.5762	0.4573	0.4148	0.1785	0.4230
		3	0.7902	0.5833	0.4699	0.4325	0.0908	0.4714
		5	0.7774	0.5732	0.4584	0.4237	0.0000	0.4554
	0.5	1	0.7699	0.5895	0.4873	0.4368	0.0461	0.5181
		3	0.7799	0.6054	0.4975	0.4491	0.0006	0.5191
		5	0.7623	0.5986	0.4915	0.4242	0.0215	0.5190
	1	-	0.7312	0.5660	0.4796	0.4160	0.0013	0.4820
ResNet50	0	1	0.7884	0.5850	0.4730	0.4326	0.1850	0.4598
		3	0.7880	0.5872	0.4807	0.4394	0.0119	0.4680
		5	0.7835	0.5859	0.4734	0.4373	0.0000	0.4608
	0.5	1	0.7714	0.6017	0.5116	0.4373	0.0659	0.5067
		3	0.7619	0.6011	0.5105	0.4295	0.0000	0.4980
		5	0.7604	0.6031	0.5027	0.4340	0.0000	0.4972
	1	-	0.6917	0.5460	0.4752	0.4144	0.0531	0.4187

According to PQ and F-score results, we assert that the best configuration for quite all encoders is with $\alpha = 0.5$, aka when we use both ground truth and HoVerNet predictions because F-score is usually better. In our vision, the F-score is more expressive in this task because it gives a more significant weight to over and under-segmentation. Therefore, as our final version of Fast-HoVerNet, we considered the configuration with MixViT-B2 backbone, employing $\alpha = 0.5$ and $T = 1$ as the hyperparameters.

Furthermore, Figure 4.10 provides some examples where we compare ground truth, HoVerNet predictions, and Fast-HoVerNet predictions in terms of instance map (second, third, and fourth columns) and instance classification (last three columns). The analysis of these visual results shows that our version practically makes the same prediction as HoVerNet.

4.3.4 Comparison with State-of-the-art

Here, we present our best-performing results, selected as described before, in comparison to DIST [55], Mask-RCNN [30], Micro-Net [64], and Fast HoVerNet [24].

Table 4.24. Performance comparison of various models including DIST, Mask-RCNN, Micro-Net, HoVerNet, and Fast-HoVerNet (ours) across different Panoptic Quality (PQ) Metrics using Pannuke dataset

Model	PQ_B	PQ_M	PQ^N	PQ^I	PQ^C	PQ^D	PQ^E
DIST	0.5346	0.3406	0.4390	0.3430	0.2750	0.0000	0.2900
Mask-RCNN	0.5528	0.3688	0.4720	0.2900	0.3000	0.0690	0.4030
Micro-Net	0.6053	0.4059	0.5040	0.3330	0.3340	0.0510	0.4420
HoVerNet	**0.6596**	**0.4629**	**0.5510**	**0.4170**	**0.3880**	**0.1390**	**0.4910**
Ours	0.6286	0.4475	0.5239	0.4009	0.3794	0.0762	0.4779

As shown in Table 4.25, we assessed the performance of each model across a range of tissue types. The mPQ and bPQ scores for each tissue type clearly compare the models' performance in different biological contexts. Our proposed solution demonstrates competitive performance across various tissue types, exhibiting consistently high mPQ and bPQ scores. This consistency is a testament to the robustness of our solution. Table 4.24 presents the Panoptic Quality (PQ) evaluation results. Compared with HoVerNet, our solution achieved lower scores yet in line with

HoVerNet's performance and outperformed the other networks listed in the table. However, when considering the results of the $F-score$, as shown in Table 4.26, our solution demonstrates comparable performance to HoVerNet and outperforms the other networks. Furthermore, our solution offers the advantage of being approximately three times faster than HoVerNet, making it an efficient choice for this task.

We evaluated the models based on their ability to classify five different cell types: Neoplastic cells, Inflammatory cells, Connective/Soft tissue cells, Dead cells, and Epithelial cells. As depicted in Table 4.24, our solution stands out due to its consistently high performance across different cell types, demonstrating its versatility. In addition to the panoptic quality metrics, we compared the models based on their F-score metrics. As shown in Table 4.26, our proposed U-Net model exhibits competitive performance across most F-score variants, demonstrating its effectiveness. Notably, our U-Net model stands out due to its lower computational time, making it a more efficient choice for tasks where accuracy and computational efficiency are crucial. Finally, we compared the models' computational efficiency, as shown in Table 4.20. Our proposed U-Net models demonstrate competitive performance across all metrics. In particular, the U-Net model with the MixViT-B2 encoder stands out due to its balance between the number of parameters, MACs, and inference time. This balance is crucial in ensuring comprehensive and accurate cell classification without compromising computational efficiency. In conclusion, our proposed U-Net models demonstrate robustness and versatility, as evidenced by their consistently high performance across different tissue types and cell types, competitive F-scores, and computational efficiency. These findings underscore the compelling advantages of our solution for this study.

4.3.5 Results on external dataset

We undertook a comparative analysis between HoVerNet and our proposed solution, leveraging the CoNSeP dataset [27] for external evaluation purposes on out-of-domain data. Given that the classification of nuclei only showed partial correspondence of classes between the Pannuke and Consep and consequently between the trained Fast HoVerNet and Conspe targets, we remapped labels into several subclasses for a more detailed comparison. The newly defined classes were the Neoplastic, Inflammatory, Epithelial,

Table 4.25. Comparison over tissue

	DIST		Mask-RCNN		MicroNet		HoVerNet		Ours	
Tissue	**mPQ**	**bPQ**	**mPQ**	**bPQ**	**mPQ**	**bPQ**	**mPQ**	**bPQ**	**mPQ**	**bPQ**
Adrenal Gland	0.3442	0.5603	0.3470	0.5546	0.4153	0.6440	0.4812	0.6962	0.4596	0.6706
Bile Duct	0.3614	0.5384	0.3536	0.5567	0.4124	0.6232	0.4714	0.6696	0.4422	0.6363
Bladder	0.4463	0.5625	0.5065	0.6049	0.5357	0.6488	0.5792	**0.7031**	0.5629	0.6752
Breast	0.3790	0.5466	0.3882	0.5574	0.4407	0.6029	0.4902	0.6470	0.4780	0.6296
Cervix	0.3371	0.5309	0.3402	0.5483	0.3795	0.6101	0.4438	0.6652	0.4554	0.6468
Colon	0.2989	0.4508	0.3122	0.4603	0.3414	0.4972	0.4095	0.5575	0.3838	0.5278
Esophagus	0.3942	0.5295	0.4311	0.5691	0.4668	0.6011	0.5085	0.6427	0.5040	0.6285
Head & Neck	0.3177	0.4764	0.3946	0.5457	0.3668	0.5242	0.4530	0.6331	0.4430	0.5999
Kidney	0.3339	0.5727	0.3553	0.5092	0.4165	0.6321	0.4424	0.6836	0.4807	0.6528
Liver	0.3441	0.5818	0.4103	0.6085	0.4365	0.6666	0.4974	0.7248	0.4600	0.6796
Lung	0.2809	0.4978	0.3182	0.5134	0.3370	0.5588	0.4004	0.6302	0.3561	0.5592
Ovarian	0.3789	0.5289	0.4337	0.5784	0.4387	0.6013	0.4863	0.6309	0.4988	0.6224
Pancreatic	0.3395	0.5343	0.3624	0.5460	0.4041	0.6074	0.4600	0.6491	0.4330	0.6216
Prostate	0.3810	0.5442	0.3959	0.5789	0.4341	0.6049	0.5101	0.6615	0.4823	0.6215
Skin	0.2627	0.5080	0.2665	0.5021	0.3223	0.5817	0.3429	0.6234	0.3347	0.5945
Stomach	0.3369	0.5553	0.3684	0.5976	0.3872	0.6293	0.4726	0.6886	0.4312	0.6552
Testis	0.3278	0.5548	0.3512	0.5420	0.4088	0.6300	0.4754	0.6890	0.4477	0.6555
Thyroid	0.2574	0.5596	0.3037	0.5712	0.3712	0.6555	0.4315	0.6983	0.4222	0.6691
Uterus	0.3487	0.5246	0.3683	0.5589	0.3965	0.5821	0.4393	0.6393	0.4265	0.5978
Average across tissues	0.3406	0.5346	0.3688	0.5528	0.4059	0.6053	0.4629	0.6596	0.4475	0.6286
STD across splits	0.0156	0.0097	0.0046	0.0076	0.0082	0.0050	0.0076	0.0036	0.0052	0.0021

Table 4.26. Comparative evaluation of various models Including DIST, Mask-RCNN, Micro-Net, HoVerNet, and Fast HoVerNet (ours) across different F-Score Metrics using Pannuke dataset

Model	F_d	F^N	F^I	F^C	F^D	F^E
DIST	0.73	0.50	0.42	0.39	0.00	0.35
Mask-RCNN	0.72	0.59	0.50	0.42	0.22	0.52
Micro-Net	0.80	0.62	0.52	0.47	0.19	0.58
HoVerNet	**0.80**	0.62	**0.54**	**0.49**	**0.31**	0.56
Ours	0.79	**0.64**	0.53	0.48	0.18	**0.62**

Table 4.27. Multiclass results on CoNSeP dataset

Model	F_d	F^N	F^I	F^E	F^O	PQ	Inf. Time(s)
HoVerNet	**0.818**	0.526	**0.758**	0.495	**0.559**	0.415	~ 48
Ours	0.742	**0.594**	0.681	**0.603**	0.557	**0.599**	**~ 17**

and Miscellaneous. We mapped the Neoplastic class with *Pannuke's neoplastic* and *CoNSeP's dysplastic/malignant epithelial*, the Inflammatory class with *Pannuke's inflammatory* and *CoNSeP's inflammatory*, the Epithelial class with *Pannuke's epithelial* and *CoNSeP's healthy epithelial*, and the Miscellaneous class with *Pannuke's dead and connective tissues* as well as *CoNSeP's other types, which include fibroblast, muscle, and endothelial tissues*. The comparison metrics included the F-score for detection, the F-score for each class, and mPQ (mean Panoptic Quality). The results of this comparison are presented in Table 4.27. The results demonstrate that our solution surpasses HoVerNet in terms of panoptic quality, though it falls short in terms of F-score detection. Regarding classification metrics, our solution outperforms HoVerNet across all nuclei types except the inflammatory ones.

Furthermore, we provide visual examples in Figure 4.11, comparing the CoNSeP ground truth, HoVerNet, and our predictions. The qualitative similarity between the results further confirms the effectiveness of our approach.

4.3.6 Discussion

According to the results analyzed in this section, the use KD to obtain a fast version of HoVerNet had promising results. We observed that our network achieved results close to the teacher network. Despite our results on some nuclei types being worse than HoVerNet, the visual results are not quite distinguishable, as shown in Figure 4.10. Furthermore, we also showed that our network works well on CoNSeP, which is an external dataset. However, we proposed a version faster than the original one because, from experimental results, we observed that it is three times faster. This result is useful in real-world applications because, on a WSI, the inference time is reduced from about six hours to two hours, while this difference is not evident on a single TRoI. Furthermore, the training time is no higher than HoVerNet because our network has fewer parameters and fewer MACs, and the distillation affects one time in the inference of HoVerNet because we used an offline approach, and in the loss faction where we compute the distillation loss. Finally, we report an inference example on WSI, Figure 4.12, where we zoomed some portion of it to show the results of nuclei instance segmentation and classification.

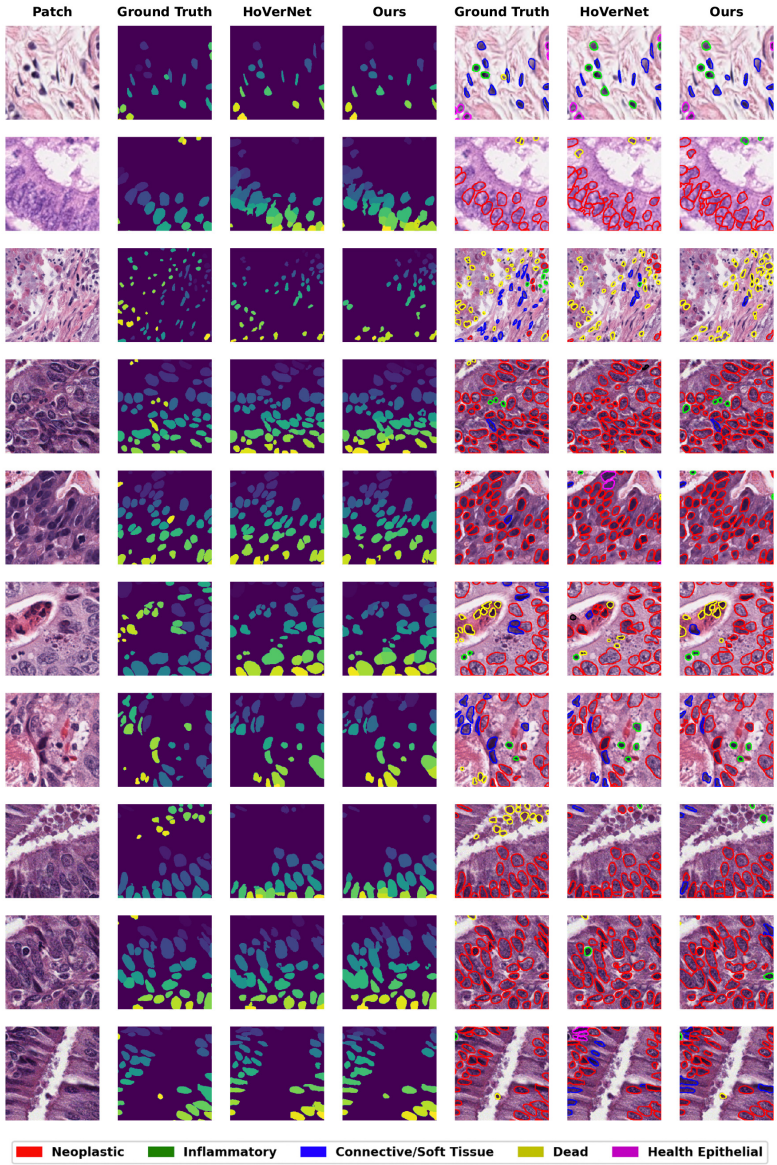


Figure 4.10. Comparison between HoVerNet and Fast-HoVerNet (Ours). From left to right: patch, ground truth, HoVerNet, and Fast-HoVerNet instance maps, ground truth, HoVerNet, Fast-HoVerNet nuclei instance classifications

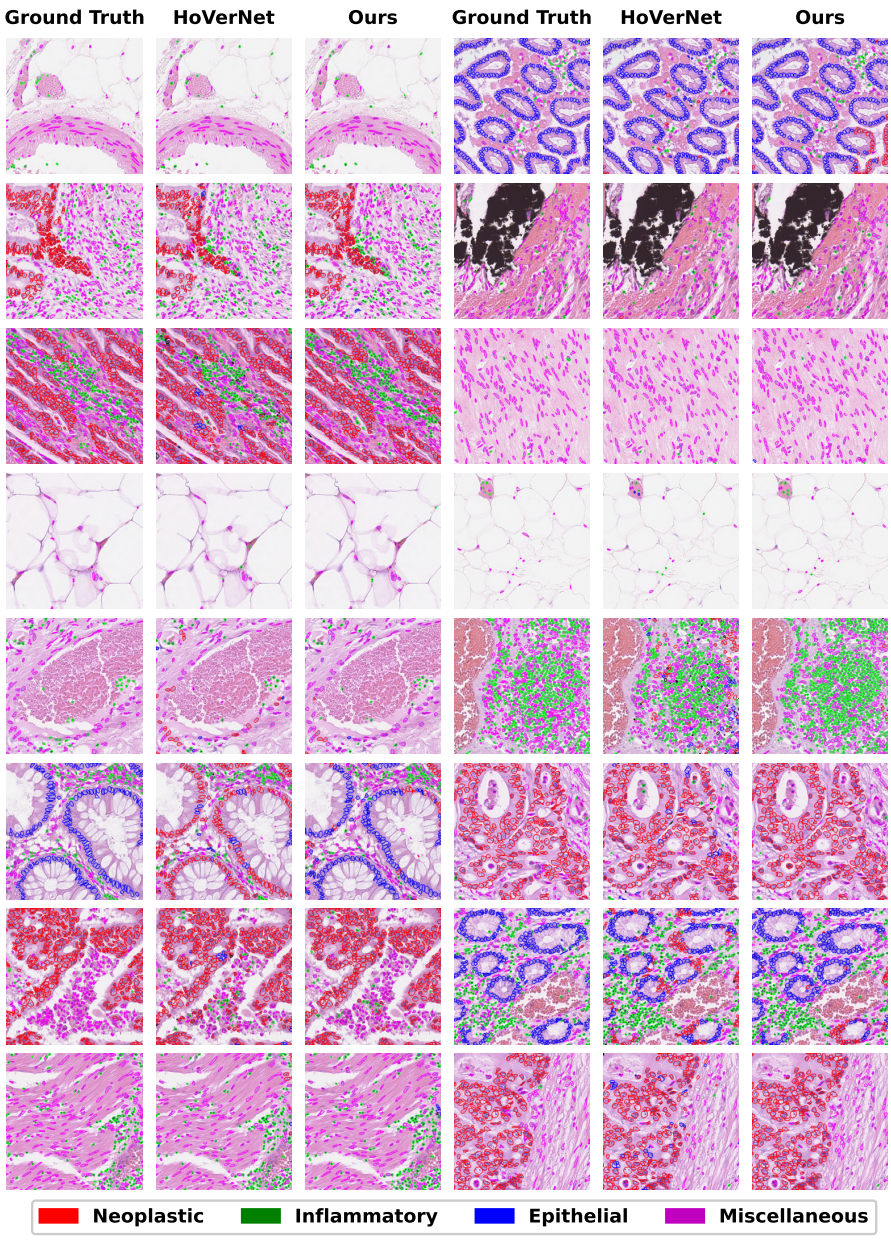


Figure 4.11. Nuclei segmentation and classification comparison between CoNSeP ground truth, HoVerNet, and our predictions.

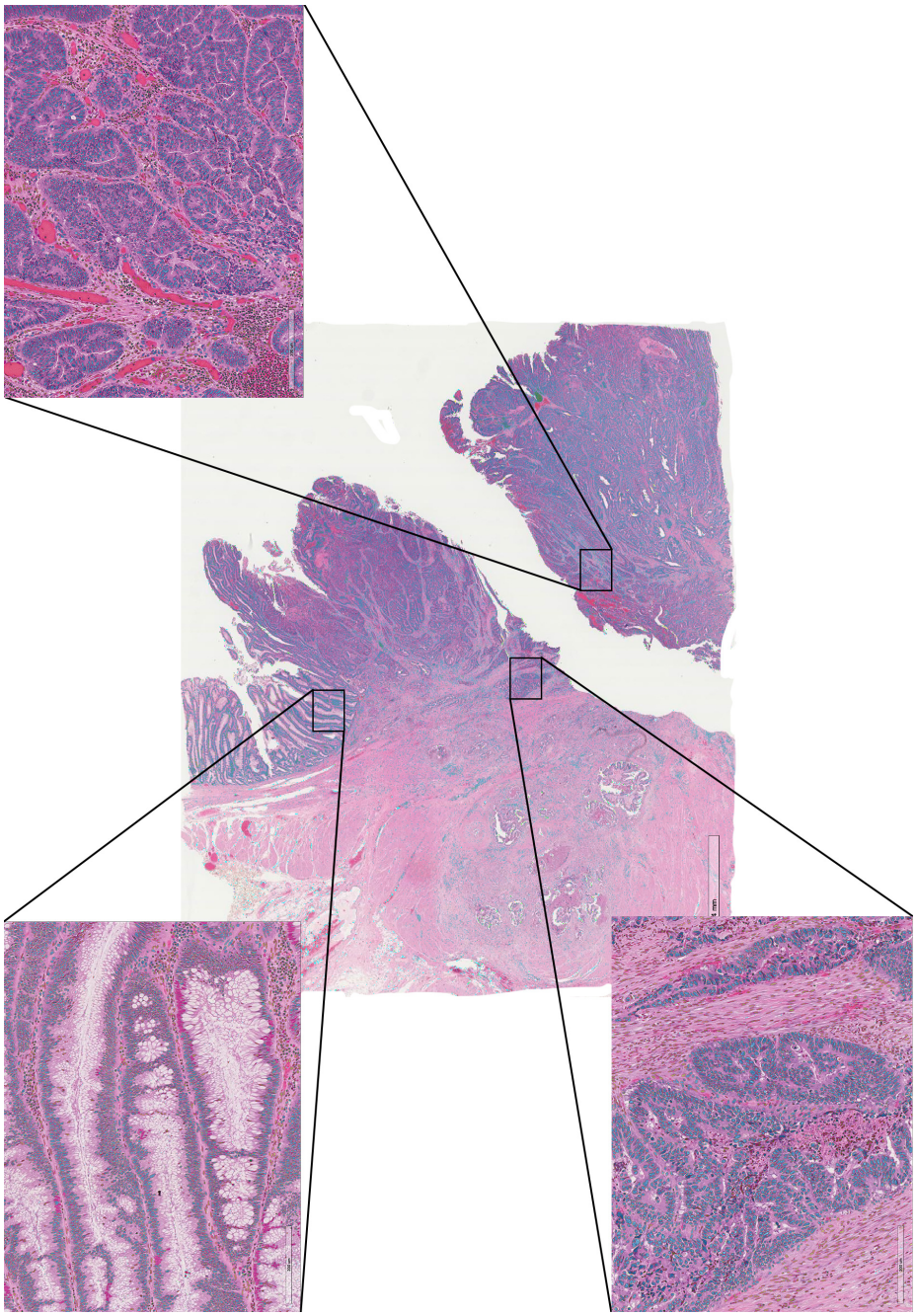


Figure 4.12. An example of Fast-HoVerNet inference of a WSI

4.4 Cell Graph Classification: Experimental Results

This section presents the results concerning the CG classification. In particular, we show our experimental strategy to choose the best way to represent the CG according to the description given in Section 3.3.2. Practically, we use a data-centric AI approach, where we choose a model and work on the dataset in order to find the best way to represent it. Afterwards, we provided a comparison with a SOTA.

According to architecture introduced in Section 3.3.3, we configured each PNA layers with 1 tower and mean, max, min, std aggregators, and identity, amplification, attenuation scalers, and 64 hidden channels. We used 3 PNA layers with batch normalization and ReLU after each one. Lastly, we applied LSTM JK before the readout operation. On the head, we used an MLP with two layers, hidden channels 64, dropout 0.5, batch normalization, and ReLU as activation function.

To train our network, we used Adam optimizer with an initial learning rate 10^{-3} end weight decay $5 \cdot 10^{-4}$, and we used to reduce on the plateau as learning scheduler with patience of 5 epochs.

Finally, we used BRACS dataset for tuning and comparison with SOTA. Figure 4.13 depicts one example for each BRACS class, while Figure 4.14 a more detailed example for one TRoI of BRACS.

4.4.1 Cell-Graph representation tuning

According to the parameters reported in Section 3.3.2, we choose the number of neighbors (k) 5 and minimum distance (min_{dist}) 50 pixels, as also reported in [60]. Then, we performed some experiments to identify the best way to represent the cell feature. We tried different configurations, but we report only the most promising ones here. In particular, we used statistics, mean, max, and sum according to the definition in Equation 3.13. So we have the following configuration:

- A: [centroids, mean], where we computed the mean only where the nucleus is in the crop.
- B: [centroids, sum], where we normalized the sum on the max sum value over all nuclei.

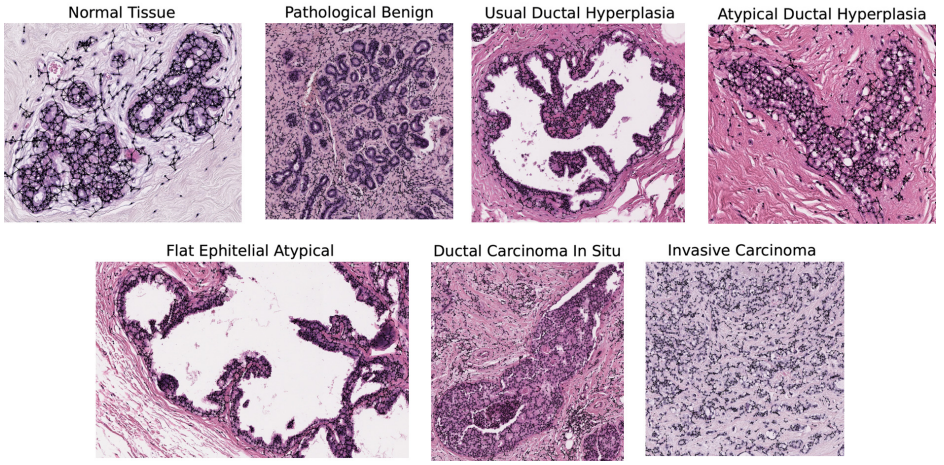


Figure 4.13. Cell-Graph over all BRACS classes

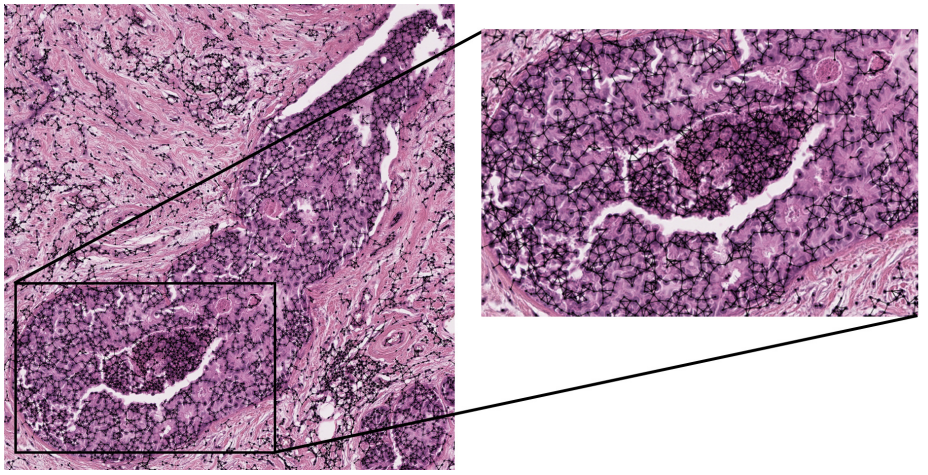


Figure 4.14. A detailed example of a Cell-Graph on a BRACS TRoI

- C: [centroids, max].

According to Table 4.28, we select the best feature representation configuration A.

Therefore, our final CG has each node represented as a feature with shape 1×18.

Table 4.28. F1 measure for A, B, and C configurations.

Config.	Weighted F1	Normal	PB	UDH	ADH	FEA	DCIS	Invasive
A	**58.65 ± 0.83**	**57.82 ± 3.64**	47.4 ± 1.65	44.54 ± 3.32	**40.28 ± 4.89**	**63.58 ± 2.98**	**67.05 ± 0.96**	83.04 ± 0.46
B	57.69 ± 0.96	56.33 ± 5.05	**49.24 ± 1.53**	45.67 ± 4.39	30.18 ± 4.56	62.75 ± 2.56	66.45 ± 2.59	**85.86 ± 0.64**
C	54.5 ± 1.56	55.8 ± 6.67	47.75 ± 2.63	**47.09 ± 1.76**	24.08 ± 6.47	55.86 ± 3.61	62.32 ± 1.73	82.31 ± 3.53

4.4.2 Comparison with state-of-the-art

Here, we compare with SOTA on BRACS dataset. All approaches to which we make a comparison are proposed [60] and are CG-GNN that use CG, TG-GNN that uses TG, CONCAT-GNN that uses a concatenation graph between CG and TG and HACT-Net that use hierarchical representation. In order to make a fair comparison, we consider CG-GNN as the baseline. However, we also compare our network with their other approaches, HACT-Net especially, which is the best one.

We did two settings, first over seven classes of BRACS dataset and second over four classes aggregation of the classes on risk cancer as suggested in [60]. For the second setting, the classes are normal, non-cancerous (PB and UDH), pre-cancerous (ADH and FEA), and cancerous (DCIS, invasive) classes. Furthermore, we make the comparison on four classes without retraining the network but using the network obtained on seven classes and grouping them following the setting described above.

Seven classes

Table 4.29 shows the F1 for each class and weighted F1. As mentioned above, we mainly compared our network with CG-GNN. Therefore, we argue that our results are better than the baseline in terms of weighted F1. Analyzing each class, we observe that our result is lower than the baseline for normal, but our standard deviation is lower. Instead, we outperform the baseline over PB, DCIS, and Invasive. While for ADH, our F1 is slightly over baseline, but our std is higher. Instead, for FEA, the F1 measures are practically equal, but our std is lower. Lastly, for normal, our F1 is lower than baseline, but our std is lower. Taking into account TG-GNN, Concat-GNN, and HACT-Net, our F1 measures are better for DCIS and are worse for normal, invasive, and FEA. While F1-measures are similar for UDH, PB, ADH. Figure 4.15, which shows the comparison over seven classes, and a vertical line on the top of each bar is the standard

deviation. Considering weighted F1, our network achieved the second-best results after HACT-Net.

Table 4.29. F1 comparison over seven classes with SOTA

Model	Normal	PB	UDH	ADH	FEA	DCIS	Invasive	Weighted F1
CG-GNN	58.77 ± 6.82	40.87 ± 3.05	46.82 ± 1.95	39.99 ± 3.56	63.75 ± 10.48	53.81 ± 3.89	81.06 ± 3.33	55.94 ± 1.01
TG-GNN	63.59 ± 4.88	47.73 ± 2.87	39.41 ± 4.70	28.51 ± 4.29	72.15 ± 1.35	54.57 ± 2.23	82.21 ± 3.99	56.62 ± 1.35
CONCAT-GNN	60.97 ± 4.54	43.06 ± 2.26	41.96 ± 4.67	26.10 ± 3.73	71.29 ± 2.09	60.83 ± 3.71	85.42 ± 2.70	57.01 ± 2.27
HACT-Net	61.56 ± 2.15	47.49 ± 2.94	43.60 ± 1.86	40.42 ± 2.55	74.22 ± 1.41	66.44 ± 2.57	88.40 ± 0.19	61.53 ± 0.87
Ours	57.82 ± 3.64	47.4 ± 1.65	44.54 ± 3.32	40.28 ± 4.89	63.58 ± 2.98	67.05 ± 0.96	83.04 ± 0.46	58.65 ± 0.83

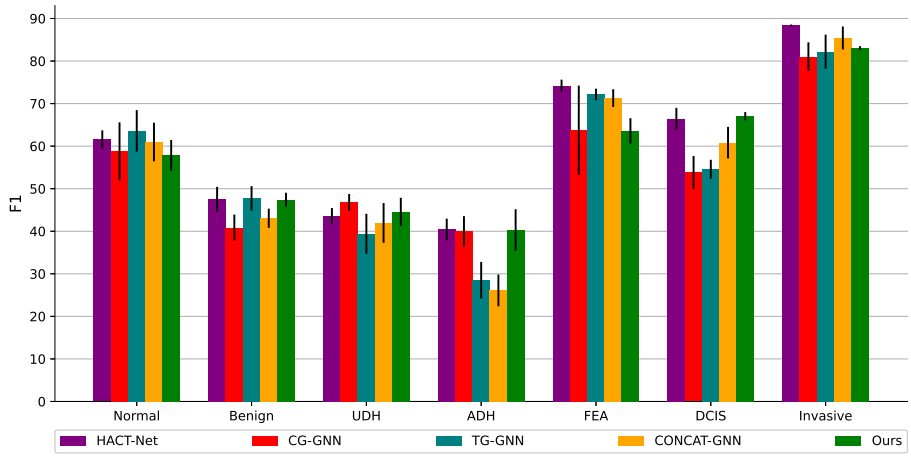


Figure 4.15. Comparison F1 and standard deviation seven classes

Lastly, we reported a normalized confusion matrix in Figure 4.17a.

Four classes

Table 4.30 shows the F1 comparison over four classes, i.e., normal, non-cancerous, precancerous, and cancerous, and the weighted F1. Analyzing these results, according to weighted F1, we can argue that the baseline slightly outperforms ours because its standard deviation is higher than ours. Considering each class, our results are better for normal, comparable for precancerous, and worse for non-cancerous and cancerous. However, considering the SOTA, our results are worse but comparable for non-cancerous. Furthermore, we reported a bar plot in Figure 4.16 to compare all models. The line on top of each bar is the deviation standard.

Table 4.30. F1 comparison over four classes with SOTA

Model	Normal	Non-cancerous	Precancerous	Cancerous	Weighted F1
CG-GNN	52.95 ± 12.11	56.55 ± 3.70	61.53 ± 3.03	84.47 ± 0.87	66.10 ± 2.58
TG-GNN	52.96 ± 6.81	56.52 ± 2.85	64.36 ± 1.05	82.21 ± 0.78	66.24 ± 1.11
CONCAT-GNN	54.54 ± 1.64	56.63 ± 1.68	62.58 ± 1.45	81.80 ± 0.77	65.83 ± 0.04
HACT-Net	66.08 ± 3.69	55.28 ± 1.74	66.21 ± 0.87	84.91 ± 0.79	69.04 ± 0.46
Ours	57.82 ± 3.64	52.17 ± 1.38	61.28 ± 1.24	82.33 ± 0.44	64.87 ± 0.61

Lastly, we reported a normalized confusion matrix in Figure 4.17b.

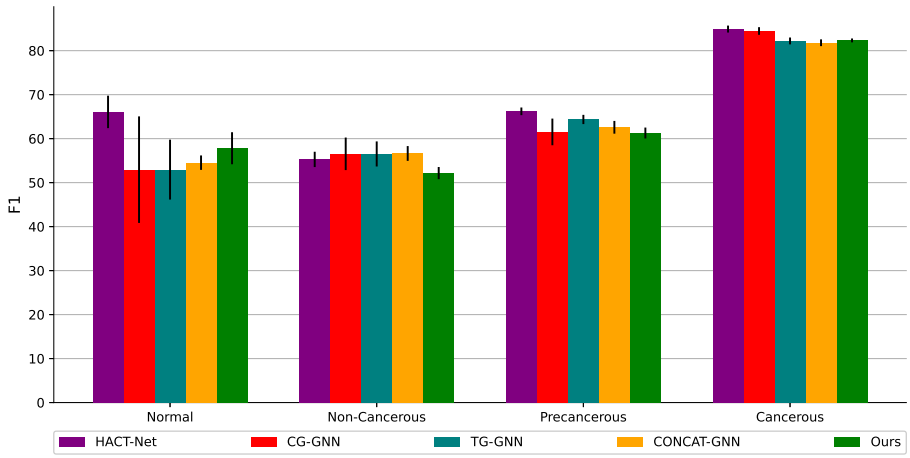
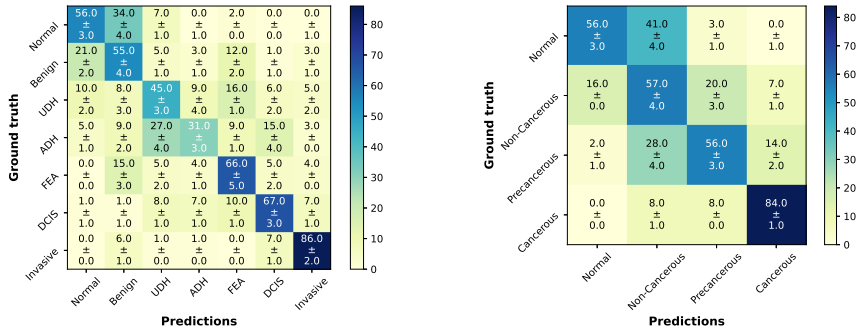


Figure 4.16. Comparison F1 and standard deviation four classes

4.4.3 Discussion

According to the results in this section, we can conclude that our method to extract nuclei features requires more investigation because the results are promising, obtaining comparable and, in some cases, better of SOTA that use CG representation. However, our technique does not outperform HACT-Net, which represents the SOTA, but this model uses a hierarchical representation also considering tissue features. Analyzing the results for seven classes, we can affirm that our version has better results DCIS, outperforming HACT-Net instead for classes PB and UDH, and ADH our results are really close to it, this further proves that our



(a) Normalized confusion matrix for seven classes

(b) Normalized confusion matrix for four classes

Figure 4.17. Confusion matrics

approach needs feature developments with a hierarchical representation of WSI even because HACT-Net results highlighted that hierarchical representation is useful to achieve higher results on FEA and also improve F1 on normal and invasive. Analyzing the confusion matrix over seven classes in Figure 4.17a, we observe that our model makes mistakes between normal and benign TRoI and further, it is not able to distinguish ADH class, confounding it mainly with UDH. Instead, for UDH, the models correctly classify enough TRoI but make some mistakes, confounding mainly them with normal and FEA. Instead, it classifies invasive TRoI in a good way. For FEA TRoI, it can classify them but commits some mistakes with benign. About DCIS, it makes a few errors, mistaking it with FEA. While analyzing the confusion matrix over for classes in Figure 4.17b, we can conclude that results on cancerous TRoI are high, while on normal non-cancerous and precancerous, our network commits many mistakes on class with close cancer risk. Despite not outperforming HACT-Net, our approach is faster for CG building because we used Fast-HoVerNet, which is faster than HoVerNet. Furthermore, we unified the process using the same network to extract cell features. Again, our CG graph is smaller than HACT because we represent each cell with a feature 1×18 while it has a shape 1×514 because it uses ResNet as a features extractor.

Chapter 5

Conclusions

In this Ph.D. work, we implemented some CPATH tasks, shown in detail in 3. These tasks are included in a CPATH general framework, as also shown in 3. In particular, we first proposed a TRoI CBIR system providing the empirical results on the best feature extract, in terms of a layer of pre-trained CNN, and estimation of the effects of CSN. Then, we proposed Fast-HoVerNet, a faster version of HoVerNet. Lastly, we proposed a novel approach to extract nuclei-cell features for CG representation. Our method achieved good results in CG classification, comparable with SOTA. As supported by experimental results in 4, we achieved our goal on each task, designing and developing modules that, in some cases, outperform that SOTA results. In other cases, we are really close best results. However, we achieved our goal in terms of time complexity because our solutions are designed to reduce the inference time of existing tools.

Here, we answer to Research Questions (RQs) formulated in Chapter 1.

- **RQ1**: Are pre-trained CNNs suitable as a feature extractor in TRoIs CBIR systems?
  The experimental results, shown in Section 4.2.4, suggest that pre-trained CNN can be used as a feature extractor, and we also studied and identified the best layer of them that can used as a feature extractor.

- **RQ2**: What are the effects of CSN in a TRoIs CBIR systems?
  The results, shown in Section 4.2.5, suggest that CSN improves the

performances in CBIR. In fact, the Macenko method improves the results in our CBIR system.

- **RQ3**: How can we obtain a CNN for Nuclei Instance Segmentation and Classification faster than SOTA ones?
  In this task, we proposed Fast-HoVerNet, a fast version of HoVerNet, a SOTA network, obtained using KD. Results, shown in Section 4.3, prove that it achieved results similar to HoVerNet in terms of F-score and PQ, but it is three times faster. We also demonstrate the effectiveness of the external dataset.

- **RQ4**: How can we obtain a unified framework for CG building?
  We proposed a novel approach using Fast-HoVerNet as a feature extractor, detailed in Section 3.3.2. This method allows us to obtain a really small feature compared with techniques used in literature, i.e. our feature has shape 1×18, and one extracted using ResNet50 has shape 1×514; both features include centroids.

- **RQ5**:Is our CG representation suitable for breast cancer subtype classification?
  We conducted several experiments, as shown in Section 4.4, and our representation obtained promising results.

In future work, we will point to developing more accurate techniques of retrieval that also work on graph structure. Furthermore, we will explore other techniques to achieve better results on nuclei instance segmentation and classification and further reduce the inference time to make this application useful in real applications. Due to promising results obtained on CG classification, in future works, we will investigate the hierarchical representation using CG representation proposed in this Ph.D. thesis. In this way, we will compare our results fairly with SOTA. Lastly, we will explore techniques of XAI to provide tools that can help pathologists in cancer diagnosis.

Bibliography

[1] Esther Abels, Liron Pantanowitz, Famke Aeffner, Mark D Zarella, Jeroen van der Laak, Marilyn M Bui, Venkata NP Vemuri, Anil V Parwani, Jeff Gibbs, Emmanuel Agosto-Arroyo, et al. Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the digital pathology association. *The Journal of pathology*, 249(3):286–294, 2019.

[2] Mohammed Adnan, Shivam Kalra, and Hamid R Tizhoosh. Representation learning of histopathology images using graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 988–989, 2020.

[3] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. A survey on graph-based deep learning for computational histopathology. *Computerized Medical Imaging and Graphics*, 95:102027, 2022.

[4] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, et al. Bach: Grand challenge on breast cancer histology images. *Medical image analysis*, 56:122–139, 2019.

[5] Peter Bankhead, Maurice B Loughrey, José A Fernández, Yvonne Dombrowski, Darragh G McArt, Philip D Dunne, Stephen McQuaid, Ronan T Gray, Liam J Murray, Helen G Coleman, et al. Qupath: Open source software for digital pathology image analysis. *Scientific reports*, 7(1):1–7, 2017.

[6] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, et al. Bracs: A dataset for breast carcinoma subtyping in h&e histology images. *Database*, 2022:baac093, 2022.

[7] Juan C Caicedo, Fabio A Gonzalez, and Eduardo Romero. A semantic content-based retrieval method for histopathology images. In *Asia Information Retrieval Symposium*, pages 51–60. Springer, 2008.

[8] Juan C Caicedo, Jorge A Vanegas, Fabian Páez, and Fabio A González. Histology image search using multimodal fusion. *Journal of Biomedical Informatics*, 51:114–128, 2014.

[9] Ana Caramelo, António Polónia, João Vale, Mónica Curado, Sofia Campelos, Vanessa Nascimento, Mariana Barros, Diana Ferreira, Tânia Pereira, Beatriz Neves, et al. Demonstrating the interference of tissue processing in the evaluation of tissue biomarkers: the case of pd-l1. *Pathology-Research and Practice*, page 154605, 2023.

[10] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7:1–11, 2006.

[11] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. Dcan: deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2487–2496, 2016.

[12] Zewei Chen, Fengwei Zhou, George Trimponias, and Zhenguo Li. Multi-objective neural architecture search via non-stationary policy gradient. *arXiv preprint arXiv:2001.08437*, 2020.

[13] François Chollet. Xception: deep learning with depthwise separable convolutions. corr abs/1610.02357 (2016). *arXiv preprint arXiv:1610.02357*, 2016.

[14] Emily L Clarke and Darren Treanor. Colour in digital pathology: a review. *Histopathology*, 70(2):153–163, 2017.

[15] David A Clunie. Dicom format and protocol standardization—a core requirement for digital pathology success. *Toxicologic Pathology*, 49(4):738–749, 2021.

[16] Toby C Cornish, Ryan E Swapp, and Keith J Kaplan. Whole-slide imaging: routine pathologic diagnosis. *Advances in anatomic pathology*, 19(3):152–159, 2012.

[17] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33:13260–13271, 2020.

[18] Miao Cui and David Y Zhang. Artificial intelligence and computational pathology. *Laboratory Investigation*, 101(4):412–422, 2021.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Navid Farahani, Anil V Parwani, and Liron Pantanowitz. Whole slide imaging in pathology: advantages, limitations, and emerging perspectives. *Pathology and Laboratory Medicine International*, pages 23–33, 2015.

[21] Ada T Feldman and Delia Wolfe. Tissue processing and hematoxylin and eosin staining. *Histopathology: methods and protocols*, pages 31–43, 2014.

[22] Renato Ferreira, Bongki Moon, Jim Humphries, Alan Sussman, Joel Saltz, Robert Miller, and Angelo Demarzo. The virtual microscope. In *Proceedings of the AMIA Annual Fall Symposium*, page 449. American Medical Informatics Association, 1997.

[23] Karl Francis and Bernhard O Palsson. Effective intercellular communication distances are determined by the relative time constants for cyto/chemokine secretion and diffusion. *Proceedings of the National Academy of Sciences*, 94(23):12258–12262, 1997.

[24] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benes, Simon Graham, Mostafa Jahanifar, Syed Ali Khurram, Ayesha Azam, Katherine Hewitt, and Nasir Rajpoot. Pannuke dataset extension, insights and baselines. *arXiv preprint arXiv:2003.10778*, 2020.

[25] KC Gatter, A Heryet, C Alcock, and DY Mason. Clinical importance of analysing malignant tumours of uncertain origin with immunohistological techniques. *The Lancet*, 325(8441):1302–1305, 1985.

[26] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[27] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019.

[28] Tatyana S Gurina and Lary Simms. Histology, staining. In *StatPearls [Internet]*. StatPearls Publishing, 2022.

[29] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

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

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

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[33] Narayan Hegde, Jason D Hipp, Yun Liu, Michael Emmert-Buck, Emily Reif, Daniel Smilkov, Michael Terry, Carrie J Cai, Mahul B Amin, Craig H Mermel, et al. Similar image search for histopathology: Smily. *NPJ digital medicine*, 2(1):1–9, 2019.

[34] Markus D Herrmann, David A Clunie, Andriy Fedorov, Sean W Doyle, Steven Pieper, Veronica Klepeis, Long P Le, George L Mutter, David S Milstone, Thomas J Schultz, et al. Implementing the dicom standard for digital pathology. *Journal of pathology informatics*, 9(1):37, 2018.

[35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[36] Jason D Hipp, Anna Fernandez, Carolyn C Compton, and Ulysses J Balis. Why a pathology image should not be considered as a radiology image. *Journal of pathology informatics*, 2, 2011.

[37] Mahdi S Hosseini, Babak Ehteshami Bejnordi, Vincent Quoc-Huy Trinh, Danial Hasan, Xingwen Li, Taehyo Kim, Haochen Zhang, Theodore Wu, Kajanan Chinniah, Sina Maghsoudlou, et al. Computational pathology: A survey review and the way forward. *arXiv preprint arXiv:2304.05482*, 2023.

[38] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[39] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[40] Guillaume Jaume, Pushpak Pati, Antonio Foncubierta-Rodriguez, Florinda Feroce, Giosue Scognamiglio, Anna Maria Anniciello, Jean-Philippe Thiran,

Orcun Goksel, and Maria Gabrani. Towards explainable graph representations in digital pathology. *arXiv preprint arXiv:2007.00311*, 2020.

[41] Oscar Jimenez-del Toro, Sebastian Otálora, Manfredo Atzori, and Henning Müller. Deep multimodal case–based retrieval for large histopathology datasets. In *International Workshop on Patch-based Techniques in Medical Imaging*, pages 149–157. Springer, 2017.

[42] Philipp Kainz, Martin Urschler, Samuel Schulter, Paul Wohlhart, and Vincent Lepetit. You should use regression to detect cells. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 276–283. Springer, 2015.

[43] Shivam Kalra, Hamid R Tizhoosh, Charles Choi, Sultaan Shah, Phedias Diamandis, Clinton JV Campbell, and Liron Pantanowitz. Yottixel–an image search engine for large archives of histopathology whole slide images. *Medical Image Analysis*, 65:101757, 2020.

[44] A Kirillov, K He, R Girshick, C Rother, and P Dollar. Panoptic segmentation. in 2019 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9396–9405, 2018.

[45] Navid Alemi Koohbanani, Mostafa Jahanifar, Neda Zamani Tajadin, and Nasir Rajpoot. Nuclick: a deep learning framework for interactive segmentation of microscopic images. *Medical Image Analysis*, 65:101771, 2020.

[46] Meghana Dinesh Kumar, Morteza Babaie, and Hamid R Tizhoosh. Deep barcodes for fast retrieval of histopathology scans. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[47] Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging*, 36(7):1550–1560, 2017.

[48] Haydee Lara, Zaibo Li, Esther Abels, Famke Aeffner, Marilyn M Bui, Ehab A ElGabry, Cleopatra Kozlowski, Michael C Montalto, Anil V Parwani, Mark D Zarella, et al. Quantitative image analysis for tissue biomarker use: a white paper from the digital pathology association. *Applied Immunohistochemistry & Molecular Morphology*, 29(7):479–493, 2021.

[49] Zaibo Li, Marilyn M Bui, and Liron Pantanowitz. Clinical tissue biomarker digital image analysis: A review of current applications. *Human Pathology Reports*, 28:300633, 2022.

[50] Jianfang Liu, Tara Lichtenberg, Katherine A Hoadley, Laila M Poisson, Alexander J Lazar, Andrew D Cherniack, Albert J Kovatich, Christopher C Benz, Douglas A Levine, Adrian V Lee, et al. An integrated tcga pan-cancer clinical data resource to drive high-quality survival outcome analytics. *Cell*, 173(2):400–416, 2018.

[51] David N Louis, Georg K Gerber, Jason M Baron, Lyn Bry, Anand S Dighe, Gad Getz, John M Higgins, Frank C Kuo, William J Lane, James S Michaelson, et al. Computational pathology: an emerging definition. *Archives of pathology & laboratory medicine*, 138(9):1133–1138, 2014.

[52] Wenqi Lu, Simon Graham, Mohsin Bilal, Nasir Rajpoot, and Fayyaz Minhas. Capturing cellular topology in multi-gigapixel pathology images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 260–261, 2020.

[53] Marc Macenko, Marc Niethammer, James S Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE international symposium on biomedical imaging: from nano to macro*, pages 1107–1110. IEEE, 2009.

[54] Michael C Montalto. Pathology re-imagined: the history of digital radiology and the future of anatomic pathology. *Archives of pathology & laboratory medicine*, 132(5):764–765, 2008.

[55] Peter Naylor, Marick Laé, Fabien Reyal, and Thomas Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE transactions on medical imaging*, 38(2):448–459, 2018.

[56] Royal College of Pathologists. Digital pathology, 2023.

[57] Liron Pantanowitz, Ashish Sharma, Alexis B Carter, Tahsin Kurc, Alan Sussman, and Joel Saltz. Twenty years of digital pathology: an overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives. *Journal of pathology informatics*, 9(1):40, 2018.

[58] Liron Pantanowitz, Paul N Valenstein, Andrew J Evans, Keith J Kaplan, John D Pfeifer, David C Wilbur, Laura C Collins, and Terence J Colgan. Review of the current state of whole slide imaging in pathology. *Journal of pathology informatics*, 2(1):36, 2011.

[59] Ankush Patel, Ulysses GJ Balis, Jerome Cheng, Zaibo Li, Giovanni Lujan, David S McClintock, Liron Pantanowitz, and Anil Parwani. Contemporary whole slide imaging devices and their applications within the modern pathology department: A selected hardware review. *Journal of Pathology Informatics*, 12(1):50, 2021.

[60] Pushpak Pati, Guillaume Jaume, Antonio Foncubierta-Rodriguez, Florinda Feroce, Anna Maria Anniciello, Giosue Scognamiglio, Nadia Brancati, Maryse Fiche, Estelle Dubruc, Daniel Riccio, et al. Hierarchical graph representations in digital pathology. *Medical image analysis*, 75:102264, 2022.

[61] Emily S Patterson, Mike Rayo, Carolina Gill, and Metin N Gurcan. Barriers and facilitators to adoption of soft copy interpretation from the user perspective: Lessons learned from filmless radiology for slideless pathology. *Journal of pathology informatics*, 2(1):1, 2011.

[62] Xin Qi, Daihou Wang, Ivan Rodero, Javier Diaz-Montes, Rebekah H Gensure, Fuyong Xing, Hua Zhong, Lauri Goodell, Manish Parashar, David J Foran, et al. Content-based histopathology image retrieval using cometcloud. *BMC bioinformatics*, 15(1):1–17, 2014.

[63] Ashwin Raju, Jiawen Yao, Mohammad MinHazul Haq, Jitendra Jonnagaddala, and Junzhou Huang. Graph attention multi-instance learning for accurate colorectal cancer staging. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*, pages 529–539. Springer, 2020.

[64] Shan E Ahmed Raza, Linda Cheung, Muhammad Shaban, Simon Graham, David Epstein, Stella Pelengaris, Michael Khan, and Nasir M Rajpoot. Micro-net: A unified model for segmentation of various objects in microscopy images. *Medical image analysis*, 52:160–173, 2019.

[65] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[66] Yates Ricardo Baeza and Neto Berthier Ribeiro. Modern information retrieval, 2011.

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

[68] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[69] Roger Schaer, Sebastian Otálora, Oscar Jimenez-del Toro, Manfredo Atzori, and Henning Müller. Deep learning-based retrieval system for gigapixel histopathology cases and the open access literature. *Journal of pathology informatics*, 10, 2019.

[70] Jun Shi, Ruoyu Wang, Yushan Zheng, Zhiguo Jiang, and Lanlan Yu. Graph convolutional networks for cervical cell classification. In *MICCAI 2019 Computational Pathology Workshop COMPAY*, 2019.

[71] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.

[72] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[73] Linda Studer, Jannis Wallau, Heather Dawson, Inti Zlobec, and Andreas Fischer. Classification of intestinal gland cell-graphs using graph neural networks. In *2020 25th International conference on pattern recognition (ICPR)*, pages 3636–3643. IEEE, 2021.

[74] Mookund Sureka, Abhijeet Patil, Deepak Anand, and Amit Sethi. Visualization for histopathology images using graph convolutional neural networks. In *2020 IEEE 20th international conference on bioinformatics and bioengineering (BIBE)*, pages 331–335. IEEE, 2020.

[75] C Szegedy, V Vanhoucke, S Ioffe, J Shlens, and Z Wojna. Rethinking theinception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2015.

[76] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[77] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[78] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[79] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021.

[80] Clive Roy Taylor and Richard J Cote. Immunomicroscopy: a diagnostic tool for the surgical pathologist. *(No Title)*, 1986.

[81] Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Melissa Schlitter, Irene Esposito, and Nassir Navab. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE transactions on medical imaging*, 35(8):1962–1971, 2016.

[82] Jan G Van den Tweel and Clive R Taylor. A brief history of pathology: preface to a forthcoming series that highlights milestones in the evolution of pathology as a discipline. *Virchows Archiv*, 457:3–10, 2010.

[83] Quoc Dang Vu, Simon Graham, Tahsin Kurc, Minh Nguyen Nhat To, Muhammad Shaban, Talha Qaiser, Navid Alemi Koohbanani, Syed Ali Khurram, Jayashree Kalpathy-Cramer, Tianhao Zhao, et al. Methods for segmentation and classification of digital microscopy tissue images. *Frontiers in bioengineering and biotechnology*, page 53, 2019.

[84] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[85] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. Mobiledets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3825–3834, 2021.

[86] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[87] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

[88] Pengshuai Yang, Yupeng Zhai, Lin Li, Hairong Lv, Jigang Wang, Chengzhan Zhu, and Rui Jiang. A deep metric learning approach for histopathological image retrieval. *Methods*, 179:14–25, 2020.

[89] Mehran Yazdi and Hamed Erfankhah. Multiclass histology image retrieval, classification using riesz transform and local binary pattern features. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 8(6):595–607, 2020.

[90] Mo Zhang, Bin Dong, and Quanzheng Li. Ms-gwnn: multi-scale graph wavelet neural network for breast cancer diagnosis. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2022.

[91] Xiaofan Zhang, Wei Liu, Murat Dundar, Sunil Badve, and Shaoting Zhang. Towards large-scale histopathological image analysis: Hashing-based image retrieval. *IEEE Transactions on Medical Imaging*, 34(2):496–506, 2014.

[92] Yushan Zheng, Zhiguo Jiang, Yibing Ma, Haopeng Zhang, Fengying Xie, Huaqiang Shi, and Yu Zhao. Content-based histopathological image retrieval for whole slide image database using binary codes. In *Medical Imaging 2017: Digital Pathology*, volume 10140, pages 266–271. SPIE, 2017.

[93] Yanning Zhou, Simon Graham, Navid Alemi Koohbanani, Muhammad Shaban, Pheng-Ann Heng, and Nasir Rajpoot. Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[94] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition.(2017). *arXiv preprint arXiv:1707.07012*, 2(6), 2017.

Author's publications

1. C. Tommasino, F. Merolla, C. Russo, S. Staibano, A.M. Rinaldi. Histopathological Image Deep Feature Representation for CBIR in Smart PACS, Journal of Digital Imaging, 36(5), 2194-2209, 2023. DOI: 10.1007/s10278-023-00832-x.

2. K. Madani, A.M. Rinaldi, C. Russo, C. Tommasino. A combined approach for improving humanoid robots autonomous cognitive capabilities, Knowledge and Information Systems, 65(8), 3197-3221, 2023. DOI: 10.1007/s10115-023-01844-3.

3. G. Renzi, A.M. Rinaldi, C. Russo, C. Tommasino. A storytelling framework based on multimedia knowledge graph using linked open data and deep neural networks, Multimedia Tools and Applications, 82(20), 31625-31639, 2023. DOI: 10.1007/s11042-023-14398-x.

4. A.M. Rinaldi, C. Russo, C. Tommasino. Automatic image captioning combining natural language processing and deep neural networks, Results in Engineering, 18, 2023. DOI: 10.1016/j.rineng.2023.101107.

5. A. Bosco, S. Capuozzo, B. Celano, M. Gravina, S. Marrone, M.P. Maurelli, V. Moscato, G. Pontillo, M. Postiglione, A.M. Rinaldi, L. Rinaldi, C. Russo, G. Sperlì, C. Tommasino, G. Cringoli, C. Sansone. AI in healthcare: Activities of the University of Naples Federico II node of the CINI-AIIS Lab, CEUR Workshop Proceedings, 3486, 112-117, 2023.

6. M. Montanaro, A.M. Rinaldi, C. Russo, C. Tommasino. A rule-based obfuscating focused crawler in the audio retrieval domain, Multimedia Tools and Applications, 2023. DOI: 10.1007/s11042-023-16155-6.

7. A.M. Rinaldi, C. Russo, C. Tommasino. An Augmented Reality CBIR System Based on Multimedia Knowledge Graph and Deep Learning Tech-

niques in Cultural Heritage, Computers, 11(12), 2022. DOI: 10.3390/computers11120172.

8. A.M. Rinaldi, C. Russo, C. Tommasino. A Novel Approach to Populate Multimedia Knowledge Graph via Deep Learning and Semantic Analysis, ACM International Conference Proceeding Series, 40-47, 2022. DOI: 10.1145/3508397.3564846.

9. M. Muscetti, A.M. Rinaldi, C. Russo, C. Tommasino. Multimedia ontology population through semantic analysis and hierarchical deep features extraction techniques, Knowledge and Information Systems, 64(5), 1283-1303, 2022. DOI: 10.1007/s10115-022-01669-6.

10. A.M. Rinaldi, C. Russo, C. Tommasino. Effects of Color Stain Normalization in Histopathology Image Retrieval using Deep Learning, Proceedings - 2022 IEEE International Symposium on Multimedia, ISM 2022, 26-33, 2022. DOI: 10.1109/ISM55400.2022.00010.

11. A.M. Rinaldi, C. Russo, C. Tommasino. An Approach Based on Linked Open Data and Augmented Reality for Cultural Heritage Content-Based Information Retrieval, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13376 LNCS, 99-112, 2022. DOI: 10.1007/978-3-031-10450-3_8.

12. A.M. Rinaldi, C. Russo, C. Tommasino. A semantic approach for document classification using deep neural networks and multimedia knowledge graph, Expert Systems with Applications, 169, 2021. DOI: 10.1016/j.eswa.2020.114320.

13. A.M. Rinaldi, C. Russo, C. Tommasino. Web Document Categorization Using Knowledge Graph and Semantic Textual Topic Detection, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12951 LNCS, 40-51, 2021. DOI: 10.1007/978-3-030-86970-0_4.

14. A.M. Rinaldi, C. Russo, C. Tommasino. Visual Query Posing in Multimedia Web Document Retrieval, Proceedings - 2021 IEEE 15th International Conference on Semantic Computing, ICSC 2021, 415-420, 2021. DOI: 10.1109/ICSC50631.2021.00086.

15. A.M. Rinaldi, C. Tommasino, C. Russo. A knowledge-driven multimedia retrieval system based on semantics and deep features, Future Internet, 12(11), 1-20, 2020. DOI: 10.3390/fi12110183.

16. Michele, A., Pasquale, A., Davide, A., Francesco, B., Ugo, B., Francesco, C., Andrea, C., Francesco, E., Antonio, G., Maurizio, M., Andrea, S., Stefano, S., Cristian, T. A project of fitness telemonitoring in an information

technology course, 2017 IEEE International Workshop on Measurement and Networking, M and N 2017 - Proceedings, 2017. DOI: 10.1109/I-WMN.2017.8078358.