FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Quality Control in Digital Pathology: Improving Fragment Detection and Counting

Maria José Valente da Silva Carneiro

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Dr. Jaime dos Santos Cardoso

Co-Supervisors: Dr. Tomé Mendes Albuquerque, Dr. Diana Leitão Montezuma Pego Felizardo

July 22, 2024

© Maria Carneiro, 2024

Quality Control in Digital Pathology: Improving Fragment Detection and Counting

Maria José Valente da Silva Carneiro

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Dr. Rui Carlos Camacho de Sousa Ferreira da Silva
Referee: Prof. Dr. Inês Prata Machado

July 22, 2024

Resumo

Os processos de patologia digital devem garantir que a qualidade das amostras examinadas é mantida durante todo o processo para prever um diagnóstico exato, que é crucial para muitos casos de oncologia. O controlo de qualidade é, portanto, fundamental para garantir que não se perde material importante durante a preparação bruta das lâminas. Os técnicos do laboratório de patologia têm de verificar o número de fragmentos presentes manualmente nas lâminas com o número de fragmentos descritos nos relatórios macroscópicos. Este processo moroso e trabalhoso interrompe a cadeia digital, causando atrasos na obtenção do diagnóstico completo pronto a ser analisado pelos patologistas. Para ultrapassar este problema, investigadores desenvolveram um sistema autónomo capaz de detetar o número de fragmentos e conjuntos em cada lâmina através de métodos convencionais de aprendizagem autónoma e de redes convolucionais profundas. Apesar disso, muitos problemas continuam a dificultar o desempenho do algoritmo, como as diferenças de tamanho dos tecidos e fragmentos desconectados. Esta dissertação baseia-se na pesquisa anterior, explorando os modelos previamente desenvolvidos e propondo correcções e melhorias. Apresentamos os resultados para as tarefas de deteção e contagem separadamente. Utilizamos o YOLOv5 e o YOLOv9 para detetar fragmentos e conjuntos. Para a contagem, não só derivamos a contagem de fragmentos por conjunto e dos conjuntos a partir das detecções dadas pelos modelos de deteção, como também prevemos contagens utilizando classificadores padrão baseados em diferentes arquiteturas. Os resultados de todas as experiências provam que tanto a deteção como a contagem são melhores quando se utilizam as detecções do YOLOv9. Analisamos extensivamente os resultados de contagem do YOLOv9 através de validação cruzada e entre os domínios das características das amostras, utilizando um conjunto de dados alargado composto por 2053 imagens de treino, 499 de validação e 701 de teste de vários órgãos, natureza e tipos de técnicas de coloração. Além disso, exploramos a classificação de grafos para a contagem de fragmentos por conjunto como uma nova abordagem para este caso. Os resultados experimentais não são competitivos em comparação com os outros métodos propostos, mas continuam a ser relevantes como uma análise exploratória do problema e para aprofundar a compreensão das relações intrínsecas entre fragmentos e conjuntos.

Abstract

Digital pathology processes must ensure that the quality of the examined specimens is maintained during the whole pipeline to predict the most accurate diagnosis, which is crucial for many oncology cases. Quality control is then critical to ensure no loss of valuable material during the gross preparation of slides. Pathology lab technicians must manually cross-check the number of fragments per set present on slides with the number of fragments per set described in macroscopic reports. This time-consuming and labor-intensive process breaks the digital pipeline, causing delays in obtaining the complete diagnosis ready for pathologists to review. Researchers have developed an autonomous system that can detect the fragments and sets on each slide through conventional machine learning and deep convolutional network methods to surmount this. Despite that, many issues still hinder the algorithm's performance, like differences in tissue sizes and disconnected fragments. This dissertation builds upon previous work by exploring the previously developed models and proposing new methods that improve detection and counting. We present the results for the detection and counting tasks separately. We use YOLOv5 and YOLOv9 to detect fragments and sets. For counting, we not only derive fragments per set and set counts from the detections given by the detection models but also predict counts using standard classifiers with different backbones. The results from all experiments prove that both detection and counting are improved using YOLOv9 detections. We extensively analyze the YOLOv9 counting results through cross-validation and across sample domains, using an extended dataset comprised of 2053 train, 499 validation, and 701 test histopathology images of various organs, specimens, and staining technique types. Furthermore, we explore graph classification for counting fragments per set as a novel approach for this case. Experimental results are not competitive compared with the other proposed methods but remain relevant as an exploratory analysis of the problem and to deepen the understanding of the intrinsic relationships between fragments and sets.

Acknowledgements

I want to express my deepest gratitude to my supervisors, Jaime Cardoso, Tomé Albuquerque, and Diana Felizardo, who relentlessly offered me much-needed guidance and support throughout the development of this thesis. Their unwavering commitment, extensive expertise, and especially their passion for this field have inspired me beyond measure.

The collaboration with the IMP Diagnostics Laboratory was crucial to the development of this work, as they provided the carefully annotated data that this research is based upon. I extend my sincere appreciation to them and, once again, to Dr. Diana Felizardo for her clinical supervision throughout the project. I also want to thank everyone in the pathology research group at INESCTEC for the opportunity to engage with their research and for all the helpful insights.

I am profoundly grateful to my parents, Fátima and Manuel, for all the love and support they so selflessly give me. Your relentless confidence in my abilities constantly inspires me to strive for greatness. For all the patience and comfort, my most sincere thank you. This also goes out to my vó Lina, whose kindness and generosity know no match, and to my sister Carolina, who is my rock.

To all my lifelong friends, thank you. All the joy, motivation, and inspiration you give me is immeasurable. It is such a privilege to grow alongside such wonderful people; my happiness is yours. A special appreciation goes to Juliana for understanding me even when I don't understand myself and to Filipa for being home far away from home for so many years; I will always cherish the memories in our little place.

To Sofia, Tuna, João, André, Sérgio, and Edgar, my most profound gratitude for making five years feel like the most joyful second. I would not have achieved this without your incredible support and friendship; you are the sole reason my academic experience was truly wonderful. All our memories will always have a special place in my heart.

Maria José Carneiro

"One's life has value so long as one attributes value to the life of others."

Simone de Beauvoir

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
CAD	Computer-Aided Diagnosis
CCA	Connected Component Analysis
CML	Conventional Machine Learning
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DL	Deep Learning
FCN	Fully Connected Network
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
ML	Machine Learning
MLP	Multi-Layer Perceptron
OOF	Out-Of-Focus
OOD	Out-Of-Distribution
RoI	Region of Interest
RPN	Region Proposal Network
ReLU	Rectified Linear Unit
WSI	Whole-Slide Image

Chapter 1

Introduction

Digital Pathology (DP) implements the study and diagnosis of disease in a digitized environment through systems and tools that transform physical pathology slides into digital images along with their corresponding meta-data, enabling their analysis, review, and storage [1].

The digital counterparts of the glass slides are whole slide images (WSIs) that, enriched with experts' annotations and labels, mutated digital pathology into computer-aided diagnosis (CAD) [87], typically using Artificial Intelligence (AI) tools that aid professionals in routine practice decisions, meaningfully impacting patient care [14, 4]. However, the widespread use of these systems still raises concerns among specialists: is the quality of diagnosis from an AI system comparable to a human? Do the images used for digital diagnosis embed sufficient information that is as valuable as their physical counterparts? Most errors in the laboratory testing pipeline occur during the pre-analytical testing phase before even considering the pathology slides for interpretation [50], so focusing on guaranteeing that slides accurately represent information is critical.

In this context, it is paramount to guarantee that the quality of digital slides is consistent with their physical representation and that the information embedded in them is enough for an accurate diagnosis while also maintaining reliability when presented with out-of-scope or erroneous samples [23]. For this, pathology lab technicians routinely perform quality control checkpoints throughout the pathology pipeline, namely cross-checking the number of fragments present on macroscopic lab reports with the number of fragments on slides after mounting and scanning. To expedite this process, Albuquerque et al. [9] proposed automatically doing this quality control process by performing the analysis with the help of AI tools that alert technicians when discrepancies are found. In Figure 1.1, we describe their proposed pipeline for automated detection in comparison with the traditional manually intensive method: on A, the fragments are scanned and manually compared to the macroscopic report; on B, the slides are scanned and posteriorly assessed by an automated system.

Thus, the motivation for this work comes from the need for more research on developing intelligent systems that automatically perform fragment detection and counting reliably so that pathology clinicians can benefit from a less labor-intensive but equally trustworthy process and

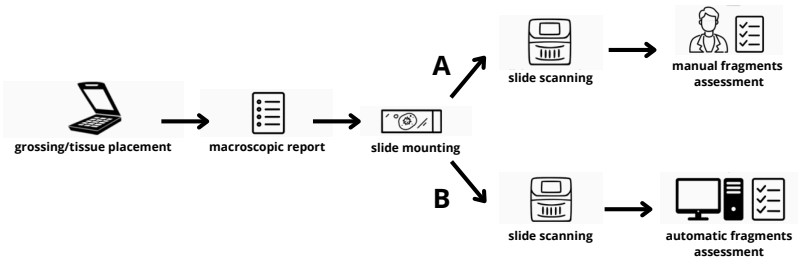

Figure 1.1: Pipeline for the number of fragments assessment checkpoint. Recreated from [9].

patients can benefit from expedited diagnosis, which is crucial in time-sensitive cases.

The counting that pathology technicians verify in the cross-checking process is the number of fragments present in each set, so this is the objective task to optimize for our research, along with counting sets, as this is also a significant value to collect for pathologists. Nevertheless, detecting fragments and sets is crucial for this assessment, as it is essential that the system that provides this count also visually presents the results for aided confirmation.

We empirically evaluate detection models and counting methods that solve these tasks. First, we analyze the results obtained by the previous work on detecting fragments and sets by Albuquerque et al. [9] to examine the critical difficulties when detecting structures in these images. Since we have access to an extended dataset of a total of 2053 train, 499 validation, and 701 test images, we finetune the YOLOv5 [117] detection model to verify how the model performs with access to more data. We also finetune the state-of-the-art detector, YOLOv9 [122], to analyze its detection performance for both fragments and sets and its average across both classes. Considering the counting tasks, we study two different approaches: to derive counts from the detections inferred from the detection models and to predict counts by training simple classifiers with varying backbone models, such as ResNet [52], ViT [36], and DINOv2 [89]. To test whether the best-performing model generalizes well across domains, we analyze this method by specimen, organ, and technique type. We further examine its behavior across each type when some domains are withheld from training. We also perform cross-validation across all counting methods to consolidate results.

We further perform an exploratory study of graph-based approaches as novel methods for counting fragments per set in samples, motivated by the strength of these methods in capturing structural and hierarchical relationships between nodes. For this, we reframe the problem as a graph classification and transform our dataset accordingly. First, we detect fragments using a fragment detection method and crop around the detected fragments centroids. Afterward, we extract features from these crops and use them to define our graph nodes, which are connected by edges with weight determined by the distance between these centroids. The ground truth labels are then assigned to each image graph, and a graph classification model predicts the number of fragments per set present in each image. During training, we use bounding box annotations as our fragments to crop. Still, during evaluation, we crop according to the detections inferred from YOLOv9 or with a fixed resolution around fragments defined by Connected Component Analysis

(CCA). We also extract features from the fragment crops by handcrafting representative features or extracting them through a contrastive-based feature extractor model. The different versions of the model are evaluated using the same metrics as the other counting methods.

Considering our approach, our hypothesis is that improving automated detection and counting methods for this quality control process can effectively decrease the time spent by pathology technicians on this task. Since we obtained good results for both detection and counting, we are confident that clinically integrating these methods would positively affect pathology clinicians' routines. Therefore, the main contribution of our work is providing strong counting and detection methods while analyzing and exploring approaches to continuously improve and increase the reliability of results as further research into future clinical integration.

The impact of this work is aligned with the promotion of prosperity, peace, and sustainable progress promoted by the United Nations. The 17 Sustainable Development Goals (SGD) [88], presented in Figure 1.2, encourage the eradication of poverty and other inequalities while handling global issues like climate change by fostering improvements in education, health, and the environment, through economic and social growth. This work is integrated within the following goals[1]:

> **Goal 3: Good Health and Well-being** ensures healthy lives and promotes well-being for all. This work enhances diagnostic efficiency in pathology through AI, leading to faster disease detection and treatment, thereby improving patient outcomes and overall health.
>
> **Goal 8: Decent Work and Economic Growth** promotes sustained economic growth and decent work for all. Automating pathology tasks reduces manual labor, increases productivity, and allows technicians to focus on more complex tasks. This enhances job satisfaction and supports economic growth through technological innovation.
>
> **Goal 9: Industry, Innovation, and Infrastructure** aims to build resilient infrastructure and foster innovation. By integrating AI in digital pathology, this work modernizes diagnostic processes, driving innovation in healthcare infrastructure and contributing to a more advanced medical industry.

Figure 1.2: The 17 Sustainable Development Goals [88].

[1]The content of this publication has not been approved by the United Nations and does not reflect the views of the United Nations or its officials or Member States.

The document is organized as follows. Chapter 2 defines the background concepts necessary for understanding this work and presents an overview of the existing literature and state-of-the-art methods related to quality control in digital pathology, detection and counting, and its application in the digital pathology field. Chapter 3 details the detection and counting problem, the methodology proposed to improve it, the metrics and protocols used for evaluation, and the comprehensive analysis of the results obtained, including cross-validation and domain generalization. Chapter 4 introduces the exploratory work around graph classification as a counting approach, the methodology proposed, the metrics and protocols used for evaluation, and the review and discussion of the results obtained. Finally, Chapter 5 establishes conclusions from the present research and proposes directions of improvement for further work.

Chapter 2

Literature Review

This chapter introduces the central notions of quality control in digital pathology and detection and counting approaches. We review the existing literature to establish a background for the relationship between these concepts, considering the context of fragment detection and counting as a measure of quality control in an automated pathology environment.

2.1 Quality Control in Digital Pathology

Artifacts present on slides, either during their preparation or digitization, can negatively affect slide quality and, consequently, the quality of diagnosis [23]. As WSI images are the primary material to automate the fragment detection and counting process, in sections 2.1.1 and 2.1.2, respectively, we analyze the effects of slide preparation and digitization on image quality and their approaches for quality control. We also explore the role of data annotation and quality as input for digital AI tools that perform quality control in Section 2.1.3.

2.1.1 Slide Preparation

Slide preparation compiles a series of processes that transform biological samples into observable microscopic slides, and it varies according to the biological sample's nature, which can account for distinct errors further along the testing pipeline. The most common samples are either histological or cytological, which describe either the tissues or cells' structure [86]. The slides must be well-preserved, transparent, thin, and with precise components distinguishable by color [23]. Table 2.1 describes the principal steps of preparing histology and cytology samples and the most common errors in each step.

Errors along the pipeline can cause artifacts - alterations in tissue details due to improperly fixed or mishandled samples during tissue processing [114] - which can affect image quality when digitizing slides. These artificial structures or alterations either stem from intrinsic characteristics of the specimens or are due just to their preparation, independently of their domain.

Artifacts have multiple classifications, such as prefixation, fixation, tissue-processing, staining, and mounting artifacts, and are also related to bone tissue, microtomy, or floatation and mounting

Table 2.1: Steps and common errors present on histology and cytology pipelines. Recreated from Brixtel et al. [23].

Histology		**Cytology**	
1. Sample	Bad tissue quality Sample air dried before fixation Error in sample identification	*1. Sample*	Difficulties during sampling Altered sample due to lubricant or other reagent Sample air dried before immersion into fixative
2. Recording / Grossing	Error during recording procedure Not enough fixation time Too small fixative volume Grossing irrelevant pathological regions	*2. Recording*	Error during recording procedure
3. Tissue Processing	Bad Dehydration	*3. Sample Processing*	Dysfunction in procedure
4. Embedding / Sectioning	Bad Tissue position Section too thick Artifacts	*4. Staining*	Baths not filled up correctly Out-of-date reagents Wrong protocol
5. Staining	Baths not filled up correctly Out-of-date reagents Wrong protocol	*5. Coverslipping*	No coverslip Bubble
6. Coverslipping	No coverslip Bubble	*6. Digitization*	Slide identifier not recognized Wrong area scanned Inappropriate color profile Sub-optimal focus plan selected
7. Digitization	Slide identifier not recognized Wrong area scanned Inappropriate color profile Sub-optimal focus plan selected	*8. Diagnosis*	
8. Diagnosis			

[114]. Some of the most common artifacts are tissue folding, wrinkling, scoring, and tearing, biological or foreign object contamination, dust or dirt particles, sample thickness variation, air bubbles, or ink and marker stains [114, 23]. These artifacts decrease slide quality and affect analysis accuracy [124]. To target this, researchers explore approaches that can detect and quantify the severity of the artifacts present on slides, such as methods for automated quality estimation [13, 62], low-resolution artifact detection [105, 54, 41, 47, 48], identification of tissue folds [17, 71, 92, 15], pen ink markers [10], and staining quality [137], among many others. CAD benefits from such systems as additional input to improve slide accuracy [23].

These artifacts often serve as edge cases that models may not be adequately equipped to handle but are imperative to address. Moreover, understanding how artifacts impact the quality of slides is essential for developing effective quality control or computer-aided decision systems.

2.1.2 Slide Digitization

Slide digitization involves the procedures required to convert a physical pathology slide into an identical digital counterpart. Ensuring minimal to no loss of information is crucial, thereby maintaining the diagnostic accuracy of digitized samples on par with the one performed on physical slides. We provide an overview of the most common issues related to slide digitization regarding

image format and compression in Section 2.1.2.1, color variations in Section 2.1.2.2, and out-of-focus areas in Section 2.1.2.3.

2.1.2.1 Whole-Slide Images: scanners, format, and compression

After preparation, slides are digitized into WSIs using WSI scanners. There are multiple market options for WSI scanners that can vary on objective lens type and magnification (focal planes), scanning camera and speed, illumination, and slide capacity [93, 23]. Despite differences among scanners, regulations are needed to ensure image stability and quality. While different scanners may not significantly impact diagnosis performance if image quality is maintained [95], the diversity of data sources is still essential for training deep learning models not to skew predictions.

WSIs are formatted as multiresolution pyramids that hold optical data at different magnifications, from the tiled baseline image at full resolution on the bottom to a thumbnail with reduced pixel dimensions on the top [42]. They also hold a macro image that is a low-resolution overview snapshot of the entire glass slide, helpful to guide the scanner detection system and for focus-point selection [42]. This structure allows for the retrieval of slide images at various zooming levels depending on the task at hand, expediting its display and facilitating sharing through networks, consequently changing the temporal and spatial domain of pathologic diagnosis [23, 87]. Still, issues like poor scan coverage and failure of automatic detection can arise when macro images are absent or misrepresent the slide under analysis [42, 9].

Scanners use file compression for efficient transmission and interoperability. The compression rate set by the scanner affects effectiveness, with low rates potentially introducing artifacts [23]. Although scanners commonly use lossy compression, its impact on image quality in machine learning tasks is minimal. The widely used JPEG format and its compression rate are typically sufficient to preserve information during transmission [66].

2.1.2.2 Color

Staining encompasses techniques designed to "highlight important features of tissue as well as to enhance the tissue contrast" [11] for interpreting slides using different stains and dyes. Laboratories may adopt distinct staining and scanning protocols, along with other stain dyes or scanning equipment, leading to potential color variations in samples of the same specimen across multiple laboratories or even diversity within the same institution [87]. Even though there are standardized staining protocols [72, 136], in practice, discrepancies may persist, and systems must perform cohesively in all cases. Ensuring color calibration in scanners or color normalization on WSIs helps establish a standardized foundation for posterior slide interpretation, either manually by pathologists or as a preprocessing step for CAD systems [23, 102]. Color calibration and color normalization works are described as follows.

Color Calibration [127] consists of comparing known color patches with a digitized image. Particularly for pathology, the general recommendation is to compare slides with a target slide with unique spectral characteristics, but its use is rare among scanners [33].

Researchers propose approaches to surmount this, including using different target slides [16, 134] or calibrations that eliminate the need for target slides [107, 31]. The consensus is that calibration reduces system-to-system variability, generating consistent outputs between laboratories and scanners. [23].

Color Normalization [102] consists of doing the mean color transformation from one image to another. Color normalization methods are divided into (1) global color normalization, which separates color and intensity information in space through histogram matching [70] or color transfer [99]; (2) stain separation, which estimates images' stain vectors and uses them for stain intensity correction and replacement, through either supervised [81, 65, 46] or unsupervised models [77, 18]; and (3) generative-model-based approaches, which use adversarial deep learning to apply a style transfer to a WSI and do not require prior references for learning like the other methods [104, 30, 138].

Color calibration and normalization techniques ensure images have consistent color for analysis. However, the impact of these techniques as a processing step for posterior computer-aided analysis cannot be related to increased accuracy [23]. Even though some studies show improvements using color normalization methods [112, 32], particularly with more recent generative-adversarial network (GAN) approaches [90, 119], many studies prove the opposite, reporting a reduction in accuracy [21, 43, 116]. However, these techniques

2.1.2.3 Blurriness

Scanners can produce WSIs with out-of-focus (OOF) areas, either locally, regionally, or globally, defined as poorly focused or blurry regions of interest [23]. Issues related to slide preparation, like the ones described on 2.1.1, may affect image acquisition, causing "thermal variations, internal or external vibrations, errors in the focus determination of a focus point, or in the generation of a WSI focus map" [23], which are the root for OOF areas. These artifacts can hinder pathologists' rendering of accurate diagnoses or impact the accuracy of automated image analysis [68].

To tackle this, automated focus quality assessment (FQA) aims to identify instances where a slide needs complete rescanning and to generate an FQA map, built according to local path-level focus estimations, which facilitates visual inspection and guides subsequent processing steps [23].

Approaches have evolved from simple image processing methods to manually engineered and learned feature classification [23]. Feature classification approaches need training data manually annotated by human experts or generated automatically using in-focus images that can be enhanced synthetically through blurring techniques, most commonly Gaussian blur [68, 23]. Research on each method is described as follows.

Image Processing Methods use traditional tools for image processing to detect OOF areas through features like contrast and entropy [121] or variations in brightness [12]. The complexity of these methods escalates quickly since each threshold has to be defined manually for every focus configuration.

Manually Engineered Feature Classification Methods require a handcrafted selection of measures of quality that can describe blurriness confidently as features, such as neighborhood contrasts, local intensity statistics, wavelet-based derivative-based, or sharpness-based features, to name a few [44, 55]. Considering manually labeled and automatically generated data, these methods are proven to have poor transfer ability, either between datasets, scanners, or stains [80, 55].

Learned Feature Classification Methods are mainly convolutional deep learning methods that require minimal human input regarding hand-engineered features or hand-picked thresholds, improving transfer ability [135]. Solutions are either specifically developed for WSI FQA, like DeepFocusNN [103] and FocusLiteNN [126] or adapted from standard architectures by retraining them for the task [8]. Generally, CNN-based methods trained on automatically generated data perform better than other FQA methods.

In line with other quality control measures, FQA provides helpful inputs that can minimize the influence of focus problems on the clinical workflow during and after digitization. Quantitative OOF maps can "flag regions that might otherwise be misclassified by image analysis algorithms, preventing OOF-induced errors" [68], which, along with the identification of blurred images, can aid pathologists in deciding which cases are still analyzable or the ones that need full rescans.

2.1.3 Data Annotation and Quality

As described in the last sections, many quality control methods rely on deep learning algorithms, from artifact and OOF area detection to stain normalization. These are often the best-performing approaches for all cases. Since they depend highly on data for training, data quality and quantity directly affect the outcome of these algorithms.

Deep Learning approaches particularly require large datasets for training, which must deal with the high data variability present in clinical routines by guaranteeing good generalization - the algorithm must handle both previously seen and new, unforeseen cases [82]. Good generalization is hard to achieve since unexpected cases are often artifacts that are, in fact, not frequently present on datasets. Moreover, fully supervised approaches demand pixel-level annotations, a time-consuming and resource-intensive process for laboratories [82]. Therefore, not only do datasets have to ensure that they are robust to variations, but they also must represent high-quality, carefully labeled, and annotated data [23].

Robust algorithms require high-quality annotations that fully support their outcomes. Standardizing annotation protocols is a way to ensure that quality measures are transverse to all datasets. Such protocols are defined by pathology professionals' established reference standards, though often through their subjective interpretations [85]. Considering principles that show a rigorous ground truth when annotating data for computer-aided pathology models is crucial for applying standardization, namely increasing the number of evaluators for each case, recruiting expert evaluators, establishing a fair resolution method in cases of grading discrepancies, and implementing a systematic voting process or using a neutral arbiter in cases of disagreement [83, 29]. These

parameters are only sometimes feasibly implemented, which could cause bias in the algorithm's results.

Addressing the data quantity challenge is crucial in utilizing deep learning models for CAD solutions. Constructing comprehensive datasets poses heightened difficulties, particularly in quality control tasks such as artifact detection or stain normalization. Handling edge cases is essential for seamlessly integrating algorithms into clinical practice, preventing potential misdiagnoses further along the testing pipeline. Data Augmentation, transfer learning, domain adaptation, and weakly supervised learning approaches are the leading solutions to alleviate data scarcity [23]. An extensive overview of these approaches can be found on [110].

2.2 Detection and Counting

Most tasks within digital pathology invariably need the identification/quantization of histologic primitives. Since our study concerns detecting and counting fragments that can be of various specimens, it is crucial to review the most common Machine Learning (ML) algorithms for object detection and classification since counting is usually handled as such.

In Section 2.2.1, we study Conventional Machine Learning (CML) methods; in Section 2.2.2, Deep Learning (DL) methods; and in Section 2.2.3 Graph Neural Network (GNN) methods. We also introduce evaluation and generalization concepts and techniques in Section 2.2.4 to outline the importance of robust models.

2.2.1 Conventional Machine Learning Methods

Conventional machine learning methods can be used throughout the framework for both localization and classification. These tasks rely heavily on extracting features from the WSIs to feed the classical ML models representative information according to the goal task, which is usually performed by assigning quantitative values to textures, color, and morphological and topological characteristics [22, 7]. Generally, the most common feature extraction methods are color histograms and wavelet scattering, which, among many others, are extensively described in [7].

Some of the most common classical machine learning techniques that are used to interpret samples, fed with the features extracted beforehand, include Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Random Forest classifiers (RF), Bayesian classifiers, Logistic Regression (LR), K-Nearest Neighbor Regression (KNNR), K-Means clustering (K-Means) and Ensemble Boosting (EB) [7]. Some practical applications of these techniques regarding DP are presented in Section 2.3.1.1.

2.2.2 Deep Learning Methods

Most studies for classification and detection use deep networks to extract features and, in many cases, provide predictions. Supervised, weakly supervised, unsupervised, and transfer learning are learning schemas within deep learning that researchers exploit to solve DP problems. We define

image recognition models relevant to our work in Section 2.2.2.1, as they are the basis of many vision-related tasks and can be used for image classification, and define specific object detection models in Section 2.2.2.2.

2.2.2.1 Image Recognition Models

These models are leveraged for various tasks across many fields, such as digital pathology, for both image classification and detection algorithms:

Convolutional Neural Network (CNN) [74] is a feed-forward neural network characterized by convolutional layers, the first in a series of layers that collectively identify intricate features within an image. A feature map is produced by performing convolutions using a filter that moves through the input image's receptive field. This hierarchical process continues through additional convolutional layers interleaved with pooling layers that reduce parameter dimensionality and model complexity. This process progressively identifies more complex patterns. The final layer, the fully connected layer, classifies the input based on the extracted features, directly connecting each node in the output layer to nodes in the previous layer. Figure 2.1 represents a simplified architecture of a CNN.

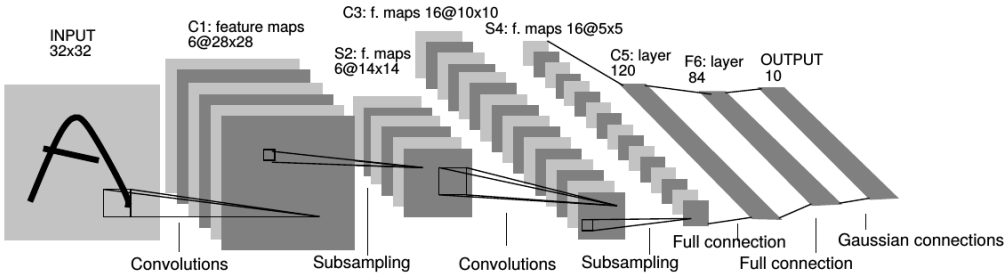

Figure 2.1: Generic architecture of Convolutional Neural Networks [73].

Residual Neural Network (ResNet) [52] is an advanced neural network architecture designed to address the vanishing gradient problem encountered in deep networks. It introduces the concept of residual learning through shortcut connections that bypass one or more layers. These shortcuts allow gradients to flow directly through the network, facilitating the training of very deep networks. Each residual block typically consists of convolutional layers and identity mappings, enabling the model to learn residual functions about the layer's inputs rather than unreferenced functions. This structure allows ResNet to maintain accuracy while significantly increasing network depth, improving performance in recognizing intricate patterns. The architecture usually ends with global average pooling followed by a fully connected layer for classification. Figure 2.2 exemplifies a generic residual block used in the ResNet architecture.

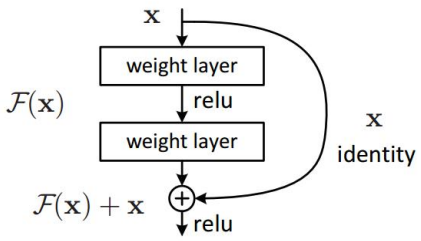

Figure 2.2: Residual learning building block [52].

Vision Transformer (ViT) [36] is a neural network architecture designed for image recognition tasks, characterized by its use of transformer layers instead of traditional convolutional layers. The input image is divided into fixed-size patches, which are then linearly embedded and combined with position embeddings to retain spatial information. These embedded patches are processed through a series of transformer encoder layers, which use self-attention mechanisms to capture intricate features and relationships within the image. This process continues through multiple transformer layers, progressively enabling the model to learn more complex patterns. The final layer, typically a fully connected layer, classifies the input based on the aggregated features from the transformer layers.

Self-Distillation with No Labels (DINO, DINOv2) [26, 89] is a self-supervised learning framework that leverages vision transformers (ViTs) for image representation learning without the need for labeled data. The architecture employs a student-teacher model where the student network learns to predict the output of the teacher network, known as knowledge distillation. This approach allows the model to capture intricate features within an image through a hierarchical learning process, described succinctly in Figure 2.3. DINOv2 is a foundation model based on DINO that introduces several enhancements to the original architecture, such as training acceleration, improved training stability, and feature representation quality. It refines the self-supervised learning process by incorporating advanced data augmentations incorporating curated data and refined losses, centering, and regularization. DINOv2 is considered a foundation model as it generates robust, generalizable, and universal features that can be leveraged for many tasks, both at pixel or image level.

2.2.2.2 Detection Models

Object detection models can be divided into two main types: One-stage object detectors and Two-stage object detectors. One-stage object detectors aim to detect objects within images in a single forward pass without requiring a previous separate feature proposal step. Some of the most common one-stage detection models are as follows:

You-Only-Look-Once (YOLO) [97] is a one-stage object detector designed to predict bounding boxes and class probabilities. The entire image is initially divided into various

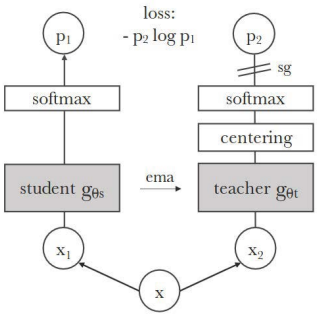

Figure 2.3: Simplified view of the DINO model [26].

grids of different sizes, and anchor boxes are generated within each grid based on predefined scale and size parameters. Each anchor box simultaneously predicts the objectness score, box center offset (x and y), box width, box height, and class scores in a single step. The network quickly processes predictions without the need for separate feature extractors. There are several YOLO versions that improve over this initial approach, with the latest being YOLOv9 [122], released in February 2024, that incorporates Programmable Gradient Information (PGI) with a novel Generalized Efficient Layer Aggregation Network (GELAN).

Detection Transformer (DeTR) [25] integrates transformers into a single-stage detector framework. DETR operates by treating object detection as a direct set prediction problem. The input image is processed through a backbone CNN to extract features, which are then passed into a transformer encoder-decoder structure. Within the transformer, each position encodes a global context of the image and predicts the presence and location of objects using a set-based prediction approach. This eliminates the need for predefined anchor boxes by directly regressing bounding boxes and assigning class probabilities through self-attention mechanisms.

Two-stage object detectors detect objects within images in two separate steps: a region proposal network (RPN) and a subsequent object detection network. Some of the most common two-stage detection models are as follows:

Fast and Faster-RCNN [100] is a two-module object detection network. Fast-RCNN and Faster-RCNN behavior only differs in how region proposals are obtained: Fast-RCNN uses Selective Search, which iteratively finetunes over-segmented input images to predict regions, while Faster-RCNN uses a Region Proposal Network (RPN), which uses feature maps extracted from the input images to predict regions. Both methods identify potential object regions within an image. The second module is a CNN (VGG-16) used to classify objects in the proposed regions.

Mask-RCNN [51] is an extension of Faster R-CNN, introducing a third branch for object mask prediction. While Faster R-CNN has two outputs per candidate object, Mask R-CNN

enhances it by incorporating pixel-to-pixel alignment. The architecture retains the two-stage process, with the first stage being an RPN, proposing candidate object bounding boxes, and the second predicting class labels, bounding-box offsets, and binary masks for each region of interest (RoI). During training, a multi-task loss is defined on each sampled RoI, including classification, bounding-box regression, and average binary cross-entropy loss for mask prediction.

2.2.3 Graph Neural Network Methods

Graph Neural Networks (GNNs) are deep learning methods based on neural networks capable of handling data represented by graphs. In graph-structured data, entities are represented as nodes, and relationships between entities are represented as edges. GNNs can capture node and graph-level relationships and often complex dependencies between nodes and edges. These analyses are challenging due to the heterogeneity and diversity of graphs, their irregular and considerable structure, and the incorporation of other interdisciplinary domains. GNNs have several applications, from computer vision to recommender systems and software mining, among many others, with an impressive performance on link prediction and classification tasks [128]. We define the generic GNN framework in Section 2.2.3.1 and some other standard GNN models in Section 2.2.3.2.

2.2.3.1 Generic Definition

GNNs aim to update node representations iteratively by combining the representations of neighboring nodes with their representation from the preceding iteration. Considering that graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ and $\mathscr{V}$ is the set of nodes with features X_v, where $H^0 = X_v$, the learning process combines two steps, applied to each layer k [128]:

Aggregate, which applies a permutation-invariant function to the node's neighbors, generating the node's features:

$$a_v^k = Aggregate^k\{H_u^{k-1} : u \in N(v)\} \tag{2.1}$$

where $N(v)$ is the set of neighbors for the v-th node.

Combine, which updates the node representations by combining the aggregated features with the current node representations, generating the node embeddings:

$$H_v^k = Combine^k\{H_v^{k-1}, a^k{}_v\} \tag{2.2}$$

This iterative process, also called message-passing, is repeated for a fixed number of steps or until convergence, resulting in the final node representations that can be used for downstream tasks, like node classification or link prediction. This generic framework is adapted in the literature to other variants, like Graph Convolutional Networks (GCN), Graph Autoencoders (GAE), and

Graph Recurrent Networks (GRN), to name a few, for both supervised and unsupervised learning approaches [128].

The outputs of the GNNs can be used to perform both node-level and graph-level predictions. Node-level predictions compute values for each node, which is useful for classification and regression tasks since the labels of each node are predicted, and the node embedding is then fed to a Multi-Layer Perceptron (MLP). Graph-level predictions predict a single value for the whole graph, typically used to determine graph similarities or whole graph classifications, in which the node embedding follows a pooling process before being fed to a separate MLP [3].

2.2.3.2 Standard Model Definitions

The following models are typically mentioned in literature and are useful for various tasks, namely for detection and classification, so understanding their behavior is critical for our research. These models can be generically defined as follows [128]:

Graph Convolutional Network (GCN) [67] extends upon the convolutional neural network to deal with graph-structure data instead of grid-based data by applying localized first-order approximation of spectral graph convolutions, through normalized Laplacians. For each layer, the node embeddings are updated according to the following propagation rule, which generally defines the Aggregate and Combine functions:

$$H^{k+1} = \sigma(\tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}H^{k}W^{k}) \tag{2.3}$$

where $\tilde{A} = A + I, I \in \mathbb{R}^{NxN}$ represents the adjacency matrix of the graph with self connections, so self features are considered when updating node embeddings; $\tilde{D}$ is a diagonal matrix that represents the degree of $\tilde{A}$; σ is the activation function, like ReLU; and $W^k \in \mathbb{R}^{FxF'}$ is the learnable weight laywise linear transformation matrix for the k-th layer. In simpler terms, GCNs aggregate the neighborhood features with the ones from the present layer through an aggregator function, which is then combined with the learnable weight matrix followed by the node activation to extract the embeddings.

GraphSAGE [49] can be viewed as an extension of GCN. It is a spatial-based graph neural network with a general inductive framework (it does not require the entire graph structure during learning) that, instead of training individual embeddings for each node, learns a function that generates embeddings by sampling and aggregating features from a node's local neighborhood. It uses LSTM and Pooling aggregators, unlike the mean aggregator for GCNs and concatenates, instead of summing to combine the learned weight matrix and the node activations. Since it does not require the full graph for training, it uses mini-batches that only contain the nodes that are being computed, making it much more computationally lightweight and suitable for large graphs. The Aggregate and Combine functions for this

model are defined as follows:

$$H_{\mathcal{N}(v)}^{(l+1)} = Aggregate_l(\{H_u, \forall u \in \mathcal{N}(v)\}) \tag{2.4}$$

where $H_{\mathcal{N}(v)}^{(l+1)}$ represents the node embedding generation at the current $(l+1)$-th depth from the target node $v \in \mathcal{V}$.

$$H_v^{(l+1)} = \sigma(W^{(l+1)} \cdot Concat(H_v^{(l)}, H_{\mathcal{N}(v)}^{(l+1)})) \tag{2.5}$$

where $W^{(l+1)}$ is the learning weight matrix and σ the activation function.

Graph Attention Network (GAT) [120] is a spatialbased graph neural network that uses Self-Attention as the aggregator for neighborhood features, by enabling the assignment of different weights when aggregating information. In simple terms, Self-Attention performs a weighted mean of the node features through the application of a LeakyReLU, being essentially a single fully connected layer parameterized by a weight vector **a**:

$$e_{i,j} = attn(H_l^i, H_l^j) = LeakyReLU(\mathbf{a}[WH_l^i || WH_l^j]) \tag{2.6}$$

This self-attention mechanism is used to compute an attention score $\alpha_{i,j}$ that determines the importance of node j to node i, by applying a softmax function to the attention block:

$$\alpha i, j = softmax(e_{i,j}) \tag{2.7}$$

The aggregated node features are then combined by multiplying with a learnable weight matrix followed by a non-linear activation to extract the embeddings. The node embedding update can be written as follows:

$$H_{(l+1)}^i = \sigma(\sum_{j \in \mathcal{V}_i \cup \{i\}} \alpha_{i,j} H_l^{(j)}) \tag{2.8}$$

This mechanism could be extended for multi-attention, where K-independent attention mechanisms are executed in parallel. We only described a single block for simplification purposes.

2.2.4 Evaluation and Generalization

Developing robust and reliable models requires extensive evaluation to guarantee model performance remains stable even when presented with unseen data and under varying conditions. In this Section, we explore methodologies and metrics used to evaluate the performance of detection and counting models. Additionally, we discuss strategies for verifying and ensuring the generalizability of these models when faced with changes in data and clinical settings.

2.2.4.1 Evaluation Metrics

Evaluation metrics are adopted to assess and compare the performance of each model effectively. Despite the existence of specific performance metrics adapted to each domain, most of them rely on the combinations between the ground truth class and predicted class, namely true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) [140].

Concerning detection, it is crucial to distinguish correct and incorrect predictions spatially. The Intersection over Union (IoU), also called the Jaccard Index, is frequently used to make this distinction in detection by measuring the area of overlap between the predicted bounding box and the ground truth bounding box and comparing it against a predefined threshold tailored for the task at hand [91]:

$$IoU = \frac{area(B_b \cap B_{gt})}{area(B_b \cup B_{gt})} \tag{2.9}$$

These are other relevant measures for our work regarding object detection:

- *Precision* [91] represents how well the model can identify only relevant objects. It is given by the fraction of correct positive predictions:

$$P = \frac{TP}{TP + FP} \tag{2.10}$$

- *Recall* [91] represents how well the model can find all the relevant cases. It is given by the fraction of correct positive predictions among all the ground truths:

$$R = \frac{TP}{TP + FN} \tag{2.11}$$

- *Average Precision* [91] is given by the weighted average of the precision at different recall levels, calculated by the area under the precision-recall curve:

$$AP = \sum_{n}^{N} (R_{n+1} - R_n) P_{\text{interp}}(R_{n-1}) \tag{2.12}$$

  where $P_{\text{interp}}(R_{n-1})$ is the maximum precision whose recall value is greater or equal to R_{n+1}.

- *Mean Average Precision* [91] measures the accuracy of object detectors by averaging AP over all classes or at specific IoU thresholds:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{2.13}$$

  where AP_i is the AP in the ith class and N is the total number of classes evaluated.

Concerning classification, these are the most relevant evaluation metrics for our research:

- *Accuracy* [40] represents the frequency with which the model correctly predicts cases. It is calculated as the proportion of correct predictions overall predictions:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.14}$$

- *F1-score* [40] represents the harmonic mean of precision and recall, providing a balance between the two metrics:

$$f1 = 2 * \frac{P * R}{P + R} \tag{2.15}$$

  where P is precision and R is recall.

- *Mean Absolute Error* [40] measures the extent to which predictions differ from the actual probability by the absolute value of this difference:

$$mae = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.16}$$

  where n is the total number of predictions, y_i is the ground truth value and $\hat{y}_i$ is the predicted value.

2.2.4.2 Domain Generalization

Domain Generalization (DG) involves training a model using data from one or more related but different source domains to effectively generalize to any out-of-distribution (OOD) target domain [139]. This is based on the assumption that the model cannot access all the data domains during training, as it is common for many machine-learning tasks. As mentioned in Section 2.1.3, this is particularly true in pathology settings since there are many variability sources within the clinical processes while gathering relevant data and between the data in itself.

Various methods are proposed in the literature to counter the effect of the shift in distribution between the source and target data for both multiple or single source domains. Some of the most relevant methods fall into the following categories [139, 123]:

Data Manipulation These methods focus on manipulating input data so that models can access a broader and improved representation during training, encouraging generalization when learning. Examples are data augmentation methods, such as image transformations, task or domain adversarial gradients, and learnable or random augmentation networks, among others. Another approach is to use generative models to expand source data domains.

Learning Strategies These methods directly apply various strategies during training that promote domain generalization. These methods include ensemble learning, self-supervised learning, regularization strategies, reinforcement learning, and meta-learning, among others.

Representation Learning These methods encompass two distinct strategies: domain invariant representation learning, which is based on the intuition that if the model is invariant to the domain shift in the source data, it will also be invariant to the domain shift in the target data; and feature disentanglement, that decouples domain-shared or specific features into generalizable representations. Some domain-invariant learning strategies include kernel methods, explicit feature alignment, domain adversarial learning, and invariant risk minimization.

Average or worst-case performances on domains held out from training are commonly used to test domain-shift scenarios and verify how the model generalizes to unseen data or different domains [139]. However, model selection is crucial when evaluating DG models, as the distribution between data splits directly affects performance interpretations [45]. Three strategies are considered [45]: *Test-domain validation set*, which uses training and validation subsets pooled from per-domain training and validation splits and considers the highest accuracy on the validation set; *Leave-one-domain-out cross-validation*, which considers k training models each without a training domain and averages accuracies over the performance on held-out domains; and *Test-domain validation set (oracle)*, which considers the accuracy of on a validation set that has similar distribution to the test domain. This last requires prior knowledge of the test domain, which is not always available.

2.3 Detection and Counting applied to Digital Pathology

The analysis of WSIs allows the extraction of valuable quantitative and qualitative features useful to several tasks, such as localization, segmentation, detection, and classification of biological tissues [61]. When applied to the pre-analytical testing phase, this analysis helps control the images' quality, which is posteriorly interpreted manually by pathologists or fed to AI diagnosis tools. It can relieve pathology clinicians of time-consuming and labor-intensive preprocessing tasks, namely manually comparing fragments in slides with their corresponding macroscopic reports [9], as we explore in this work. Consequently, expediting quality control protocols also expedites testing results, crucial in time-sensitive diagnosis.

As we have described in Section 2.1, analysis of WSIs is a challenging endeavor: images are large, require multiscale awareness, and can contain artifacts that hinder model performance, which may also be affected by poorly generalized training data. Considering these challenges, it was necessary to adapt existing models or develop new ones targeted to the digital pathology domain [110, 61]. Cell and nuclei segmentation, tissue classification, tumor detection, and disease prediction/prognosis are the most common tasks in this domain. Tasks regarding counting structures are not commonly explored in the field, although counting mitosis or cells is considered in some disease prediction models [110].

As such, we explore the most relevant applications of detection and counting in digital pathology in the following sections. We also review the literature related to our work for fragment detection and counting in Section 2.3.2.

2.3.1 Detection and Counting

This Section showcases some practical examples of ML approaches applied to general digital pathology tasks related to object detection, classification, and localization. As specific applications of counting structures in pathology images are scarce, and counting can be approached as a classification/localization problem, we mention applications related to these tasks as well as for object detection.

2.3.1.1 Conventional Machine Learning Methods

Some of the applications in studies for classification, localization, and detection-related tasks within digital pathology are listed in Table 2.2. There are multiple tasks within the field, with different motivations and concerns, so we group them into specimen classification, region-of-interest (ROI) localization, and tumor detection, as these are broader categories.

Table 2.2: Conventional machine learning techniques for object detection by digital pathology task [7].

Method	Specimen Classification	ROI Localization	Tumor Detection
Support Vector Machine (SVM)	[27] [19] [6]	[98] [94]	[38] [37] [53] [22]
Linear Discriminant Analysis (LDA)	[27]	-	-
Random Forest (RF)	-	[98]	[22]
Bayesian Classifier	[27]		[37]
Logistic Regression (LR)	-	-	[38]
K-Nearest Neighbor Regression (KNN)	[27] [19]	-	[22]
K-Means Clustering	-	[94] [84]	-
Ensemble Boosting (EB)	-	[98]	[38] [22]

Some publications compare the performance of CML methods with DL methods, and generally, the DL methods outperform the CML ones. The studies that use only CML approaches for the goal task are, in most cases, before the widespread use of DL methods in the digital pathology realm. Today, conventional machine learning approaches are primarily used in collaboration with deep learning methods.

Extracting relevant features is crucial for the success of these methods, as they define the distinction between tissues, cells, or simply areas of interest in the samples that drive the predictions. However, extracting meaningful handcrafted features can be complex and computationally burdening and often requires extensive prior domain knowledge of the disease to define highly representative features. This limited what could be learned from available data, leading researchers to explore deep learning techniques that automatically extract global features and that do not exclusively perform the goal task [22].

2.3.1.2 Deep Learning Methods

Deep learning has been widely used as an effective technique for multiple tasks within digital pathology and applied to various cancer types. Computer-aided diagnosis solutions for detection and classification-related problems usually use CNN variants adjusted to the specific task they are solving [110]. We list some applications of deep learning methods per the learning schema adopted:

Supervised Learning Supervised learning relies on training over labeled data to make predictions during inference time. Classification methods are the most common, but regression and segmentation models can also be applied to the detection and classification of bio-structures.

Considering classification, CNNs are the gold standard for region-level and image-level tasks, although attention-based models have gained some traction in global image analysis [110]. In digital pathology, local-level approaches have been used for various tasks, such as mitosis [115, 96] and cell/nuclei detection and classification [64, 101]. As for global-level approaches, their applications focus more on disease prediction or grading, such as detecting breast cancer metastases [78, 20, 69] or identifying and classifying invasive breast cancer [39, 35, 132].

Considering regression, approaches have been used in digital pathology by exploiting Fully Connected Networks (FCN) for mitosis [28] and nuclei [131] detection and classification tasks and applying CNN-based methods for nuclei [109] and cell detection [130]. Xie et al. [129] also explored Fully Convolutional Regression Networks for automated cell counting.

Weakly Supervised Learning Weakly supervised learning models use scarcely labeled data to perform predictions. They exploit image-level annotations to automatically infer pixel/path-level information, which is particularly valuable in DP since annotating patches or pixels is complex and time-consuming for pathology technicians [110]. Weakly supervised learning has been applied to numerous detection and classification tasks in digital pathology for gastric cancer, [125], prostate cancer [24], breast cancer metastasis [5], and mitosis detection [76]. Huang and Chung also explored weakly supervised learning methods for localizing cancerous evidence in histopathology images [60].

Unsupervised Learning Unsupervised learning aims to identify patterns in underlying data without the help of labels. These methods use raw input data without expert annotations, making them hard to adapt for heavily label-reliant tasks like object detection [110]. Nonetheless, unsupervised learning approaches applied to detection and classification are present in, for example, nuclei detection [56]. Hu et al. [58] also used GANs to extract cell-visual representations, which can be used for cell counting. Research in this area is valuable since it helps circumvent the lack of specialized annotated data in the DP field.

Transfer Learning Transfer learning aims to apply knowledge from one source domain to a target domain, relaxing the assumption that the train and test set should be dependent [110]. It

is frequently used in DP since large pre-trained models are readily available and can be finetuned on task-related images. It is typically done with VGGNet [108], InceptionNet [113], ResNet [52], MobileNet [57] and DenseNet [59], as with other variants trained on ImageNet or other large image datasets, as the COCO dataset for YOLO [97]. Transfer learning models have been used for several detection tasks within DP, for example, breast cancer metastasis detection and classification [79, 75], and cell detection [118].

2.3.1.3 Graph Neural Networks

Histological images portray the micro-anatomy of a tissue sample, serving as a diagnostic tool for pathologists who analyze morphological alterations in tissues, spatial cell relationships, cell density, and other relevant factors. GNNs can model these relationships more efficiently than conventional or deep learning methods by transforming biological samples into graph representations considering their morphology, topology, and spatial representation, customized to tackle the defined task, thus avoiding the limited context given by patch-based detection methods [3]. Models like CNNs fail to consider these relationships within their computation, so fine-grained dependencies, critical to understanding and perceiving visual data, go unnoticed. GNNs address this issue by modeling those relationships and leveraging drops in performance when faced with unfamiliar samples [128].

A traditional workflow for graph-based tasks in digital pathology can be defined by constructing the graph to represent the input data, modeling the GNN-based algorithm that performs the prediction, and interpreting the resulting graph for the given task. Graph construction is essential to build the input for the models, and it can be described by the following steps [3]:

> **Node definition:** Background and tissue regions are segmented by Gaussian smoothing and Otsu thresholding. Nodes can be defined by detected or segmented cells (cell graphs), fixed-sized patches (patch graphs), or tissue regions (tissue graphs), done through clustering, segmentation, or selection algorithms.
>
> **Node embeddings:** Features are extracted according to morphological and topological properties, such as shape, size, orientation, nuclei intensity, and chromaticity. This is done using deep neural methods (CNNs), aggregated features from neighboring patches, or self-supervised approaches (autoencoders).
>
> **Edge definition:** Edges are defined by how likely two nodes interact, which can be by predefined proximity thresholds, Pearson correlation-based graphs, probabilistic models, distance-based graphs, or simply by defining an adjacency graph by the centroid distance [106].

The input graph is then processed using a graph-based deep learning model that analyzes the graph-structured data. Standard GNN models namely GCN, GraphSAGE, GAT (as described in 2.2.3.2) as well as their variants or other models, such as, for example, Graph Isomorphism Networks (GIN) [133], have been explored to solve classification and detection tasks within digital

pathology, for breast [63], lung [2], colorectal [111] and prostate [125] cancer detection, among other tasks.

Graph pooling is finally performed to reduce computational complexity, minimize graph output for posterior predictions, and provide relationship details and interpretations. Global Pooling, or the readout layer, applies simple aggregators like the mean, sum, or even global attention mechanisms that only focus on relevant nodes to reduce the graph to a single pooled vector for all nodes [3]. Hierarchical Pooling, however, learns a hierarchical representation of the graph and is used through another graph pooling layer that pools information from multiple vertices to one vertex to reduce the graph, such as DiffPool and SAGPool [3].

Generally, graph-based techniques have shown impressive results in most cancer detection tasks within digital pathology since using entity-based models allows for the interpretability of their semantic relationships, making algorithms more robust to unknown samples [3].

2.3.2 Application to Fragment Detection and Counting

As discussed in the previous sections, recent advances in detection and classification methods can positively influence results in tasks related to digital pathology. Applying GNN-based methods can capture hidden morphological and topological relationships between tissues that generally improve results.

General detection or classification methods for digital pathology specifically applied to fragment detection and counting have barely been explored, except by Albuquerque et al. [9]. New approaches for detection and counting are worth exploring to bridge this gap, which could potentially improve on previous results. The work done by Albuquerque et al. and its outcomes are highly relevant to our research, so we detail the methods they explored in the following sections. In Section 2.3.2.1, we explain the annotation and labeling protocol followed for the images in the dataset. We describe their conventional machine learning approach and deep learning strategies in sections 2.3.2.2 and 2.3.2.3, respectively. In Section 2.3.2.4, we review the outcomes and limitations of their work.

2.3.2.1 Annotation and Labelling

The whole-slide images used for the task include 1276 samples from colorectal biopsies and polypectomies specimens, manually annotated in 11,300 fragments and 3,517 sets by one pathologist and two biomedical scientists, with an 80/20 split on train and test images.

Fragments represent individual tissues, and sets are groups of fragments repeated within the slide. Slides have repeated cuts as a precaution to prevent the potential loss of slide material caused by artifacts, inadequate focus, or similar factors. This redundancy ensures that if any of the repetitions is compromised, clinicians can still make informed decisions based on the unaffected repetitions, thereby maintaining the reliability of diagnostic assessments. When slides are mounted, they can be of 3 types, as represented in figure 2.4: a) several sets with only one fragment (number of fragments equal to the number of sets); b) same set with multiple different fragments;

and c) several copies of the same set of different fragments. Sometimes, equal sets can have a different number of fragments since repeated cuts are not always completely equal.

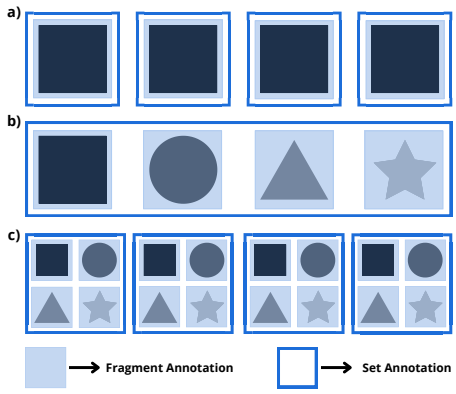

Figure 2.4: Fragment and Set Annotation. Recreated from [9].

The images contain annotations based on the protocol used for slide mounting. There are two types of annotations: counting and spatial annotations. Counting annotations follow the protocol described above but account for the number of fragments per set (fragments/set), as pathologists assign the real value to samples when counting. For example, in the case of a) in Figure 2.4, the ground truth would be 1, as there is a single fragment in each set, and in the case of b) and c) would be 4, as there are four fragments/set. Cases where the number of fragments/set is ten or higher are annotated as containing various fragments. Spatial annotations allow for detecting fragments and sets, as they are annotated with each corresponding rectangular bounding box coordinates, with a label of 0 for fragments and 1 for sets. The variability in sets and fragments and their topological and morphological characteristics are a big challenge for counting and detection since the developed models must account for all cases. Figure 2.5 shows some real examples of tissue differences.

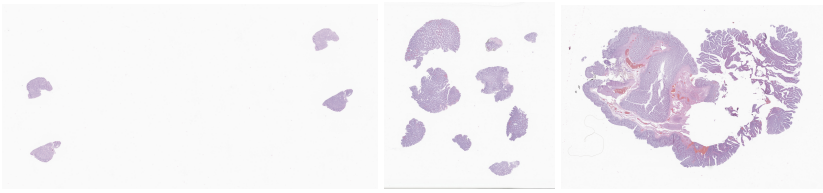

Figure 2.5: Examples of macro slide images present in the dataset.

2.3.2.2 Conventional Machine Learning

This first approach applies supervised classifiers to binarized histopathology images for fragment and set classification, followed by unsupervised hierarchical clustering for their localization. The steps performed can be divided into image pre-processing, feature extraction, binary classification, and grouping and are described as follows.

Image Pre-Processing applies several transformations to the image, namely removing black spots using a region-growing algorithm in the blue channel, binarizing the image through Otsu's adaptive thresholding, applying morphological operations to remove noise, and finally, reversing the image to have white tissue objects on a dark background.

Feature Extraction extracts representative features from the image from each pair of connected components and uses them to calculate pairwise features (distance between centroids, percentile distance between points in object contours, normalized area of the bounding box, and circularity ratios), to decide if they belong to the same fragment or set. There is also noise removal by only considering connected components that are at least 5% bigger than the most prominent component.

Binary Classification predicts whether pairs of connected components belong to the same fragment or set by training five different classifiers (LDA, QDA, NB, LR, SVM) with cross-validation. Training of the entire dataset uses the best parameters for each model and is done twice: once to check if the components belong to the same fragment and another to check if they belong to the same set.

Grouping creates two graphs: a Strong Graph with highly confident connections (threshold of 0.9) and a Weak Graph with all pairwise predictions. After identifying connected subgraphs in the Strong Graph, additional connections are considered between pairs of subgraphs that likely represent the same fragment/set. This involves calculating mean predictions between components and establishing connections if they surpass a permissive threshold (above 0.5). The process is repeated until no more pairs of subgraphs meet the criteria.

To tackle discrepancies generated by the pre-processing step, they explore another approach that applies the feature extraction, binary classification, and grouping steps to patches limited by the ground truth fragments' bounding boxes instead of the whole image.

2.3.2.3 Deep Learning

The deep learning approach for detecting and classifying fragments relies on two state-of-the-art detection deep learning models, Faster R-CNN [100] and YOLOv5 [117]. The hyperparameters used were optimized by a grid search, resulting in the best parameters for YOLOv5 being an image size of 512x512, batch size of 32, and 200 training epochs, while Faster R-CNN performed best with a batch size of 8; both models utilized Stochastic Gradient Descent with a starting learning rate of 10^{-4}.

2.3.2.4 Outcomes and Limitations

Regarding conventional ML methods, results indicate that the preprocessing step does not induce considerable inconsistencies in images since using ground truth fragments hardly improves metrics across classifiers compared to component labeling on the entire image. Logistic regression

emerges as the top-performing traditional model, consistently achieving mAP@0.5 values over 0.8 for fragment, set, or both classes detection. In contrast, while SVM excels in fragment detection, it shows lower mAP@0.5 values for set detection.

Deep learning models, particularly YOLOv5, outperform conventional methods in most metrics, achieving a top mAP@0.5 of 0.977 for all classes. At the same time, Fast R-CNN remains competitive in precision and mAP@0.5, though it requires more time and effort to optimize hyperparameters compared to YOLOv5.

Generally, the methods employed performed positively for detecting and counting fragments. Nevertheless, challenges may emerge when applying these techniques in clinical settings. The findings could exhibit bias towards the specific domain, as slides of a diverse nature might present unique common patterns. Additionally, the data employed might lack adequate representation of edge cases like slides with artifacts or uncommon patterns and size variations. Moreover, given that the data originates from a single lab, it may generalize poorly to WSIs obtained using different scanners or resolutions. GNN approaches could improve these results as they rely on the intrinsic relationships between objects, which provide more context than image patches, and effectively detect and count these different tissues. In their work, no result metrics are presented that effectively count the fragments and sets in the images. Gathering these values is crucial to aid pathology technicians in cross-checking the quality control process.

2.4 Summary

In this chapter, we provided an overview of the background knowledge of detection and classification methods applied to quality control in digital pathology, namely for fragment detection and counting.

In digital pathology, the quality control of WSIs is crucial for accurate diagnosis and the effectiveness of CAD systems. Slide preparation and digitization are critical stages, with potential errors and artifacts that can impact image quality. We analyzed solutions for handling these artifacts relating to color and blurriness and the importance of using high-quality standardized annotated data for training automated solutions.

Fragment detection and counting is presented as a quality control checkpoint that ensures the consistency and reliability of digital slides compared to their physical counterparts. We analyzed the most common detection and classification models as approaches for solving the task at hand, as fragments and sets need to be identified and counted to improve the quality control process effectively.

Finally, we reviewed known applications of detection and classification in the digital pathology realm, highlighting our task of detecting and counting tissue fragments, which has barely been explored in literature and which we build upon. As far as we know, deep graph-based approaches have never been studied for fragment detection and counting in histopathology samples.

Chapter 3

Improving Fragment Detection and Counting

This chapter proposes a methodology for improving fragment detection and counting. We introduce the motivation for this problem in Section 3.1, analyze the dataset used throughout the work in Section 3.2, and describe the methods applied to tackle the issue in Section 3.3. We detail the experimental setup in Section 3.4 and discuss the results and accompanying reflections in Section 3.5.

3.1 Problem Statement

As discussed in the previous chapter, digital pathology must rely on quality control procedures to ensure that computed-assisted diagnosis is as reliable as manual evaluations. Fragment detection and counting are routine procedures that could benefit significantly from automated approaches due to their time-consuming and manual-intensive nature.

The previously described approach by Albuquerque et al. [9] already demonstrated positive results in correctly detecting sample fragments and sets, proving that the practicability of such systems is achievable. However, specimens have high shape and size variability, so the model is not sufficiently robust, especially for OOD samples, which is crucial for integration into clinical routines. Additionally, the published work did not provide results for counting, which is the central task in real laboratory settings, as pathologists manually compare the number of fragments per set present to assess if the mounted slides match their macroscopic lab report.

In this work, we look at the detection and counting problem as an improvement of the work already developed by Albuquerque et al. [9], focusing on refining erroneous predictions without hindering the correct predictions of familiar cases while improving metric results. Common errors consisted of the following: a) Large fragments that are not homogeneous, so tissues within that fragment are detected independently when they should not; b) Fragments with disconnected tissues that should not be counted independently (in the example, the top disconnected part of the middle fragment is considered another separate fragment by the model); c) Artifacts or negligible tissues

that should be ignored and not detected; d) Sets that have slight differences and are not detected as such. Figure 3.1 shows examples of these errors obtained by implementing their previous best-performing model on their original dataset.

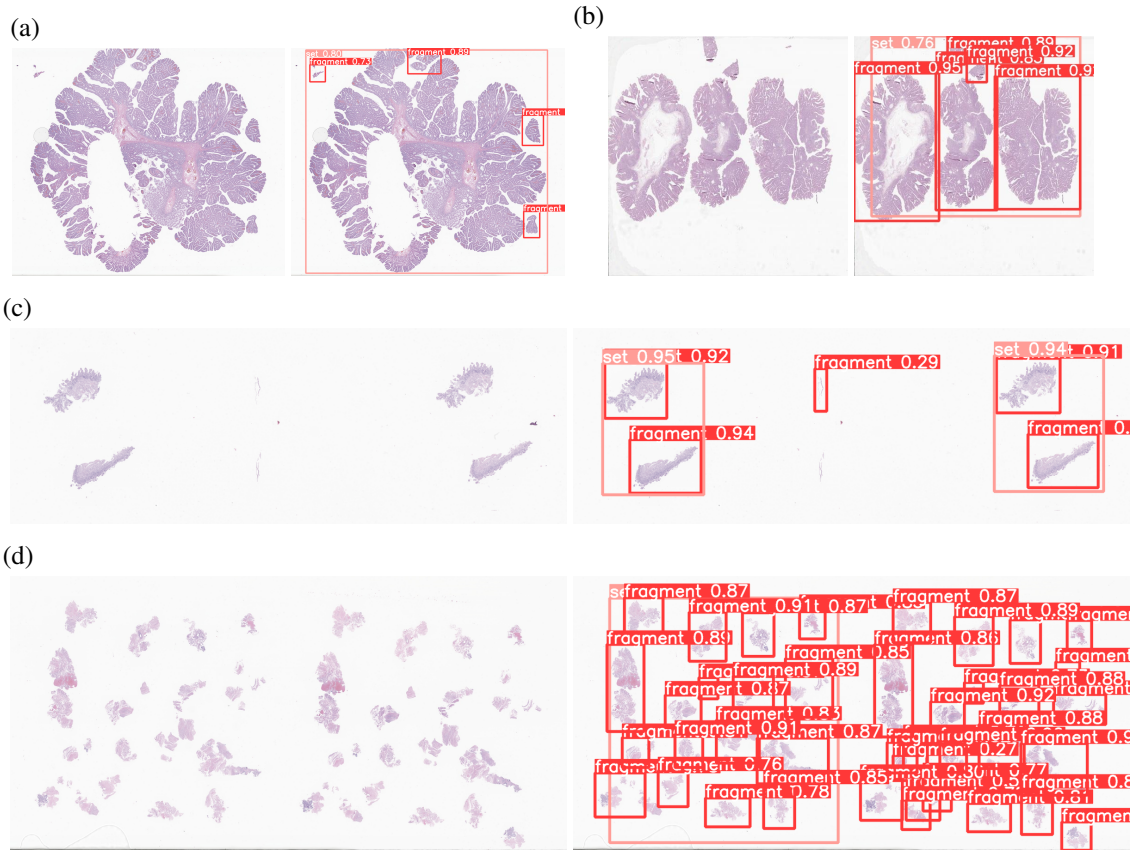

Figure 3.1: Common errors with the fragment detection and counting model proposed by [9].

To address these and other issues and enhance the results of the previously developed models, we propose a methodology for overall detection and counting improvement, focusing on a detailed analysis of the performance of the models to assess robustness. Our objective is to solve the detection of fragments and sets in the WSIs and count the fragments per set and sets present in the samples. The pathology technicians use the fragments per set value in the manual comparison process. These are the tasks we focus on further in this chapter.

3.2 Dataset Analysis

The problem we aim to improve is highly linked to the data, as the WSIs are the primary vehicle for the research. We must detail and analyze the samples available for our study before addressing the proposed methodology to provide a broader understanding moving forward.

For this study, we used an extension of the previous dataset used by Alburquerque et al. [9]. Two pathologists and two biomedical scientists manually annotated the dataset, following the standard protocol described in Section 2.3.2.1. All cases were retrieved from PoTURgal's IMP Diagnostics Laboratory data archive and digitized with 2 Leica GT450 WSI scanners at 40x equivalent magnification.

This extended version has 2053 train images, 499 validation images, and 701 test images. Although both train and validation images have counting and spatial annotations, test images only have counting annotations. Consequently, algorithms that depend directly on spatial annotations cannot be evaluated using the test set, as is the case of the detection models. The same goes for the task of counting sets, as the counting annotations only have the count of fragments per set, as this is the objective task that pathology technicians need to solve in their laboratory routines. Since the counting annotations do not include the count for sets, we also do not use the test set to evaluate this task.

The histopathology images present in the dataset are characterized by their specimen type - *Biopsy, Surgical Specimen, Polypectomy, TUR (Transurethral Resection) and Biopsy or Polypectomy*; organ type - *Gastric, Ovary, Breast, Prostate, Duodenum, Jejuno/Ileum, Colorectal, Bladder, Esophagus, Seminal vesicle, Ganglia, Cervix, Uterus, Liver, Skin, Gallbladder, Appendix, Soft tissue, Oral Cavity, Vulva, Pancreas, Thyroid, and Others*; and staining technique - *H&E (hematoxylin and eosin) staining and Immunostaining*. Figures 3.2, 3.3, and 3.4 show the Distribution of the cases in the dataset by specimen type, organ type, and staining technique, respectively.

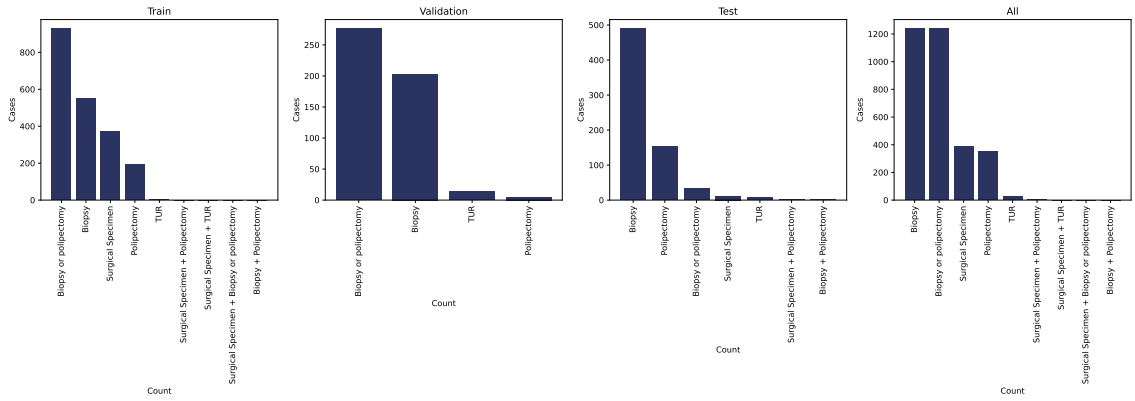

Figure 3.2: Distribution of cases by specimen type for the train, validation, test, and all sets.

The test set is built considering the specimen, organ, and staining technique type distribution in the train and validation sets, allowing a thorough evaluation of some OOD data.

Besides these attributes, cases are also described by the ground truth associated with their counting annotations, which reflect the number of fragments per set in the image. These values follow a "gold standard" defined by the annotators' agreement on the number of fragments per set in each image, as some cases are ambiguous even among technicians. Following the "gold standard" simplifies experiments and allows generalization.

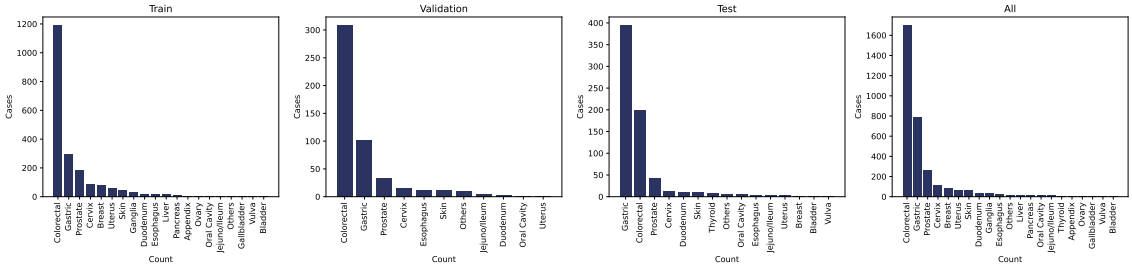

Figure 3.3: Distribution of cases by organ type for train, validation, test, and all sets.

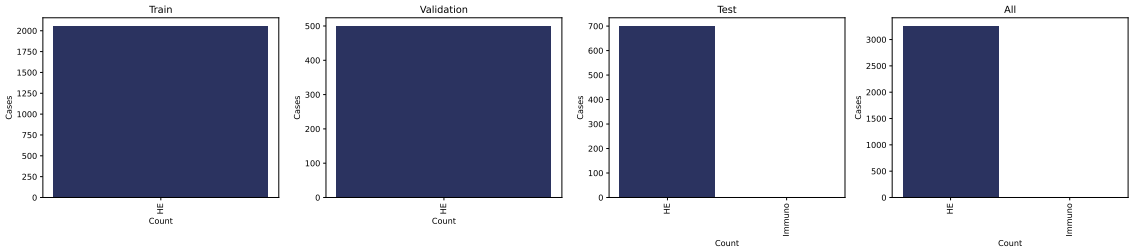

Figure 3.4: Distribution of cases by staining technique for train, validation, test, and all sets.

Since there is no direct counting annotation for sets, the ground truth used during training and validation for the set counting task is gathered from counting the number of sets identified in the spatial annotations, as they are labeled with '1'. The distribution of the dataset's fragments per set and set counts are shown in Figures 3.5 and 3.6, respectively.

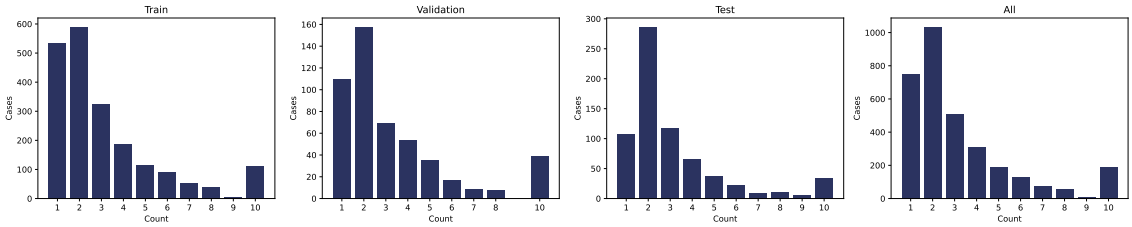

Figure 3.5: Distribution of cases by their fragment per set count for train, validation, test, and all sets.

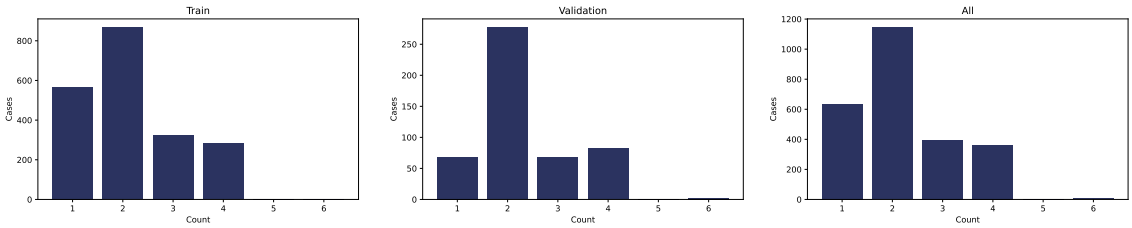

Figure 3.6: Distribution of cases by their set count for train, validation, and both sets.

As explained previously in 2.3.2.1, the representation of fragments and sets in the image directly affects their count. Various fragments per set are cases where the number of fragments per set is equal to or higher than 10, according to pathology technicians, so this is the highest value

that our models consider when counting. The number of sets equal to or above 6 follows the same logic. The distribution shown aligns with clinical routine samples and reflects an average distribution count in a diagnostics laboratory.

An exceptional case is not mentioned in Section 2.3.2.1. In some instances, multiple copies of the same set do not match, as one or more sets have a slightly different number of fragments, usually just one more fragment. In cases like those in Figure 3.7, the "gold standard" ground truth for fragments per set instinctively should be 2.5 and 1.5, respectively. In these cases, pathology technicians count as if a fragment was cut from the set with a fragment missing, so the number of fragments per set is modified to 3 and 2. Due to how we approach the problem, we only consider the integer ground truths in these cases.

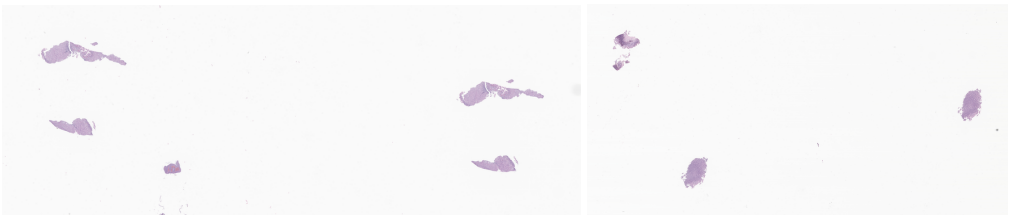

Figure 3.7: Samples which have a different number of fragments in each set. The ground truth considered for each case is 3 and 2, respectively.

3.3 Methodology

Since our objective is to improve results for both detection and counting, the methodology we propose is divided into a detection task and a counting task. In Section 3.3.1, we detail our methods for tackling the detection task, and in Section 3.3.2, our approach for handling the counting task.

3.3.1 Detection

With access to a dataset of histopathology images spatially annotated with bounding boxes, each labeled with '0' for fragments and '1' for sets, the natural step is to apply detection models that can learn to identify the bio-structures in new unlabelled data.

Considering the results previously obtained by Albuquerque et al. [9], we first replicate their best results using their original dataset to identify common errors in their proposed model. The cases described in Section 3.1, where the model failed to detect fragments or sets correctly, were gathered from this initial review. Considering this, we finetune the same best-performing model, YOLOv5, but using the extended dataset, which more than doubles the number of samples. Additional data benefits training as the model is exposed to more samples and, in our case, also more diverse, increasing robustness. This helps us gain perspective on the impact of the extended dataset on the model.

Different authors proposed iterations that improve the original YOLO model, briefly described in Section 2.2.2.2, such as YOLOv5 [117]. YOLOv5 improves upon the original YOLO by introducing enhancements like better network architectures, enhanced training techniques, and more efficient data augmentation strategies. These improvements help achieve higher accuracy and faster inference times than previous versions. The latest iteration is YOLOv9 [122], a state-of-the-art detection model that further builds on previous YOLO versions' advancements by addressing data loss challenges during deep network transmission. YOLOv9 introduces the concept of programmable gradient information (PGI) to retain complete input information, ensuring reliable gradient updates. Additionally, it employs a new lightweight architecture, the Generalized Efficient Layer Aggregation Network (GELAN), which optimizes parameter utilization and enhances performance. Hence, we finetune YOLOv9 using the extended dataset since it generally achieves better benchmarks in standard datasets.

All in all, we consider these two models to detect individual fragments and sets. We compare the methods and apply 5-fold cross-validation to evaluate further and consolidate results.

3.3.2 Counting

Pathology technicians verify that mounted slides match their macroscopic lab reports by checking the number of fragments per set. Counting sets helps confirm the number of bio-material repetitions on each slide, although this is not the primary task. Therefore, the counting methods we propose address both tasks: counting fragments per set and counting sets.

We tackle this problem using two distinct approaches: first, by deriving counts directly from detections provided by the detection models, and second, by reframing the task as a classification problem, wherein each image is assigned labels corresponding to the number of fragments per set and the number of sets present. We compare all the models and evaluate both approaches thoroughly, applying 5-fold cross-validation to consolidate findings. We also further analyze the best-performing model by gathering counting metrics across distinct dataset characteristics and by studying domain generalization.

3.3.2.1 Deriving Counts from Detections

After finetuning the detection models, we begin inference to get detection results and then count the number of sets and fragments detected from the predicted labels. To gather the fragments per set prediction, we calculate the ratio between the number of detected fragments and the detected sets. Since the ratio might not always be whole, as the model can predict a non-divisible number of fragments by the number of sets, we split our analysis into two groups: firstly, only the cases where the model provided integer-based predictions (*confident* cases) and secondly, including also the cases where the model provided fractional-based predictions (*sensitive* cases). If the model is clinically integrated, these last cases are potential candidates for manual evaluation by pathology technicians, as they either indicate instances where the model wasn't sure about its prediction or ambiguous, complex samples. For this reason, we also included cases where the model predicted

0 fragments or sets in this group, as this indicates an error case. These cases can be used as a reject option.

3.3.2.2 Predicting Counts from Classifiers

Reframing the counting problem as classification helps simplify the approach, as the focus of the counting task does not require fragment or set localization but simply assigning counts for each image. We treat the problem as multiclass classification, where each count is considered a possible class. The general model architecture consists of a pretrained backbone feature extractor and two MLP classification heads, one that classifies the fragments per set and another that classifies sets. Each MLP has a single hidden layer and an output layer according to the number of classes for each count. As described in Section 3.2, fragments per set counts range from 1 to 10, and set counts range from 1 to 6, so the model considers 10 and 6 classes for fragments per set and sets, respectively, in the output layers. A single class prediction is extracted as the maximum predicted probability from the model output class vector, as the target labels represent a single ground truth count value. The target set counts are obtained from spatial annotations, and the target fragments per set counts are gathered from the "gold standard" counting annotations.

3.4 Experimental Setup

To demonstrate the application of the proposed methodology, we conduct an empirical evaluation of the models proposed for both detection and counting. This section outlines the key aspects of our approach, including the details of the implemented models and the evaluation method, highlighting the cross-validation and the domain generalization processes.

3.4.1 Models and Parameters

To elucidate the design choices and configurations that underpin our empirical evaluation, we discuss the specific architectures, parameter settings, and training protocols employed. All the experiments were developed with Python using the PyTorch library to build and evaluate models.

3.4.1.1 Detection Models

For YOLOv5, we use Utralytics YOLOv5 implementation, which is available at `https://github.com/ultralytics/yolov5` as it is regularly updated and maintained and provides helpful references for inference and finetuning the models with custom data. As for YOLOv9, we used the authors' implementation, which is available at `https://github.com/WongKinYiu/yolov9`. They use Ultralytics' YOLOv5 implementation as the base to implement the YOLOv9 architecture, so the training and inference procedures are similar for both implementations.

We train both models using a batch size of 32 with images of 512x512 pixels for 200 epochs, with an initial learning rate of 10^{-2} and a final learning rate of 10^{-3}. We use initial pretrained weights and then finetune the models using our data. We use the YOLOv5s model and the

YOLOv9-C model as the initial training point for finetuning YOLOv5 and YOLOv9, respectively. The hyperparameters chosen were based on the previous work by Albuquerque et al.[9] and by tuning according to empirical trial analysis. The hyperparameters were the same for both models for fair comparison.

The YOLOv5 and YOLOv9 implementations already provide precision, recall, and mAP@50 metrics for each class and their average across all classes. However, they do not explicitly give the IoU, even though it is computed to calculate the mAP@50 metric. Considering this, we altered the validation script to provide this metric for each class and its average across all classes.

To perform the counting task from the detection models, we adapt the inference script to count the fragments per set and sets as described in the methodology. The count values are only calculated after Non-Maximum Suppression (NMS) is applied on inference results to reject overlapping detections, with a confidence threshold of 0.25 and an IoU threshold of 0.45, with a maximum of 300 detections. All the models we use for evaluation correspond to those with the best validation loss during training.

3.4.1.2 Classification Models

To train the classification models for counting, we used three different pretrained feature extractor backbones: ResNet, ViT, and DINOv2.

The training process was similar for all three experiments, and the hyperparameters chosen were tuned using empirical analysis. We train all the models for 100 epochs with a learning rate of 10^{-3} and a batch size of 32. We use the AdamW optimizer with a 10^{-3} weight decay rate and a linear learning rate scheduler with a 10^{-3} warmup factor decaying by each epoch. We use the cross-entropy loss since the counting task is reframed as a multiclass classification problem. For a fair comparison with the detection models, it was logical for the input size of the images to be 512x512, but that was only possible for the ResNet model since the other two models had input size constraints, as they are transformer-based and rely on patches. Considering this, these are the particular configurations of each model:

ResNet: We use the PyTorch implementation of ResNet-18 architecture with pretrained weights as the backbone. The last fully connected layer is removed from the model, and all the other layers, except for the last basic block, remain frozen during training. The input image size we use is 512x512.

ViT: We use the PyTorch implementation of Vit-B\32 architecture with pretrained weights as the backbone. The pretrained model limits the image size to 224x224, which is the input size we use to train this model. The last fully connected layer is removed from the model, and all the other layers remain frozen during training.

DINOv2: We use the linear classification pretrained head of the DINOv2 model from Meta AI, dinov2_vitb14_lc, as the model backbone. The last fully connected layer is removed

from the model, and all the other layers remain frozen during training. Due to model constraints, the input image must have dimensions divisible by the patch size, which is 14, so the input image size we use is 504x504, as it is the highest value lower than 512, which is divisible by the patch size.

As described in the methodology, and after the backbone feature extractor, each model has two MLP classification heads, with one hidden layer: one that predicts fragments per set, with a final layer with ten output classes, and another that predicts sets, with a final layer with six output classes. All the models we use for evaluation correspond to those with the best validation loss during training.

3.4.2 Evaluation Process and Metrics

For the detection models, both YOLOv5 and YOLOv9 are evaluated on the validation set, as there are no spatial annotations for test images. Since they share a base implementation, both models' metrics are computed similarly. The precision, recall, mAP@50, and IoU are computed per class (fragment and set) and averaged to give overall results for the model performance.

As for the counting methods, the accuracy, mean absolute error, and f1-score are calculated for both fragments per set and set count predictions and averaged to give overall results for model performance in both tasks. The metrics are computed for the classification models, and the counts are derived from the detection models' inference results. For this last counting method, predictions for these cases are ceiled to the nearest integer when dealing with *sensitive* cases, as described in the methodology. Considering some cases have a different number of fragments in each set, the ceiling of the prediction value goes accordingly with how the pathology technicians approach these cases, as described in Section 3.2.

Since the sets' ground truth count can only be obtained from the spatial annotations, as explained in Section 3.2, the counting methods are evaluated on the validation dataset. Nonetheless, the fragments per set ground truth value is available for the test dataset, so the fragments per set are also evaluated with test data besides the validation. This ensures more reliable results, as test cases were not used to tune hyperparameters or draw conclusions during empirical trials. There is also a higher and more diverse number of test samples, which benefits the robustness of the results in the same regard.

3.4.2.1 Cross-Validation

To further validate findings, we apply a 5-fold cross-validation procedure. We use the training and validation images to split the data into five different folds with a randomized distribution, with three folds having 510 images and the others 512. We train the model using four folds and validate them using the remaining fold, repeating this process for all folds. We evaluate the robustness of the model by averaging the results for the validation across all metrics. We apply cross-validation to both the detection and counting tasks, and all the models are trained using the same parameters described previously.

3.4.3 Domain Generalisation

We apply techniques to increase (in-domain) generalization during model training, like data augmentation and model regularisation methods. Nonetheless, verifying if the model stands against diverse domains is crucial since, if clinically integrated, it might be presented with unforeseen samples of various shapes, sizes, and characteristics, and it is expected to predict their fragments per set count correctly.

To study how well the best-performing model generalizes to new domains, we initially analyze counting metrics for fragments per set across each specimen, organ, and staining technique type using the test dataset. This helps gauge which sample characteristics are more challenging for the model to count and how the model behaves when presented with a sample from an unforeseen domain, since some images on the test set have different organ and staining technique types from the images on the training set. These unprecedented samples already present in the test set are Thyroid and Immunostaining cases. As there are only 8 Thyroid and 2 Immunostaining samples in the test set and ten samples are insufficient to assess how well the model generalizes the domain, we train the best-performing model using fewer samples, removing the cases where the organ is Prostrate, Cervix, and Breast and place this samples in the test set. By doing this, we can analyze the reliability and robustness of the model in predicting fragments per set counts of types of samples it has not been trained on. We train the model using the same parameters as described previously. We chose to remove the Prostate, Cervix, and Breast samples as they have high representation in the training set (third, fourth, and fifth most common, respectively) and because they are hard, ambiguous samples. Examples of samples of these organ types are present in Figure 3.8.

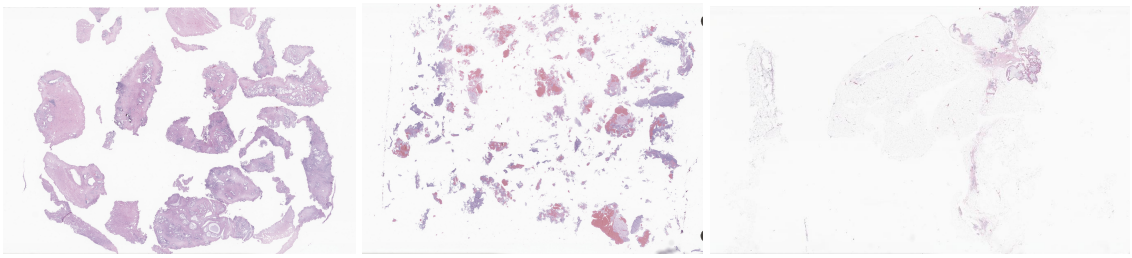

Figure 3.8: Examples of Prostrate, Cervix, and Breast organ samples, respectively.

After removing the training and validation samples and placing them in the test set, the dataset distribution is as follows: 1700 training images (187 prostrate, 86 cervix, and 80 breast samples removed), 450 validation images (33 prostate and 16 cervix samples removed) and 1103 test images.

3.5 Results

In this section, we analyze the results obtained from the experiments conducted. We examine the results in two parts: firstly, by evaluating the performance of the detection models and secondly,

by reviewing the results of the various counting methodologies. Finally, we discuss the overall impact of the experiments' outcomes and the methodology's limitations.

3.5.1 Detection

In Table 3.1, we present the results of each applied detection model and the best-performing results obtained by Albuquerque et al. [9] in their previous experiment to use as a baseline comparison. The best results for each metric are shown in bold.

Table 3.1: Comparison of the detection models metrics, including the previous research results.

	Fragment				Set				All			
Model	**IoU**	**P**	**R**	**map@50**	**IoU**	**P**	**R**	**map@50**	**IoU**	**P**	**R**	**map@50**
YOLOv5 (Albuquerque et al.[9])	0.93	**0.941**	**0.940**	**0.970**	0.920	0.957	0.969	0.985	0.925	0.949	**0.955**	**0.977**
YOLOv5	0.925	0.898	0.907	0.930	0.942	0.966	0.976	0.987	0.934	0.942	0.959	0.745
YOLOv9	**0.939**	0.940	0.917	0.959	**0.950**	**0.986**	**0.992**	**0.993**	**0.944**	**0.963**	0.954	0.976

The results obtained for both detection models are on par with the results of the baseline experiment. However, the baseline generally surpasses the new detection models in the fragment detection task, with a slightly noticeable difference for recall and map@50. Since the dataset we trained contains more than twice as many images as the previously used dataset, the examples introduced could represent more complex, ambiguous cases. Hence, the performance for this task decreases. This makes sense when considering set detection performs better for both YOLOv5 and YOLOv9 when compared with the baseline since it is a more straightforward task, and increasing samples directly improved these results. Generally, YOLOv9 is the best-performing detection model with impressive results across all metrics.

Table 3.2: Comparison of the detection models metrics with 5-fold cross-validation.

	Fragment				Set				All			
Model	**IoU**	**P**	**R**	**map@50**	**IoU**	**P**	**R**	**map@50**	**IoU**	**P**	**R**	**map@50**
YOLOv5	0.929	0.903	0.932	0.948	0.921	0.966	0.973	0.983	0.925	0.935	0.953	0.966
YOLOv9	**0.932**	**0.950**	**0.924**	**0.961**	**0.945**	**0.979**	**0.983**	**0.992**	**0.938**	**0.964**	**0.954**	**0.976**

Considering cross-validation across five-folds, YOLOv9 still outperforms YOLOv5 across all evaluation metrics, and the results obtained are similar to the results obtained by training with the original data split, proving robustness and good generalization, as shown in Table 3.2.

To visually evaluate improvements over the baseline detections by Albuquerque et al. [9], Figure 3.9 shows the detections presented in the problem statement in Section 3.1, but now using the best-performing detection model, YOLOv9.

As shown, the model correctly detects the fragments and sets for cases a), c), and d): in a), the model no longer detects disconnected tissues inside the large inhomogeneous fragment; in c), the model no longer detects the artifact/negligible tissue as a fragment and in d) the model detects

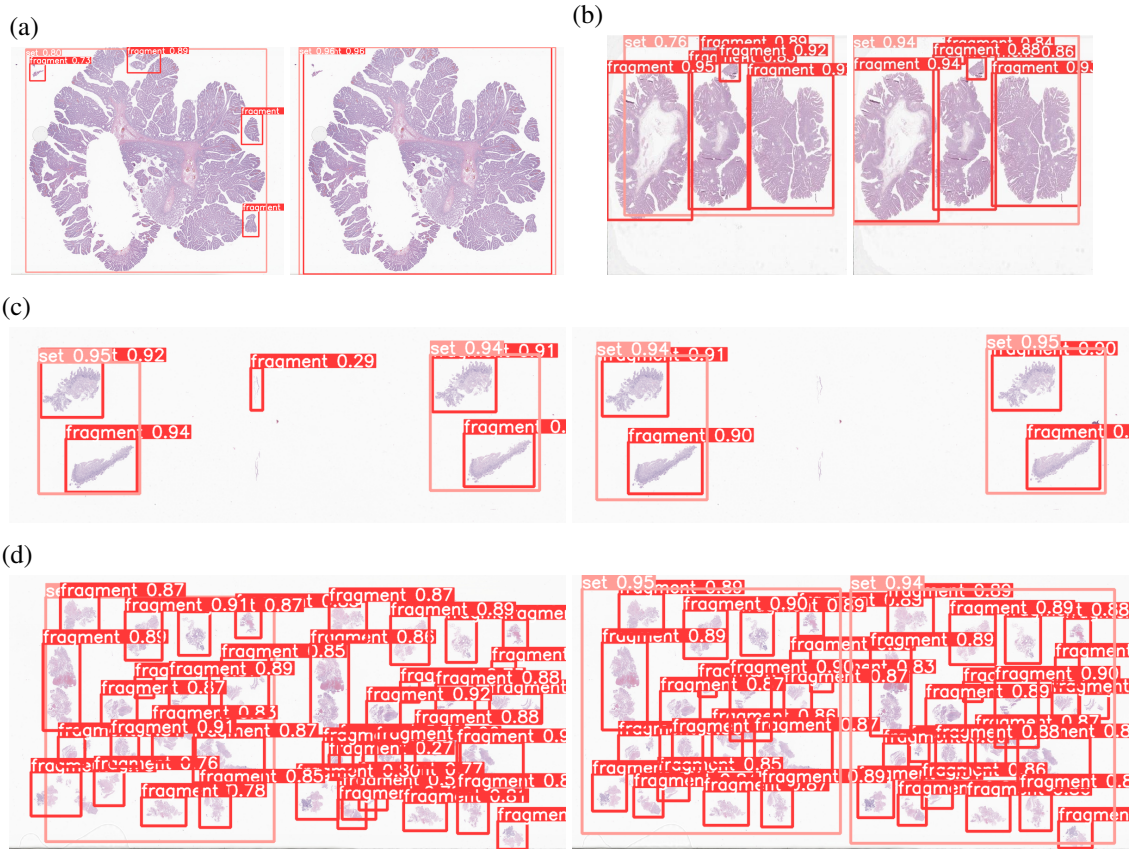

Figure 3.9: YOLOv9 detection results, on the right, in comparison to YOLOv5 detection results by [9], on the left, for the common errors experienced.

both sets, even with the slight difference in fragments. However, the model still fails to detect the middle fragment as a single fragment in case b), treating the disconnected tissues as independent fragments. This proves that the model still needs finetuning to account for all errors in detection.

3.5.2 Counting

As described in previous sections, the set count can only be evaluated using the validation dataset due to the dataset annotations. Since validation results may be skewed as they were used during experimental trials to tune hyperparameters empirically, assessing the methods on the test data is essential. However, as it only has counting annotations, it can only be used for the fragments per set task.

To evaluate which counting method is better at predicting the number of fragments per set and sets present in a histopathology image, we compare all of them across the defined evaluation metrics. The methods that derive counts from detection models have their results for only the *confident* cases and both *confident* and *sensitive* cases, as explained in Section 3.3.2.1. The best results per metric are highlighted in bold.

3.5.2.1 Using the validation set

In Table 3.3, we present the results for all the counting methods proposed using the validation set: the counting values derived from the detections inferred by the detection models, YOLOv5 and YOLOv9, and the counting values from the predictions of the classification models.

Table 3.3: Comparison of the counting methods metrics using the validation set.

	Fragments per Set			**Set**			**All**			**Total Cases**		**Sensitive Cases**
Model	**A**	**mae**	**f1**	**A**	**mae**	**f1**	**A**	**mae**	**f1**			
YOLOv5	0.847	0.214	0.680	0.922	0.095	0.716	0.884	0.155	0.699	C	359	140 (28%)
	0.687	0.471	0.489	0.814	0.232	0.551	0.751	0.352	0.510	C+S	499	
YOLOv9	**0.914**	**0.120**	**0.734**	**0.982**	**0.023**	**0.982**	**0.948**	**0.071**	**0.751**	C	441	58 (12%)
	0.858	0.202	0.673	0.954	0.062	0.903	0.906	0.132	0.694	C+S	499	
resnet_18	0.723	0.425	0.473	0.922	0.086	0.600	0.823	0.256	0.518	-	-	-
vit_b32	0.687	0.527	0.496	0.860	0.162	0.546	0.774	0.345	0.533	-	-	-
dinov2	0.661	0.573	0.479	0.890	0.118	0.569	0.776	0.346	0.538	-	-	-

YOLOv9 yields the best metrics, including for the *confident* and *confident+sensitive* cases. Compared with YOLOv5, the other detection-based approach, YOLOv9 outperforms it by far when considering all predictions (*confident+sensitive*). Even though the detection results for YOLOv5 were not far from YOLOv9's results, the model is not as good at counting fragments per set in the image as it is at localizing fragments and sets individually. Since it identifies about 28% of cases as *sensitive*, the model is most likely identifying more structures than it should, even though the structures it identifies are generally correctly placed. YOLOv9, on the other hand, only identifies about 12% of cases as *sensitive*, and even when they are considered for evaluation alongside the *confident* cases, it is still the best-performing method. Nevertheless, the performance still decreases when considering *sensitive* cases, meaning the model still detects some structures as fragments or sets when it should not.

The classification models' results are generally lower than those obtained by deriving counts from detections. The simple classifier architecture might not hold enough expressive power to classify samples correctly. Finetuning the totality of the backbone feature extractors required more data since it quickly led to overfitted models. The features extracted could not be specific enough for this task as they were not explicitly trained on medical imaging or histopathology samples, which could potentially improve results. From the classification models, the one with the ResNet18 backbone was the model with the best performance. The small amount of training data, when compared with the one used to train the more complex ViTB32 and DINOv2 models, is better suited to finetune a simpler model, such as ResNet-18, as complex models quickly led to overfitting without cohesive finetuning strategies. Notwithstanding, the results were competitive with the YOLOv5 evaluated with *confident+sensitive* cases.

As with the detection models, the counting methods perform worse for counting fragments per set than for counting sets. Realistically, counting fragments per set is a more challenging task due

to fragment variability, even causing ambiguity among professionals in some complicated cases. As for counting sets, since they are usually repetitions of a group of fragments, it is easier for methods to model this relationship (except in some rare cases where sets have a different number of fragments, typically a tiny fragment more between sets). Because of this, methods are expected to have better results for this task than for counting fragments per set.

Table 3.4: Comparison of the counting methods metrics with 5-fold cross-validation using the validation set.

	Fragments per Set			**Set**			**All**			**Total Cases (average)**		**Sensitive Cases (average)**
Model	**A**	**mae**	**f1**	**A**	**mae**	**f1**	**A**	**mae**	**f1**			
YOLOv5	0.784	0.317	0.599	0.924	0.102	0.679	0.854	0.210	0.628	C	365	146 (29%)
	0.644	0.473	0.452	0.813	0.232	0.518	0.729	0.352	0.481	C+S	511	
YOLOv9	**0.876**	**0.187**	**0.703**	**0.976**	**0.034**	**0.945**	**0.926**	**0.110**	**0.723**	C	459	52 (10%)
	0.828	0.261	0.630	0.951	0.069	0.739	0.890	0.165	0.652	C+S	511	
resnet_18	0.692	0.502	0.438	0.935	0.076	0.688	0.813	0.289	0.493	-	-	-
vit_b32	0.640	0.620	0.384	0.850	0.169	0.609	0.745	0.395	0.430	-	-	-
dinov2	0.628	0.656	0.351	0.889	0.125	0.638	0.759	0.391	0.412	-	-	-

The cross-validation results in Table 3.4 validate these findings, as YOLOv9 still stands as the best-performing method. All the models have slightly lower results on average, likely due to some variations in sample complexity between folds.

3.5.2.2 Using the test set

Generally, results evaluated using the test set strengthen the previous section's observations, as shown in Table 3.5. YOLOv9 remains the best-performing model, even when considering *confident+sensitive* cases, contrary to YOLOv5. While lacking accuracy, it does not have the lowest f1-score and mean absolute error, which proves that even when not predicting cases correctly, those errors are not as pronounced in other models. The classification models are still behind the detection-based counting methods, although the model with the ResNet18 was competitive with YOLOv5, yielding better results when considering *confident+sensitive* cases.

Table 3.5: Comparison of the counting methods metrics using the test set.

	Fragments per Set			**Total Cases**		**Sensitive Cases**
Model	**A**	**mae**	**f1**	**Cases**	**Sensitive**	
YOLOv5	0.790	0.265	0.678	C	476	225 (32%)
	0.625	0.449	0.472	C+S	701	
YOLOv9	**0.929**	**0.089**	**0.893**	C	662	39 (6%)
	0.904	0.128	0.854	C+S	701	
resnet_18	0.776	0.333	0.466	-	-	-
vit_b32	0.719	0.404	0.450	-	-	-
dinov2	0.650	0.516	0.399	-	-	-

The test set results provide higher confidence in the models developed as test samples are held out from the training/validation process and have a distinct distribution in terms of structural characteristics, namely specimen, organ, and staining technique type. It would be beneficial to have counting annotations for set counts so that set count results could also be evaluated using this data. This would ensure model robustness for both tasks, not only the fragments per set count.

3.5.3 Domain Generalisation Analysis

To verify robustness and adequate generalization across domains, we first review the results of the best-performing counting method, YOLOv9, independently by each sample's specimen, organ, and staining technique types. The results analyzed are for the fragments per set counting task and are evaluated on the test set.

Table 3.6 presents the results concerning the specimen type. The model has a lower performance on Polypectomy specimens, which makes sense considering those samples are not as frequent during training, as shown in Figure 3.2. The distribution of *sensitive* cases follows the distribution of specimen types frequency in the test dataset, which is expected, considering more samples introduce higher variability.

Table 3.6: Results of the YOLOv9 counting method by specimen type.
(C - *confident* cases; S - *sensitive* cases)

Specimen Type	Total Cases		A	mae	f1	Sensitive Cases
Biopsy	C	464	0.942	0.063	0.892	27
	C+S	491	0.921	0.096	0.850	
Surgical Specimen	C	11	0.909	0.091	0.600	1
	C+S	12	0.917	0.083	0.667	
Polipectomy	C	145	0.876	0.186	0.850	8
	C+S	153	0.843	0.242	0.781	
Biopsy + Polipectomy	C	1	1.000	0.000	1.000	0
	C+S	1	1.000	0.000	1.000	
Surgical Specimen + Polipectomy	C	1	1.000	0.000	1.000	1
	C+S	2	1.000	0.000	1.000	
TUR	C	8	1.000	0.000	1.000	0
	C+S	8	1.000	0.000	1.000	
Biopsy or Polipectomy	C	32	0.969	0.063	0.964	2
	C+S	34	0.912	0.147	0.848	

The results regarding each organ are shown in Table 3.7. Uterus and Oral Cavity samples show the worst performance of all organs, likely because they are more challenging cases. An example of a detection of a Uterus sample worth analyzing is present in Figure 3.10. As is shown, the model fails to detect one of the fragments at the bottom of the image because it is partially cut, making the count of fragments per set not whole, thus a *sensitive* case. However, since the count is above 10, it is still correct, which indicates that these cases are worth considering even if the detector partially fails.

The model performs well on Thyroid samples, even though it is not trained in cases of this organ. Most of these cases are large homogeneous fragments, exemplified in Figure 3.11, which the model is comfortable detecting and counting. Prostate samples are the third most common

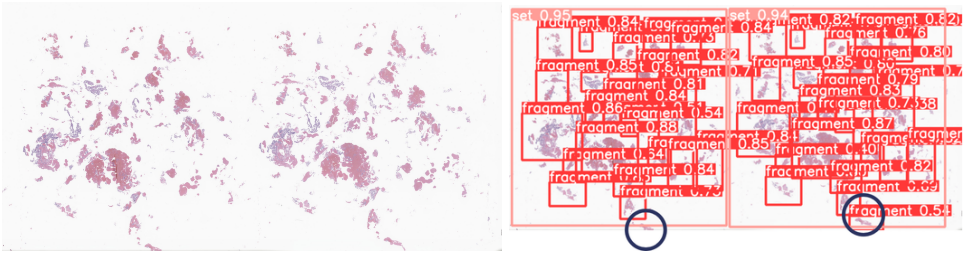

Figure 3.10: Detection of a Uterus sample. The model fails to detect the circled fragment in the left set, which corresponds to the circled fragment in the right set.

sample in the test set, and their performance is lower than that of other standard samples, especially for the f1-score. As mentioned, these are harder, ambiguous samples, which may hinder the model's learning process. Consequently, the imbalance between precision and recall in counting is more significant. The model performs well when counting Gastric samples, even though these are not the most common samples during training, proving good generalization. Generally, this is also true for most of the other organ types.

Table 3.7: Results of the YOLOv9 counting method by organ type.
(C - *confident* cases; S - *sensitive* cases)

Organ	Total Cases		A	mae	f1	Sensitive Cases
Gastric	C	373	0.957	0.043	0.918	21
	C+S	394	0.939	0.076	0.873	
Breast	C	1	1.000	0.000	1.000	0
	C+S	1	1.000	0.000	1.000	
Prostate	C	40	0.875	0.125	0.467	3
	C+S	43	0.837	0.163	0.489	
Duodenum	C	11	0.909	0.182	0.926	0
	C+S	11	0.909	0.182	0.926	
Jejuno/Ileum	C	3	1.000	0.000	1.000	0
	C+S	3	1.000	0.000	1.000	
Colorectal	C	189	0.878	0.175	0.872	11
	C+S	200	0.845	0.220	0.829	
Bladder	C	1	1.000	0.000	1.000	0
	C+S	1	1.000	0.000	1.000	
Esophagus	C	3	1.000	0.000	1.000	0
	C+S	3	1.000	0.000	1.000	
Cervix	C	11	1.000	0.000	1.000	1
	C+S	12	0.916	0.166	0.829	
Uterus	C	2	0.500	1.000	0.333	1
	C+S	3	0.667	0.667	0.500	
Skin	C	11	1.000	0.000	1.000	0
	C+S	11	1.000	0.000	1.000	
Oral Cavity	C	4	0.750	0.250	0.778	1
	C+S	5	0.600	0.600	0.542	
Vulva	C	1	1.000	0.000	1.000	0
	C+S	1	1.000	0.000	1.000	
Thyroid	C	8	1.000	0.000	1.000	0
	C+S	8	1.000	0.000	1.000	
Others	C	4	1.000	0.000	1.000	1
	C+S	5	1.000	0.000	1.000	

Regarding the staining technique, the model is not trained with immunostained samples, but

it correctly counts the number of fragments in all test images of this type, implying good generalization. This is likely due to the color jitter in the data augmentation during the detection model training. An example of a sample using this staining technique is shown in Figure 3.11. However, only two immunostaining samples are in the test dataset, so the model should be tested on more samples to draw firm conclusions.

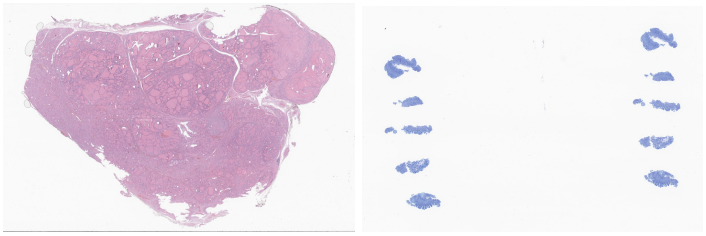

Figure 3.11: Thyroid and Immunostaining samples, respectively.

Table 3.8: Results of the YOLOv9 counting method by staining technique type. (C - *confident* cases; S - *sensitive* cases)

Technique	Total Cases		A	mae	f1	Sensitive Cases
Hematoxylin and Eosin Stain (HE)	C	660	0.929	0.089	0.891	39
	C+S	699	0.904	0.129	0.851	
Immunostaining	C	2	1.000	0.000	1.000	0
	C+S	2	1.000	0.000	1.000	

3.5.3.1 Domain Withdrawal

Since many organs are only tested on a few images, no substantial evidence can be derived from the effectiveness of the results presented above for these cases. Proving good generalization is crucial for cementing the robustness and reliability of the YOLOv9 counting method. For this, we re-analyze the results for the counting method by using specimen, organ, and staining technique types when the Prostate, Cervix, and Breast samples are removed during training. The removed samples from the train and validation sets are grouped in the test set and used to compute the results.

Table 3.9 shows results by specimen type for the fragment per set counting task when considering the YOLOv9 counting method with the removed samples. Overall, the performance is lower than when considering the complete dataset for training, but the difference in performance is not accentuated. The model with the removed samples performed slightly better in some cases, like for Polypectomy samples, as an added case during evaluation correctly predicted this type. Most of the Prostate, Cervix, and Breast samples removed were Surgical Specimens and Biopsies, which increased the test dataset with an additional 230 and 153 test samples, respectively. This increase and the simultaneous decrease in training samples did not significantly affect the model's performance. TUR's performance suffered with the removal of samples since almost all the TUR

samples were Prostate cases, which the model did not have access to during training, leading to worse performance for this specimen type.

Table 3.9: Results of the YOLOv9 counting method by specimen type, using the model trained without Prostate, Cervix or Breast samples.
(C - confident cases; S - sensitive cases)

Specimen Type	Total Cases		A	mae	f1	Sensitive Cases
Biopsy	C	588	0.871	0.173	0.790	56
	C+S	644	0.825	0.225	0.682	
Surgical Specimen	C	238	0.849	0.160	0.545	4
	C+S	242	0.839	0.182	0.542	
Polipectomy	C	145	0.883	0.228	0.751	9
	C+S	154	0.864	0.240	0.776	
Biopsy + Polipectomy	C	0	-	-	-	1
	C+S	1	1.000	0.000	1.000	
Surgical Specimen + Polipectomy	C	2	1.000	0.000	1.000	1
	C+S	3	1.000	0.000	1.000	
TUR	C	23	0.652	1.174	0.132	1
	C+S	24	0.625	1.160	0.113	
Biopsy or Polipectomy	C	34	0.912	0.089	0.711	1
	C+S	35	0.886	0.143	0.669	

It is worth noting that, even if not providing *confident* predictions for all or some cases, as in the Biopsy+Polypectomy and Surgical Specimen+Polypectomy types, respectively, the estimation of the counting values is still correct. The same happens when the model is trained using the entire dataset, even if it is more difficult to verify. This proves that predictions are often close to the ground truth, and approximating that result by ceiling yields correct counting predictions.

The results for the fragment per set counting task by organ type when considering the YOLOv9 counting method with the removed samples are shown in Table 3.10. Organs like the Jejuno/Ileum, Esophagus, Skin, Vulva, and Thyroid maintain their performance even if trained with fewer and less challenging samples, which indicates good generalization for these cases. Still, more test samples are needed to draw more decisive conclusions, as all these cases are not significantly represented on the test set. Adding to this conclusion, although slightly lower, Gastric, Duodenum, and Oral cavity samples also maintained their performance. Gastric samples, in particular, are the most common samples in the test set, and the performance is similar to that of using the complete dataset for training, meaning robust generalization for the counting method.

Interestingly, Colorectal and Uterus samples perform better using this model. This suggests that increasing the variability and difficulty of the samples affects each organ type differently, and patterns learned by some organ types may benefit the detection and counting of some cases more than others do. In contrast, Breast, Prostate, Cervix, Bladder and Others performance decreased during training, likely due to reduced representation of these complex cases during training. The lack of challenging cases hindered detection, as Prostate and Cervix have many *sensitive* cases.

The results for the fragment per set counting task by staining technique type when considering the YOLOv9 counting method with the removed samples are shown in Table 3.11. The performance of most samples decreased, likely due to the reasons pointed out previously. The model

Table 3.10: Results of the YOLOv9 counting method by organ type, using the model trained without Prostate, Cervix, or Breast samples.
(C - confident cases; S - sensitive cases)

Organ	Total Cases		A	mae	f1	Sensitive Cases
Gastric	C	383	0.953	0.047	0.889	11
	C+S	394	0.937	0.063	0.867	
Breast	C	81	0.790	0.210	0.515	0
	C+S	81	0.790	0.210	0.515	
Prostate	C	231	0.745	0.407	0.414	32
	C+S	262	0.703	0.445	0.392	
Duodenum	C	10	1.000	0.000	1.000	1
	C+S	11	0.909	0.091	0.911	
Jejuno/Ileum	C	3	1.000	0.000	1.000	0
	C+S	3	1.000	0.000	1.000	
Colorectal	C	187	0.882	0.198	0.855	13
	C+S	200	0.855	0.225	0.840	
Bladder	C	0	-	-	-	1
	C+S	1	0.000	0.500	0.000	
Esophagus	C	3	1.000	0.000	1.000	0
	C+S	3	1.000	0.000	1.000	
Cervix	C	102	0.794	0.314	0.579	12
	C+S	114	0.737	0.395	0.482	
Uterus	C	2	1.000	0.000	1.000	0
	C+S	3	1.000	0.000	1.000	
Skin	C	11	1.000	0.000	1.000	0
	C+S	11	1.000	0.000	1.000	
Oral Cavity	C	4	0.750	0.500	0.667	1
	C+S	5	0.600	0.800	0.625	
Vulva	C	1	1.000	0.000	1.000	0
	C+S	1	1.000	0.000	1.000	
Thyroid	C	8	1.000	0.000	1.000	0
	C+S	8	1.000	0.000	1.000	
Others	C	4	0.500	0.750	0.200	1
	C+S	5	0.400	0.800	0.133	

still correctly predicts immunostaining samples, but the issue of confidence in these predictions remains the same as that of the model trained with the complete dataset.

Table 3.11: Results of the YOLOv9 counting method by staining technique, using the model trained without Prostate, Cervix, or Breast samples.
(C - confident cases; S - sensitive cases)

Technique	Total Cases		A	mae	f1	Sensitive Cases
Hematoxylin and Eosin Stain (HE)	C	1028	0.864	0.197	0.773	73
	C+S	1101	0.831	0.235	0.677	
Immunostaining	C	2	1.000	0.000	1.000	0
	C+S	2	1.000	0.000	1.000	

Overall, this analysis highlights the importance of having challenging cases during training, as this drives the detection model to learn better representations and, in turn, provide more accurate counting results. Nonetheless, the lower performance when removing these cases could be attributed to a decrease in the training sample volume of 383 images, as fewer images during training generally equate to a reduced performance in every case. Following an incremental withdrawal

approach in the analysis, removing a domain independently and testing on the others could solve this issue, as the analysis will be separated, and the number of training samples will not decrease significantly in each removal. This would also allow us to verify which sample type affects the performance of results.

Still, even if lower, the performance for this task surpasses the other proposed counting methods, so the generalization quality can generally be ensured in comparison.

3.5.4 Discussion

From the conducted experiments, we can confidently say that we improved the results of the detection and counting of fragments and sets in histopathology images. The best-performing method, YOLOv9, effectively improved detection results and reliably counted both fragments per set and sets.

Regarding the detection task, the experiments could benefit from exploring other detection models, primarily state-of-the-art transformer-based detectors, to compare and challenge the results of the YOLOv9 detection models. With good detection results, more counting predictions could be derived with hopefully comparable or even better performance.

As we proved good generalization strength of the best-performing YOLOv9 model, it would be wise to gather more test data, especially for underrepresented specimen, organ, or staining types, to analyze further how the model behaves in each circumstance. This is crucial for potential clinical integration, as the model must provide reliable results for each unforeseen case it is presented with. Moreover, increasing the training data would also guarantee that the model has access to more diverse samples when training, likely providing better results if that were the case. Another problem with the data is the lack of spatial annotations in the test set that do not allow the computation of metrics for the set count task using the test set, which is crucial for an unbiased evaluation.

The separation between *confident* and *sensitive* cases serves as a good rough starting point for a reject option approach, as *sensitive* cases only constitute about 10% of the samples for the YOLOv9 counting method, which is a fair number of samples to be considered for manual evaluation by pathology clinicians in routine practice. Nonetheless, the *confident* predictions can still be improved, and considering all those cases as correct will still lead to some error margin, particularly for the fragments per set count. Imposing restrictions on the counts and predictions based on structural and hierarchical relationships between fragments and sets could narrow this error margin and gather a more robust subset of *confident* predictions. This would increase the number of rejected samples, proving it worth exploring other counting approaches that improve these results, especially for the fragments per set count. Hence, pathology technicians have to evaluate the least amount of samples manually.

Chapter 4

Exploring Graph-Based Learning for Fragment Counting

This chapter analyzes the viability of graph-based learning approaches as counting methods. Section 4.1 exposes our motivation for this approach and the proposed methodology. Details of the experiments applied are described in Section 4.2, and the analysis of the results obtained is present in Section 4.3.

4.1 Graph Neural Networks for Fragment Counting

The relationship between fragments and sets in histopathology images is intricate, as the representation of fragments directly affects the representation of sets and vice-versa. Pathology technicians count fragments per set in samples to cross-check with macroscopic reports as a quality control procedure, so any model that performs this task must provide reliable results for counting fragments per set. The experiments conducted in Chapter 3 provide a robust methodology for counting sets and fragments per set through a reliable detection model, YOLOv9. Nonetheless, there could be room for improvement, particularly for fragments per set, as they are the target task and have lower performance when compared with set counting. Considering this, we found it pertinent to explore graph-based approaches to focus on the nuances of the spatial and structural relationships between fragments and sets. With this objective, we propose a methodology for counting fragments per set in histopathology images using GNNs, described generically in Figure 4.1.

4.1.1 Graph Data Structure

We approach the task as a graph classification problem, where each sample is defined by a complete graph with a corresponding target label of fragments per set, given by the ground truth counting annotations, as in the previous methods. Fragments are detected and cropped from the sample images, and features are extracted from each cropped fragment to define the graph nodes. All fragments are connected, and the edge weights are defined as the normalized Euclidean distance between the cropped fragment's centroids. We consider a complete graph to capture the global

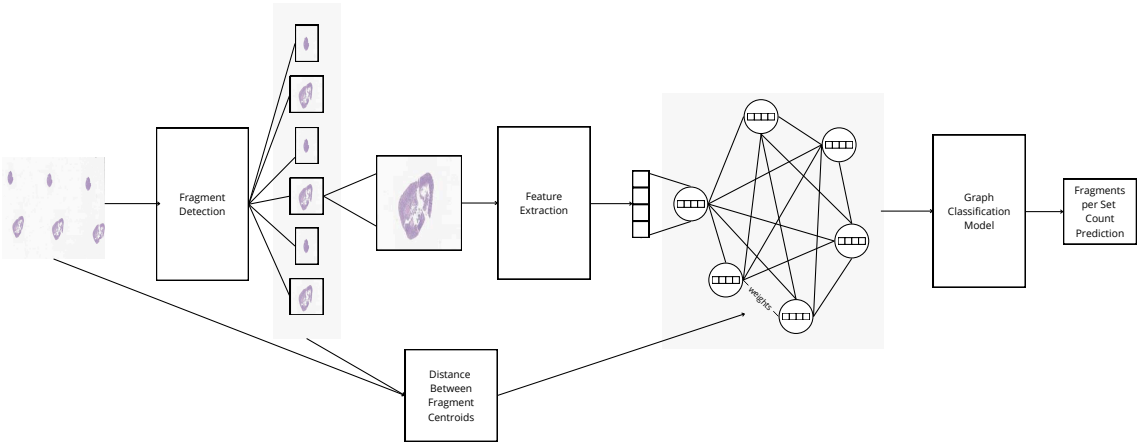

Figure 4.1: The graph classification method for fragments per set counting.

context between fragments and different sets, which is crucial to modeling the spatial relationship that separates fragments by the set they belong to.

4.1.2 Fragment Detection Methods

During training, the fragments detected are cropped from the spatial bounding box annotations as a simplification to minimize error sources. This guarantees that the errors inherent to detection using another method do not affect the training process. Logically, this is not viable during evaluation, so we consider two strategies to detect the fragments: connected component analysis or the YOLOv9 detections. For the connected component analysis, we consider the components whose areas are above 5% of the area of the biggest component computed from the binarised and OTSU thresholded sample images. These components are then cropped around their centroids to a fixed resolution. For the YOLOv9 detections, we perform inference using the YOLOv9 detection model described in the previous chapter and use each detected fragment's bounding box coordinates to crop the sample images.

4.1.3 Fragment Feature Extraction

After cropping fragments, we also follow two strategies to extract fragment features: handcrafted features and features extracted by a contrastive-based model.

For the handcrafted features, we compute the 25th percentile distance between every point in the objects' contours, their normalized bounding box area, and their circularity. These features are selected based on theoretical knowledge, as they describe the shape and size of blob-like structures such as fragments.

For the contrastive-based approach, we train a standard convolutional network architecture using triplets - anchor, positive, and negative - to extract features from the fragment crops that encode the relationship between fragments in different sets. We build this model based on the intuition that sets are similar groups of fragments, repeated through the image, so corresponding fragments

from different sets should have similar features and thus be represented closely in feature space. This helps extract structural information from the samples, which is essential for defining sets and, consequently, correctly counting the fragments per set.

However, some considerations about sample structure must be considered when building the model. We extract fragments from the bounding box spatial annotations and group them by their sets. These fragments are then used to build triplets and train the model. When slides are mounted, they can have different set configurations, as illustrated previously in Section 2.3.2.1, so each extracted fragment is an anchor, and the positive and negative pairs are built accordingly:

Multiple sets with multiple fragments: Sets are repetitions of groups of fragments in the sample image, so matches are considered as corresponding fragments from different sets. As such, the positive is randomly chosen from any matched fragments, and the negative is any other fragment that isn't a match from the same set or another.

There is an exception where the number of fragments in sets might have slight variations, a rare instance where those odd fragments have no match. In those cases, the negative is still randomly chosen from the unmatched fragments, but the positive is augmented from the anchor fragment.

Multiple sets with single fragment: Sets are repetitions of a single fragment in the sample image, so the positive is randomly chosen from any fragment from a different set. There are no negatives in this case, as all fragments are matches, so we choose a random fragment from another sample image as the negative.

Single set with multiple fragments: There is a single set, so there are no fragment repetitions and, consequently, no fragments to match. In this case, the positive is the augmented anchor fragment, and the negative is randomly chosen from the other fragments in the set.

Single set with single fragment: There is only one fragment and one set, so there is no direct positive or negative. In this case, we use the augmented anchor as the positive and a random fragment from another sample image as the negative.

This matching process is treated as a generalized assignment problem, where fragments are matched according to their bounding box area. Given that each fragment f in a set S has an area denoted by $A(f)$, the goal is to find pairs of fragments (f_i, f_j) from different sets that minimize the absolute difference in their areas, so the optimal assignment has a cost defined formally by:

$$\min \sum_{f_i \in S_i, f_j \in S_j} |A(f_i) - A(f_j)| X_{ij} \tag{4.1}$$

where $S_i = \{f_{i1}, f_{i2}, \ldots, f_{in_i}\}$ and X is the boolean assignment matrix where $X[i, j] = 1$ indicates matching between f_i and f_j. To solve this, we use SciPy's implementation of a modified Jonker-Volgenant algorithm [34] with no initialization. After computing the optimal matches, we group them by each fragment for creating triplets, as described above.

The triplets are then used to train a standard convolutional network comprised of four convolutional layers for feature extraction, followed by two fully connected layers for embedding projection. Each convolutional layer is paired with ReLU activation and max-pooling, with a final dropout layer to prevent overfitting. When graphs are built for the counting task, this model processes the cropped fragments given by the fragment detection methods, and the resulting lower-dimensional feature embeddings from the forward pass are used as node features.

4.1.4 Graph Classification Model

Using the graph representation of the sample images, we train a graph neural network model for graph classification, mirroring the classifier architecture described in the previous chapter. The backbone of the model consists of a single Graph Convolutional Network (GCN) layer that aggregates features based on the mean of neighboring nodes, as this captures the whole graph distribution, followed by a ReLu activation layer. A single GCN layer is sufficient since each graph is complete, so during message passing, all nodes aggregate information from all others. We use max pooling as the readout layer for coarsening the graph as it selects the nodes with the most representative features. After a dropout layer to introduce regularisation, an MLP classification head processes those features with a single hidden layer paired with ReLU activation and an output layer of 10 classes, as the task is to count fragments per set. A single class prediction is extracted as the maximum predicted probability from the model output class vector, as the target labels represent a single ground truth count value for each graph, gathered from the "gold standard" counting annotations.

4.2 Experimental Setup

To illustrate the application of the proposed graph counting method, we empirically evaluate its performance in the fragments per set counting task. This Section provides details of the implementation for both the contrastive-based feature extractor and the final proposed counting model, as well as the respective evaluation process and metrics used for both cases. As in the last chapter, all experiments were conducted using Python, leveraging the PyTorch library for model development and evaluation.

4.2.1 Contrastive-Based Feature Extractor

Before training, the fragments are cropped from the ground truth bounding box spatial annotations in each image. Still, as they vary significantly in size, we apply padding to a fixed resolution of 128x128 pixels. To maintain the most information in each fragment crop without losing much resolution, we resize the crops according to their aspect ratio to the largest resolution below 128 pixels in the x and y-axis and then pad to this fixed resolution. We experimented with 64x64,

256x256, and 512x512 pixels, but 128x128 was the resolution that offered the best tradeoff between computational complexity and image quality, cemented by the fact that the average fragment crop resolution is 80x90 pixels.

We train the model using a batch size of 32 for 100 epochs, with a learning rate of $5 \cdot 10^{-4}$. We use the Stochastic Gradient Descent optimizer with weight decay of 10^{-2} and 0.9 momentum and a learning rate scheduler that reduces the learning rate upon the model plateauing by the average training loss, with five epoch patience. We use triplet margin loss to force a distance between different pairings by a specified margin, which we set as 0.5. We experimented with other larger margin values but found better separation between anchor-positive and anchor-negative pairs with this value, as it is smaller and narrows the window of semi-hard negatives, improving learning. The model processes the fragment crops and extracts feature embeddings of dimension 32, which are then used to define node features in sample graphs.

4.2.1.1 Evaluation Process and Metrics

We evaluated the contrastive-based feature extractor by calculating the difference between the Euclidean distance between anchor-negative and anchor-positive pairs. A larger distance indicates better separation between negative and positive samples. We also computed accuracy, which we defined as the proportion of triplets for which the model correctly identifies the positive sample as closer to the anchor than the negative sample.

4.2.2 Graph Classification Model

We divided the graph classification model training according to the feature-extracting approach, either handcrafted or contrastive model-based. When using the handcrafted features, we use cropped fragments from the ground truth spatial annotations directly to compute those values. However, when using the contrastive model features, we need to apply the fixed padding resolution described in the Section above since 128x128 pixels is the only resolution accepted by the feature extractor model.

We train the handcrafted approach for 100 epochs and the contrastive model approach for 150 epochs, with a 10^{-3} learning rate and a batch size of 32. We use the AdamW optimizer with a 10^{-2} weight decay rate and a learning rate scheduler that reduces the learning rate upon the model plateauing by the average training loss, with five epoch patience. Similarly to the counting methods described in the last chapter, we use the cross-entropy loss as we treat the problem as a multiclass classification problem.

Regarding specific model parameters, we use a single GCN layer with 512 hidden channels for both approaches, as it performed better than other dimensions (64, 128, and 256). We consider the model with the best validation loss during training for evaluation with the test set.

4.2.2.1 Evaluation Process and Metrics

During the evaluation, when considering fragments detected from the connected component analysis, we used a patch size of 90 pixels to crop around their centroids, as this was the value with the best empirical performance. This is cemented by the fact that the average fragment bounding box size in the training data is 80x90 pixels. As mentioned before, when considering fragments inferred from the YOLOv9 detector, we cropped fragments considering their bounding box coordinates.

We conduct the experiments following the same guidelines as the other counting approaches analyzed in the last chapter but with some simplifications. We only evaluate this method using the test set, as the objective task of counting fragments per set can be evaluated using only numerical ground truth annotations. Also, no cross-validation is applied to validate results. The model is evaluated using the accuracy, mean absolute error, and f1-score and directly compared with our best-performing counting method, YOLOv9.

4.3 Results

In this Section, we present and discuss the results of the graph classification model explored for the fragments per set counting task. We also initially review the training process of the contrastive-based feature extractor model as it directly influences the analysis of the counting results that use this approach.

4.3.1 Contrastive-Based Feature Extractor

To choose the final contrastive-based feature extractor to use when building graph data for the counting task, we analyzed the evolution of the metrics during training and validation. We chose the most stable model with the lowest validation loss during training (at epoch 59).

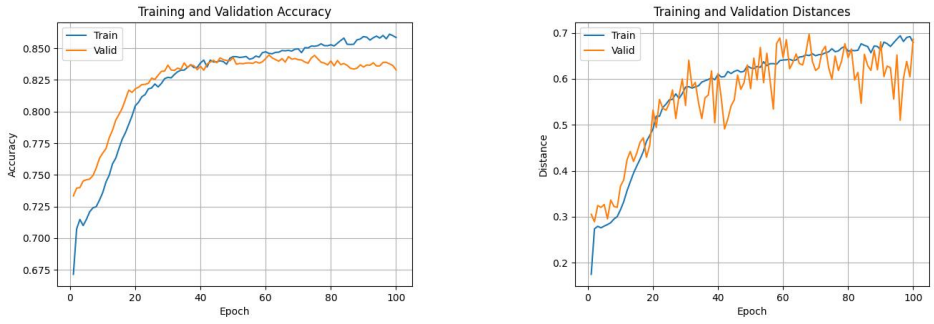

Figure 4.2: Accuracy and distance throughout the training of the contrastive-based feature extractor.

As shown in Figure 4.2, the validation set distances oscillate significantly during training, indicating that the model does not reliably separate positives from negatives. However, even if

oscillating, the distances and accuracy increase, proving that negatives are represented further from the positives, even if this distance is not great. This difficulty could be due to the similarity of anchor and positive samples and the negatives being too different from the two, forming "easy negatives" that the model can not effectively learn from. Solutions like "hard negative" mining and using another loss function that considers more than just one positive and negative sample could be worth exploring to improve the robustness of the feature extractor and, consequently, the results of the graph classification counting model.

4.3.2 Graph Classification Model

Table 4.1 shows the results of the counting graph classification model, considering the different approaches described previously to build the graph image representation. We present the results considering the two feature extraction approaches, handcrafted or contrastive-based, and the two techniques for fragment detection used in the evaluation, connected component analysis (CCA) and the YOLOv9 detections.

Table 4.1: Results for the fragments per set counting task using the graph classification model. (CCA - Connected Component Analysis)

Patches	**Node Features**	**Fragments/Sets**		
		A	**mae**	**f1**
CCA	Handcrafted	0.631	0.816	0.318
	Contrastive	0.444	1.077	0.216
YOLOv9	Handcrafted	**0.709**	0.534	**0.343**
	Contrastive	0.698	**0.492**	0.341

The graph-building approach that showed the best results was the one that used the YOLOv9 fragment detections with the handcrafted features, even though the encoded features with the same fragment detection technique showed similar results. This proves that fragment detection directly influences the performance of the counting method as more robust fragment predictions form more confident node representations that allow for better counting results. The graphs built using YOLOv9 fragments are more compact as CCA overestimates the fragments detected. Since the model was trained using the ground truth spatial annotations of the fragments bounding boxes, it is not robust enough to discern which structures are actually fragments, as this was not the primary task the model was guided to learn. This means that the problem might not be related to the fragment detection method but to what the model learned from the training data, which is partly incompatible with more crude detection approaches.

Another highlight from the results is that the contrastive-based feature extractor underperformed compared to handcrafted features. This goes against our initial intuition that the model would benefit from viewing nodes as embeddings of the spatial and structural relationships between fragments and sets. When analyzing the evolution of the training metrics of the contrastive-based approach, the distance between negative and positive samples increases unstably and within

a small range, which might indicate a lack of robustness in effectively representing opposed fragments. A more robust and expressive feature extractor might capture these relationships better and provide better counting results. Another hypothesis is that the nature of the feature extractor may not be adequate to solve the problem in itself. Considering that handcrafted features provided better counting results, assuming a feature extractor that focuses more on the intrinsic properties of the fragments instead of encoding the relationships between them could yield better performance.

Nevertheless, the handcrafted features performed well for both detection methods, proving that robust extracted features are crucial for the task and can leverage less confident detections. However, using a robust detection method is the determining factor for achieving better performance, as with solid initial fragment detections, even a weaker feature extractor could provide results.

4.3.3 Discussion

As a whole, the graph classification model results were poor when compared to the approaches presented in the previous chapter, being at most comparable with the results obtained using the DINOv2 counting method, which was the worst-performing counting model on the test set, as seen on Table 3.5. In fact, we did not improve the results previously presented for the fragments per set counting task.

Some revisions could be suggested regarding the method proposed beyond those already discussed in the review of the results. Exploring more complex GNN architectures, like GraphSAGE or GIN, or even other aggregation methods, which could be learnable, could make the GNN classification model more invariant to the quality of the fragments detected. As mentioned before, stronger or more adequate feature extractors could provide more expressive node representations from which the model could learn. Another suggestion would be to consider only connections between nodes from the same set instead of building a complete graph or even some simple connections between sets to encode this relationship. This could help the model separate better the fragments by set, improving counting.

The lack of general improvement using this method raises questions about the necessity of employing a graph-based approach altogether, as the complexity of this method might not align with the complexity of the task. Moreover, this method is highly dependent on fragment detection, and considering the results extracted directly from the detection model, YOLOv9, are already good, there is no strong evidence from experimental analysis to suggest it is worth computing the fragments per set count using the graph-based approach.

Notwithstanding, this work was valuable as a deeper analysis of the specific characteristics of fragments and sets and how they relate to each other. This can be seen as an exploratory baseline that can be reworked and improved, or simply used to extract knowledge about the subject to leverage other counting approaches for the fragments per set counting task.

Chapter 5

Conclusions

This work explores the quality control problem of detecting and counting fragments in digital pathology. We analyze how errors with the already proposed model can be mitigated and how pathologists can gain insights into how trustworthy detections are within these systems.

To understand the context of the problem and provide a comprehensive overview of the background knowledge required for its interpretation, we reviewed the existing literature on quality control in digital pathology, detection, and counting methods and their use and application within the digital pathology field. We identified the lack of research in digital pathology for the specific task of detecting and counting fragments as a quality control procedure.

To bridge this gap, we proposed a methodology for improving fragment detection and counting through detection models, deriving counts from them, and developing classification models that single-handedly solve the counting problem. We identified key challenges and variability in tissue fragment shapes and sizes through a comprehensive dataset analysis. We introduced improved detection models, such as YOLOv9, which outperformed YOLOv5 by providing more accurate and confident detections. For counting methodologies, our experiments revealed that deriving counts from detection models generally offered better results than using standalone classification models. Fragment variability posed a substantial challenge, making counting fragments per set more complex than counting sets. YOLOv9 emerged as the best-performing model throughout, cemented by the extensive analysis of its performance across various sample characteristic domains and by the solid results when some domains were withheld from training.

Furthermore, we performed an exploratory analysis on graph-based learning as another approach for counting fragments per set, as handling the problem as a graph classification method could leverage the intrinsic structural and hierarchical properties of fragments and sets. We introduced data samples as graphs of fragment crops connected by edges representing the distances between each fragment. Each node contains features extracted from the cropped fragments by a detection method, such as the bounding box annotations during training and Connected Component Analysis (CCA) or YOLOv9 detections during evaluation. We extracted features by handcrafting representations or extracting feature embeddings through a contrastive-based feature extractor model. The performance of this method was, at best, comparable with the worst counting

approach of the other counting methods. Nonetheless, it serves as a ground study of the structural properties within samples and how they interact, or even as a basis for further improvements of other graph or non-graph-based approaches.

In conclusion, this work enhances quality control systems in digital pathology by improving fragment detection and counting methodologies. Hopefully, the results analyzed in this work can be leveraged to investigate even more accurate and reliable solutions that encourage clinical integration of models in routine digital pathology systems, ultimately supporting better clinical decision-making and expediting patient outcomes.

References

[1] Esther Abels, Liron Pantanowitz, Famke Aeffner, Mark D. Zarella, Jeroen van der Laak, Marilyn M. Bui, Venkata N.P. Vemuri, Anil V. Parwani, Jeff Gibbs, Emmanuel Agosto-Arroyo, Andrew H. Beck, and Cleopatra Kozlowski. Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the digital pathology association. *Journal of Pathology*, 249:286–294, 11 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] Muhammad Joan Ailia, Nishant Thakur, Jamshid Abdul-Ghafar, Chan Kwon Jung, Kwangil Yim, and Yosep Chong. Current trend of artificial intelligence patents in digital pathology: A systematic evaluation of the patent landscape. *Cancers*, 14, 5 2022.

[5] Shazia Akbar and Anne L Martel. Cluster-based learning from weakly labeled bags in digital pathology. *arXiv preprint arXiv:1812.00884*, 2018.

[6] Khaled Al-Thelaya, Marco Agus, Nauman Ullah Gilal, Yin Yang, Giovanni Pintore, Enrico Gobbetti, Corrado Calí, Pierre J. Magistretti, William Mifsud, and Jens Schneider. Inshade: Invariant shape descriptors for visual 2d and 3d cellular and nuclear shape analysis and classification. *Computers and Graphics*, 98:105–125, 2021.

[7] Khaled Al-Thelaya, Nauman Ullah Gilal, Mahmood Alzubaidi, Fahad Majeed, Marco Agus, Jens Schneider, and Mowafa Househ. Applications of discriminative and deep learning feature extraction methods for whole slide image analysis: A survey. *Journal of Pathology Informatics*, 14:100335, 1 2023.

[8] Tomé Albuquerque, Ana Moreira, and Jaime S. Cardoso. Deep ordinal focus assessment for whole slide images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 657–663, October 2021.

[9] Tomé Albuquerque, Ana Moreira, Beatriz Barros, Diana Montezuma Felizardo, Sara Oliveira, Pedro Neto, João Monteiro, Liliana Ribeiro, Sofia Goncalves, Ana Monteiro, Isabel Pinto, and Jaime Cardoso. Quality control in digital pathology: Automatic fragment detection and counting. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, volume 2022, pages 588–593, 07 2022.

[10] Sharib Ali, Nasullah Khalid Alham, Clare Verrill, and Jens Rittscher. Ink removal from histopathology whole slide images by combining classification, detection and image generation models. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 928–932, 2019.

[11] Hani A Alturkistani, Faris M Tashkandi, and Zuhair M Mohammedsaleh. Histological stains: A literature review and case study. *Global Journal of Health Science*, 8:72, 6 2015.

[12] David Ameisen, Christophe Deroulers, Valerie Perrier, Jean-Baptiste Yunès, Fatiha Bouhidel, Maxime Battistella, Luc Legres, Anne Janin, and Philippe Bertheau. Stack or trash? fast quality assessment of virtual slides. *Diagnostic Pathology*, 8:S23, 09 2013.

[13] Ali R. N. Avanaki, Kathryn S. Espig, Albert Xthona, Christian Lanciault, and Tom R. L. Kimpe. Automatic image quality assessment for digital pathology. In Anders Tingberg, Kristina Lång, and Pontus Timberg, editors, *Breast Imaging*, pages 431–438, Cham, 2016. Springer International Publishing.

[14] Wei Ba, Shuhao Wang, Meixia Shang, Ziyan Zhang, Huan Wu, Chunkai Yu, Ranran Xing, Wenjuan Wang, Lang Wang, Cancheng Liu, Huaiyin Shi, and Zhigang Song. Assessment of deep learning assistance for the pathological diagnosis of gastric cancer. *Modern Pathology*, 35:1262–1268, 9 2022.

[15] Morteza Babaie and Hamid R. Tizhoosh. Deep features for tissue-fold detection in histopathology images. In Constantino Carlos Reyes-Aldasoro, Andrew Janowczyk, Mitko Veta, Peter Bankhead, and Korsuk Sirinukunwattana, editors, *Digital Pathology - 15th European Congress, ECDP 2019, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 125–132. Springer Verlag, 2019. Publisher Copyright: © 2019, Springer Nature Switzerland AG.; 15th European Congress on Digital Pathology, ECDP 2019 ; Conference date: 10-04-2019 Through 13-04-2019.

[16] Pinky A. Bautista, Noriaki Hashimoto, and Yukako Yagi. Color standardization in whole slide imaging using a color calibration slide. *Journal of Pathology Informatics*, 5:4, 1 2014.

[17] Pinky A. Bautista and Yukako Yagi. Improving the visualization and detection of tissue folds in whole slide images through color enhancement. *Journal of Pathology Informatics*, 1:25, 1 2010.

[18] Babak Ehteshami Bejnordi, Geert Litjens, Nadya Timofeeva, Irene Otte-Holler, Andre Homeyer, Nico Karssemeijer, and Jeroen AWM van der Laak. Stain specific standardization of whole-slide histopathological images. *IEEE Transactions on Medical Imaging*, 35:404–415, 2 2016.

[19] Andrey V. Belashov, Anna A. Zhikhoreva, Tatiana N. Belyaeva, Anna V. Salova, Elena S. Kornilova, Irina V. Semenova, and Oleg S. Vasyutinskii. Machine learning assisted classification of cell lines and cell states on quantitative phase images. *Cells*, 10(10), 2021.

[20] Aïcha BenTaieb and Ghassan Hamarneh. Predicting cancer with a recurrent visual attention model for histopathology images. In Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 129–137, Cham, 2018. Springer International Publishing.

[21] Francesco Bianconi, Jakob N. Kather, and Constantino Carlos Reyes-Aldasoro. Experimental assessment of color deconvolution and color normalization for automated classification of histology images stained with hematoxylin and eosin. *Cancers*, 12:3337, 11 2020.

[22] Said Boumaraf, Xiabi Liu, Yuchai Wan, Zhongshu Zheng, Chokri Ferkous, Xiaohong Ma, Zhuo Li, and Dalal Bardou. Conventional machine learning versus deep learning for magnification dependent histopathological breast cancer image classification: A comparative study with visual explanation. *Diagnostics*, 11:528, 3 2021.

[23] Romain Brixtel, Sebastien Bougleux, Olivier Lezoray, Yann Caillot, Benoit Lemoine, Mathieu Fontaine, Dalal Nebati, and Arnaud Renouf. Whole slide image quality in digital pathology: Review and perspectives. *IEEE Access*, 10:131005–131035, 2022.

[24] Gabriele Campanella, Vitor Werneck Krauss Silva, and Thomas J. Fuchs. Terabyte-scale deep multiple instance learning for classification and localization in pathology. *CoRR*, abs/1805.06983, 2018.

[25] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

[26] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021.

[27] Thanatip Chankong, Nipon Theera-Umpon, and Sansanee Auephanwiriyakul. Automatic cervical cell segmentation and classification in pap smears. *Computer Methods and Programs in Biomedicine*, 113(2):539–556, 2014.

[28] Hao Chen, Xi Wang, and Pheng Ann Heng. Automated mitosis detection with deep regression networks. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1204–1207, 2016.

[29] Po-Hsuan Cameron Chen, Craig H Mermel, and Yun Liu. Evaluation of artificial intelligence on a reference standard based on subjective interpretation. *The Lancet Digital Health*, 3(11):e693–e695, 2021.

[30] Xihao Chen, Jingya Yu, Shenghua Cheng, Xiebo Geng, Sibo Liu, Wei Han, Junbo Hu, Li Chen, Xiuli Liu, and Shaoqun Zeng. An unsupervised style normalization method for cytopathology images. *Computational and Structural Biotechnology Journal*, 19:3852–3863, 2021.

[31] Wei-Chung Cheng, Firdous Saleheen, and Aldo Badano. Assessing color performance of whole-slide imaging scanners for digital pathology. *Color Research and Application*, 44, 03 2019.

[32] Francesco Ciompi, Oscar Geessink, Babak Ehteshami Bejnordi, Gabriel Silva de Souza, Alexi Baidoshvili, Geert Litjens, Bram van Ginneken, Iris Nagtegaal, and Jeroen van der Laak. The importance of stain normalization in colorectal tissue classification with convolutional networks. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 160–163. IEEE, 4 2017.

[33] Emily Clarke and Darren Treanor. Colour in digital pathology: A review. *Histopathology*, 70, 09 2016.

[34] David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

[35] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, Anant Madabhushi, and Fabio González. High-throughput adaptive sampling for whole-slide histopathology image analysis (hashi) via convolutional neural networks: Application to invasive breast cancer detection. *PLOS ONE*, 13:e0196828, 05 2018.

[36] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[37] Vaishali Durgamahanthi, Ramesh Rangaswami, C Gomathy, and Anita Christaline Jhon Victor. Texture analysis using wavelet-based multiresolution autoregressive model: Application to brain cancer histopathology. *Journal of Medical Imaging and Health Informatics*, 7(6):1188–1195, 2017.

[38] Babak Ehteshami Bejnordi, Maschenka Balkenhol, Geert Litjens, Roland Holland, Peter Bult, Nico Karssemeijer, and Jeroen van der Laak. Automated detection of dcis in whole-slide h& e stained breast histopathology images. *IEEE Transactions on Medical Imaging*, 35, 04 2016.

[39] Babak Ehteshami Bejnordi, Maeve Mullooly, Ruth M Pfeiffer, Shaoqi Fan, Pamela M Vacek, Donald L Weaver, Sally Herschorn, Louise A Brinton, Bram van Ginneken, Nico Karssemeijer, et al. Using deep convolutional neural networks to identify and classify tumor-associated stroma in diagnostic breast biopsies. *Modern Pathology*, 31(10):1502–1512, 2018.

[40] C. Ferri, J. Hernández-Orallo, and R. Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27–38, 2009.

[41] Adrien Foucart, Olivier Debeir, and Christine Decaestecker. Artifact identification in digital pathology from weak and noisy supervision with deep residual networks. In *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, pages 1–6, 2018.

[42] Filippo Fraggetta, Yukako Yagi, Marcial Garcia-Rojo, Andrew J. Evans, J. Mark Tuthill, Alexi Baidoshvili, Douglas J. Hartman, Junya Fukuoka, and Liron Pantanowitz. The importance of eslide macro images for primary diagnosis with whole slide imaging. *Journal of Pathology Informatics*, 9, 1 2018.

[43] Michael Gadermayr, Sean Steven Cooper, Barbara Klinkhammer, Peter Boor, and Dorit Merhof. A quantitative assessment of image normalization for classifying histopathological tissue of the kidney. In Volker Roth and Thomas Vetter, editors, *Pattern Recognition*, pages 3–13, Cham, 2017. Springer International Publishing.

[44] Dashan Gao, Dirk Padfield, Jens Rittscher, and Richard McKay. Automated training data generation for microscopy focus classification. In Tianzi Jiang, Nassir Navab, Josien P. W. Pluim, and Max A. Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, pages 446–453, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[45] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

[46] Anubha Gupta, Rahul Duggal, Shiv Gehlot, Ritu Gupta, Anvit Mangal, Lalit Kumar, Nisarg Thakkar, and Devprakash Satpathy. Gcti-sn: Geometry-inspired chemical and tissue invariant stain normalization of microscopic medical images. *Medical Image Analysis*, 65:101788, 2020.

[47] Maryam Haghighat, Lisa Browning, Korsuk Sirinukunwattana, Stefano Malacrino, Nasullah Khalid Alham, Richard Colling, Ying Cui, Emad Rakha, Freddie C. Hamdy, Clare Verrill, and Jens Rittscher. Automated quality assessment of large digitised histology cohorts by artificial intelligence. *Scientific Reports*, 12:5002, 3 2022.

[48] Maryam Haghighat, Lisa Browning, Korsuk Sirinukunwattana, Stefano Malacrino, Nasullah Khalid Alham, Richard Colling, Ying Cui, Emad Rakha, Freddie C. Hamdy, Clare Verrill, and Jens Rittscher. Automated quality assessment of large digitised histology cohorts by artificial intelligence. *Scientific Reports*, 12:5002, 3 2022.

[49] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[50] Robert Hawkins. Managing the pre- and post-analytical phases of the total testing process. *Annals of Laboratory Medicine*, 32:5–16, 2012.

[51] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[53] Simin He, Jun Ruan, Yi Long, Jianlian Wang, Chenchen Wu, Guanglu Ye, Jingfan Zhou, Junqiu Yue, and Yanggeling Zhang. Combining deep learning with traditional features for classification and segmentation of pathological images of breast cancer. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 01, pages 3–6, 2018.

[54] Md Shakhawat Hossain, Toyama Nakamura, Fumikazu Kimura, Yukako Yagi, and Masahiro Yamaguchi. Practical image quality evaluation for whole slide imaging scanner. In Toyohiko Yatagai, Yoshihisa Aizu, Osamu Matoba, Yasuhiro Awatsuji, and Yuan Luo, editors, *Biomedical Imaging and Sensing Conference*, volume 10711, page 107111S. International Society for Optics and Photonics, SPIE, 2018.

[55] Mahdi S. Hosseini, Jasper A. Z. Brawley-Hayes, Yueyang Zhang, Lyndon Chan, Konstantinos N. Plataniotis, and Savvas Damaskinos. Focus quality assessment of high-throughput whole slide imaging in digital pathology. *IEEE Transactions on Medical Imaging*, 39:62–74, 2018.

[56] Le Hou, Vu Nguyen, Ariel B. Kanevsky, Dimitris Samaras, Tahsin M. Kurc, Tianhao Zhao, Rajarsi R. Gupta, Yi Gao, Wenjin Chen, David Foran, and Joel H. Saltz. Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. *Pattern Recognition*, 86:188–200, 2019.

[57] 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. *CoRR*, abs/1704.04861, 2017.

[58] Bo Hu, Ye Tang, Eric I-Chao Chang, Yubo Fan, Maode Lai, and Yan Xu. Unsupervised learning for cell-level visual representation in histopathology images with generative adversarial networks. *IEEE Journal of Biomedical and Health Informatics*, 23(3):1316–1328, 2019.

[59] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[60] Yongxiang Huang and Albert C. S. Chung. *Evidence Localization for Pathology Images Using Weakly Supervised Learning*, page 613–621. Springer International Publishing, 2019.

[61] Andrew Janowczyk and Anant Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*, 7:29, 1 2016.

[62] Andrew Janowczyk, Ren Zuo, Hannah Gilmore, Michael Feldman, and Anant Madabhushi. Histoqc: An open-source quality control tool for digital pathology slides. *JCO Clinical Cancer Informatics*, pages 1–7, 12 2019.

[63] 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.

[64] Muhammad Nasim Kashif, Shan E. Ahmed Raza, Korsuk Sirinukunwattana, Muhammmad Arif, and Nasir Rajpoot. Handcrafted features with convolutional neural networks for detection of tumor cells in histology images. In *Proceedings - International Symposium on Biomedical Imaging*, volume 2016-June, page 1029 – 1032, 2016. Cited by: 45.

[65] Jakob Nikolas Kather, Cleo-Aron Weis, Alexander Marx, Alexander K. Schuster, Lothar R. Schad, and Frank Gerrit Zöllner. New colors for histology: Optimized bivariate color maps increase perceptual contrast in histological images. *PLOS ONE*, 10:e0145572, 12 2015.

[66] Jason Keighley, Marc de Kamps, Alexander Wright, and Darren Treanor. Digital pathology whole slide image compression with vector quantized variational autoencoders. In John E. Tomaszewski and Aaron D. Ward, editors, *Medical Imaging 2023: Digital and Computational Pathology*, volume 12471, page 124711B. International Society for Optics and Photonics, SPIE, 2023.

[67] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[68] Timo Kohlberger, Yun Liu, Melissa Moran, Po-Hsuan Cameron Chen, Trissia Brown, Jason D. Hipp, Craig H. Mermel, and Martin C. Stumpe. Whole-slide image focus quality: Automatic assessment and impact on ai cancer detection. *Journal of Pathology Informatics*, 10:39, 1 2019.

[69] Bin Kong, Xin Wang, Zhongyu Li, Qi Song, and Shaoting Zhang. Cancer metastasis detection via spatially structured deep network. In Marc Niethammer, Martin Styner, Stephen Aylward, Hongtu Zhu, Ipek Oguz, Pew-Thian Yap, and Dinggang Shen, editors, *Information Processing in Medical Imaging*, pages 236–248, Cham, 2017. Springer International Publishing.

[70] Sonal Kothari, John H. Phan, Richard A. Moffitt, Todd H. Stokes, Shelby E. Hassberger, Qaiser Chaudry, Andrew N. Young, and May D. Wang. Automatic batch-invariant color segmentation of histological cancer images. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 657–660. IEEE, 3 2011.

[71] Sonal Kothari, John H. Phan, and May D. Wang. Eliminating tissue-fold artifacts in histopathological whole-slide images for improved image-based prediction of cancer grade. *Journal of Pathology Informatics*, 4:22, 1 2013.

[72] G.L. Kumar, J.A. Kiernan, and DAKO A/S. *Education Guide - Special Stains and H & E: Pathology*. Dako North America, 2010.

[73] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[74] Yann LeCun, Koray Kavukcuoglu, and Clement Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010.

[75] Byungjae Lee and Kyunghyun Paeng. A robust and effective approach towards accurate metastasis detection and pn-stage classification in breast cancer. In Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 841–850, Cham, 2018. Springer International Publishing.

[76] Chao Li, Xinggang Wang, Wenyu Liu, Longin Jan Latecki, Bo Wang, and Junzhou Huang. Weakly supervised mitosis detection in breast histopathology images using concentric loss. *Medical Image Analysis*, 53:165–178, 2019.

[77] Xingyu Li and Konstantinos N. Plataniotis. A complete color normalization approach to histopathology images using color cues computed from saturation-weighted statistics. *IEEE Transactions on Biomedical Engineering*, 62:1862–1873, 7 2015.

[78] Yi Li and Wei Ping. Cancer metastasis detection with neural conditional random field. *arXiv preprint arXiv:1806.07064*, 2018.

[79] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q Nelson, Greg S Corrado, et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017.

[80] Xavier Moles Lopez, Etienne D'Andrea, Paul Barbot, Anne-Sophie Bridoux, Sandrine Rorive, Isabelle Salmon, Olivier Debeir, and Christine Decaestecker. An automated blur detection method for histological whole slide imaging. *PLoS ONE*, 8:e82710, 12 2013.

[81] Marc Macenko, Marc Niethammer, J. 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, 2009.

[82] Niccolò Marini, Stefano Marchesin, Sebastian Otálora, Marek Wodzinski, Alessandro Caputo, Mart van Rijthoven, Witali Aswolinskiy, John Melle Bokhorst, Damian Podareanu, Edyta Petters, Svetla Boytcheva, Genziana Buttafuoco, Simona Vatrano, Filippo Fraggetta, Jeroen van der Laak, Maristella Agosti, Francesco Ciompi, Gianmaria Silvello, Henning Muller, and Manfredo Atzori. Unleashing the potential of digital pathology data by training computer-aided diagnosis models without human annotations. *npj Digital Medicine*, 5, 12 2022.

[83] Raphaël Marée. The need for careful data collection for pattern recognition in digital pathology. *Journal of Pathology Informatics*, 8:19, 1 2017.

[84] Ezgi Mercan, Selim Aksoy, Linda Shapiro, Donald Weaver, Tad Brunyé, and Joann Elmore. Localization of diagnostically relevant regions of interest in whole slide images: a comparative study. *Journal of Digital Imaging*, 29:496–506, 08 2016.

[85] Diana Montezuma, Sara P. Oliveira, Pedro C. Neto, Domingos Oliveira, Ana Monteiro, Jaime S. Cardoso, and Isabel Macedo-Pinto. Annotating for artificial intelligence applications in digital pathology: A practical guide for pathologists and researchers. *Modern Pathology*, 36:100086, 4 2023.

[86] Chris Murphy. *Histology, Cytology*, pages 991–993. Springer Netherlands, Dordrecht, 2013.

[87] Soojeong Nam, Yosep Chong, Chan Kwon Jung, Tae Yeong Kwak, Ji Youl Lee, Jihwan Park, Mi Jung Rho, and Heounjeong Go. Introduction to digital pathology and computer-aided pathology. *Journal of Pathology and Translational Medicine*, 54:125–134, 2020.

[88] United Nations. The 17 sustainable development goals, 2024.

[89] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[90] Sebastian Otálora, Manfredo Atzori, Vincent Andrearczyk, Amjad Khan, and Henning Müller. Staining invariant features for improving generalization of deep convolutional neural networks in computational pathology. *Frontiers in Bioengineering and Biotechnology*, 7, 8 2019.

[91] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242. IEEE, 7 2020.

[92] Sakari Palokangas, Jyrki Selinummi, and Olli Yli-Harja. Segmentation of folds in tissue section images. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2007:5642–5, 02 2007.

[93] Ankush Patel, Ulysses G.J. 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*, 12:50, 1 2021.

[94] Mohammad Peikari, Mehrdad J. Gangeh, Judit Zubovits, Gina Clarke, and Anne L. Martel. Triaging diagnostically relevant regions from pathology whole slides of breast cancer: A texture based approach. *IEEE Transactions on Medical Imaging*, 35(1):307–315, 2016.

[95] Sathyanarayanan Rajaganesan, Rajiv Kumar, Vidya Rao, Trupti Pai, Neha Mittal, Ayushi Sahay, Santosh Menon, and Sangeeta Desai. Comparative assessment of digital pathology systems for primary diagnosis. *Journal of Pathology Informatics*, 12:25, 1 2021.

[96] Siddhant Rao. Mitos-rcnn: Mitotic figure detection in breast cancer histopathology images using region based convolutional neural networks. *International Journal of Medical and Health Sciences*, 12(10):514 – 520, 2018.

[97] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[98] Zaka Ur Rehman, M. Sultan Zia, Giridhar Reddy Bojja, Muhammad Yaqub, Feng Jinchao, and Kaleem Arshid. Texture based localization of a brain tumor from mr-images by using a machine learning approach. *Medical Hypotheses*, 141:109705, 2020.

[99] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21:34–41, 2001.

[100] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[101] David Romo-Bucheli, Andrew Janowczyk, Hannah Gilmore, Eduardo Romero, and Anant Madabhushi. Automated tubule nuclei quantification and correlation with oncotype dx risk categories in er+ breast cancer whole slide images. *Scientific Reports*, 6, 2016. Cited by: 59; All Open Access, Gold Open Access, Green Open Access.

[102] Santanu Roy, Alok kumar Jain, Shyam Lal, and Jyoti Kini. A study about color normalization methods for histopathology images. *Micron*, 114:42–61, 11 2018.

[103] Caglar Senaras, M. Khalid Khan Niazi, Gerard Lozanski, and Metin N. Gurcan. Deepfocus: Detection of out-of-focus regions in whole slide digital images using deep learning. *PLOS ONE*, 13:e0205387, 10 2018.

[104] Makhmud Shaban, Christoph Baur, Nassir Navab, and Shadi Albarqouni. Staingan: Stain style transfer for digital histological images. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 953–956, 2018.

[105] Hossain Md Shakhawat, Tomoya Nakamura, Fumikazu Kimura, Yukako Yagi, and Masahiro Yamaguchi. [paper] automatic quality evaluation of whole slide images for the practical use of whole slide imaging scanner. *ITE Transactions on Media Technology and Applications*, 8:252–268, 2020.

[106] Harshita Sharma, Norman Zerbe, Sebastian Lohmann, Klaus Kayser, Olaf Hellwich, and Peter Hufnagl. A review of graph-based methods for image analysis in digital histopathology. *Diagnostic pathology*, 1(1), 2015.

[107] Prarthana Shrestha and Bas Hulsken. Color accuracy and reproducibility in whole slide imaging scanners. *Journal of Medical Imaging*, 1:027501, 7 2014.

[108] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[109] Korsuk Sirinukunwattana, Shan E. Ahmed Raza, Yee-Wah Tsang, David R. J. Snead, Ian A. Cree, and Nasir M. Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1196 – 1206, 2016. Cited by: 896; All Open Access, Green Open Access.

[110] Chetan L. Srinidhi, Ozan Ciga, and Anne L. Martel. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67:101813, 1 2021.

[111] Linda Studer, Shushan Toneyan, Inti Zlobec, Heather Dawson, and Andreas Fischer. Graph-based classification of intestinal glands in colorectal cancer tissue images. *Proceedings of MICCAI 2019, 13-17 October 2019, Shenzhen, China*, 2019.

[112] Zaneta Swiderska-Chadaj, Thomas de Bel, Lionel Blanchet, Alexi Baidoshvili, Dirk Vossen, Jeroen van der Laak, and Geert Litjens. Impact of rescanning and normalization on convolutional neural network performance in multi-center, whole-slide classification of prostate cancer. *Scientific Reports*, 10:14398, 9 2020.

[113] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[114] Syed Ahmed Taqi, Syed Abdus Sami, Lateef Begum Sami, and Syed Ahmed Zaki. A review of artifacts in histopathology. *Journal of Oral and Maxillofacial Pathology*, 22:279, 5 2018.

[115] David Tellez, Maschenka Balkenhol, Irene Otte-Höller, Rob van de Loo, Rob Vogels, Peter Bult, Carla Wauters, Willem Vreuls, Suzanne Mol, Nico Karssemeijer, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi. Whole-slide mitosis detection in h&e breast histology using phh3 as a reference to train distilled stain-invariant convolutional networks. *IEEE Transactions on Medical Imaging*, 37(9):2126–2136, 2018.

[116] David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58:101544, 12 2019.

[117] Ultralytics. YOLOv5: A state-of-the-art real-time object detection system. https://docs.ultralytics.com, 2021.

[118] Mira Valkonen, Jorma Isola, Onni Ylinen, Ville Muhonen, Anna Saxlin, Teemu Tolonen, Matti Nykter, and Pekka Ruusuvuori. Cytokeratin-supervised deep learning for automatic recognition of epithelial cells in breast cancers stained for er, pr, and ki-67. *IEEE transactions on medical imaging*, 39(2):534–542, 2019.

[119] Jelica Vasiljević, Friedrich Feuerhake, Cédric Wemmert, and Thomas Lampert. Towards histopathological stain invariance by unsupervised domain augmentation using generative adversarial networks. *Neurocomputing*, 460:277–291, 10 2021.

[120] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[121] Slawomir Walkowski and Janusz Szymas. Quality evaluation of virtual slides using methods based on comparing common image areas. *Diagnostic pathology*, 6 Suppl 1:S14, 03 2011.

[122] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *CoRR*, abs/2402.13616, 2024.

[123] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8052–8072, 2023.

[124] Nicholas Chandler Wang, Jeremy Kaplan, Joonsang Lee, Jeffrey Hodgin, Aaron Udager, and Arvind Rao. Stress testing pathology models with generated artifacts. *Journal of Pathology Informatics*, 12:54, 1 2021.

[125] Shujun Wang, Yaxi Zhu, Lequan Yu, Hao Chen, Huangjing Lin, Xiangbo Wan, Xinjuan Fan, and Pheng-Ann Heng. Rmdl: Recalibrated multi-instance deep learning for whole slide gastric image classification. *Medical image analysis*, 58:101549, 2019.

[126] Zhongling Wang, Mahdi S. Hosseini, Adyn Miles, Konstantinos N. Plataniotis, and Zhou Wang. Focuslitenn: High efficiency focus quality assessment for digital pathology. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020 - 23rd International Conference, Lima, Peru, October 4-8, 2020, Proceedings, Part V*, volume 12265 of *Lecture Notes in Computer Science*, pages 403–413. Springer, 2020.

[127] M. Wransky. *Color Calibration Techniques for True Color Measurement: Computer Interpretation of Color*. Lap Lambert Academic Publishing GmbH KG, 2015.

[128] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Xiaojie Guo. Graph neural networks: foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4840–4841, 2022.

[129] Weidi Xie, Julia Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pages 1–10, 05 2016.

[130] Yuanpu Xie, Fuyong Xing, Xiaoshuang Shi, Xiangfei Kong, Hai Su, and Lin Yang. Efficient and robust cell detection: A structured regression approach. *Medical Image Analysis*, 44:245–254, 2018.

[131] Fuyong Xing, Toby C. Cornish, Tell Bennett, Debashis Ghosh, and Lin Yang. Pixel-to-pixel learning with weak supervision for single-stage nucleus recognition in ki67 images. *IEEE Transactions on Biomedical Engineering*, 66(11):3088–3097, 2019.

[132] Bolei Xu, Jingxin Liu, Xianxu Hou, Bozhi Liu, Jon Garibaldi, Ian O. Ellis, Andy Green, Linlin Shen, and Guoping Qiu. Look, investigate, and classify: A deep hybrid attention method for breast cancer classification. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 914–918, 2019.

[133] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[134] Yukako Yagi. Color standardization and optimization in whole slide imaging. *Diagnostic Pathology*, 6:S15, 12 2011.

[135] Samuel J. Yang, Marc Berndl, D. Michael Ando, Mariya Barch, Arunachalam Narayanaswamy, Eric Christiansen, Stephan Hoyer, Chris Roat, Jane Hung, Curtis T. Rueden, Asim Shankar, Steven Finkbeiner, and Philip Nelson. Assessing microscope image focus quality with deep learning. *BMC Bioinformatics*, 19:77, 12 2018.

[136] Zeiss. A quick guide to cytological staining, 2018.

[137] Teng Zhang, Johanna Carvajal, Daniel F. Smith, Kun Zhao, Arnold Wiliem, Peter Hobson, Anthony Jennings, and Brian C. Lovell. Slidenet: Fast and accurate slide quality assessment based on deep neural networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2314–2319, 2018.

[138] Bingchao Zhao, Chu Han, Xipeng Pan, Jiatai Lin, Zongjian Yi, Changhong Liang, Xin Chen, Bingbing Li, Weihao Qiu, Danyi Li, Li Liang, Ying Wang, and Zaiyi Liu. Restainnet: A self-supervised digital re-stainer for stain normalization. *Computers and Electrical Engineering*, 103:108304, 2022.

[139] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–20, 2022.

[140] Zhi Hua Zhou. *Machine Learning*. Springer Nature, 1 2021.