UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Utilization of Deep Neural Networks in tasks of classification and semantic segmentation of medical images of colon and breast cancer.

End of Degree Project

Bachelor's Degree in Informatics Engineering

AUTHOR: Martínez Martínez, Adrián

Tutor: Gómez Adrian, Jon Ander

ACADEMIC YEAR: 2023/2024

Resum

El TFG que es proposa s'enfoca en l'exploració i aplicació de diverses tècniques de classificació i segmentació d'imatges dins l'àmbit de la imatge mèdica. El principal objectiu d'aquest TFG és millorar la precisió i eficiència en el diagnòstic de diferents tipus de càncer mitjançant l'ús de models basats en tècniques d'intel·ligència artificial, en particular, Deep Learning (Xarxes Neuronals profundes). Les tècniques de classificació i segmentació d'imatges que emprarem tenen com a propòsit contribuir a la identificació i diferenciació precisa de les característiques patològiques en imatges mèdiques, en particular, de càncer de còlon i de mama. En el cas del càncer de còlon, per a detectar pòlips sobre la base de la morfologia cel·lular observable en imatges de microscopi. I en el segon cas, el del càncer de mama, per a detectar diferents estadis de la malaltia. En l'àmbit global, aquest enfocament busca obtindre diagnòstics més precisos i de la manera més primerenca possible, per a millorar tant la taxa de supervivència com la qualitat de vida dels pacients. El treball a desenvolupar en aquest TFG es divideix en dues línies de treball similars. La primera línia treballarà amb imatges mèdiques de càncer de còlon obtingudes amb microscopi, corresponents a un dataset disponible com a públic en internet, i consistirà en la classificació de les imatges en un total de 6 classes objectiu possibles, una corresponent a teixit sa, una segona corresponent a hiperplàsia benigna, i quatre corresponents a dos tipus de càncer en dos estadis (lleu i greu) de la malaltia. La segona línia se centrarà en imatges obtingudes mitjançant microscopi de biòpsies de mama corresponents a dos datasets diferents, un d'ells on s'han identificat un total de set classes objectiu, i un altre, que implica una tasca més ambiciosa i complexa, on s'han identificat un total de 21 classes objectiu.

Paraules clau: Aprenentatge Profund; Xarxes Neuronals; Imatge Mèdica; Segmentació Semàntica; Diagnòstic de Càncer

Resumen

El TFG que se propone se enfoca en la exploración y aplicación de diversas técnicas de clasificación y segmentación de imágenes dentro del ámbito de la imagen médica. El principal objetivo de este TFG es mejorar la precisión y eficiencia en el diagnóstico de diferentes tipos de cáncer mediante el uso de modelos basados en técnicas de Inteligencia Artificial, en particular, Deep Learning (Redes Neuronales profundas). Las técnicas de clasificación y segmentación de imágenes que emplearemos tienen como propósito contribuir a la identificación y diferenciación precisa de las características patológicas en imágenes médicas, en particular, de cáncer de colon y de mama. En el caso del cáncer de colon, para detectar pólipos en base a la morfología celular observable en imágenes de microscopio. Y en el segundo caso, el del cáncer de mama, para detectar diferentes estadíos de la enfermedad. A nivel global, este enfoque busca obtener diagnósticos más precisos y de la manera más temprana posible, para mejorar tanto la tasa de supervivencia como la calidad de vida de los pacientes. El trabajo a desarrollar este TFG se divide en dos líneas de trabajo similares. La primera linea trabajará con imágenes médicas de cáncer de colon obtenidas con microscopio, correspondientes a un dataset disponible como público en internet, y consistirá en la clasificación de las imágenes en un total de 6 clases objetivo posibles, una correspondiente a tejido sano, una segunda correspondiente a hiperplasia benigna, y cuatro correspondientes a dos tipos de cáncer en dos estadíos (leve y grave) de la enfermedad. La segunda línea se centrará en imágenes obtenidas mediante microscopio de biopsias de mama correspondientes a dos datasets distintos, uno de ellos donde se han identificado un total de siete clases objetivo, y otro, que implica una tarea más ambiciosa y compleja, donde se han identificado un total de 21 clases objetivo.

Palabras clave: Aprendizaje Profundo; Redes Neuronales; Imagen Médica; Segmentación Semántica; Diagnóstico de Cáncer

Abstract

The proposed undergraduate thesis focuses on the exploration and application of various techniques for image classification and segmentation within the field of medical imaging. The main objective of this thesis is to improve the accuracy and efficiency of diagnosing different types of cancer using models of Artificial Intelligence techniques, particularly Deep Learning (Neural Networks). The image classification and segmentation techniques that we will employ aim to contribute to the precise identification and differentiation of pathological features in medical images, specifically in colon and breast cancer. In the case of colon cancer, to detect polyps based on observable cellular morphology in microscope images. And in the second case, breast cancer, to detect different stages of the disease. Globally, this approach aims to achieve more accurate and earlier diagnoses, to improve both the survival rate and the quality of life of patients. The work proposed in this thesis is divided into two similar lines of work. The first line will work with medical images of colon cancer obtained with a microscope, corresponding to a dataset publicly available on the internet, and will consist of classifying the images into a total of 6 possible target classes: one corresponding to healthy tissue, a second corresponding to benign hyperplasia, and four corresponding to two types of cancer in two stages (mild and severe) of the disease. The second line will focus on classifying and segmenting histopathological breast images corresponding to two different datasets: one with a total of seven target classes, and another, for a more ambitious and complex task, with a total of 21 target classes.

Key words: Deep Learning; Neural Networks; Medical Imaging; Semantic Segmentation; Cancer Diagnosis

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

1.1 Motivation

According to the report *"Las cifras del cáncer en España 2022"* [1], the *Agency for Research on Cancer* estimated that in 2020, 18.1 million new cases of cancer were diagnosed worldwide. Furthermore, the agency projected a significant increase in these numbers, predicting up to 27.0 million new cases by the year 2024. This alarming rise underscores the critical need for enhanced cancer research and improved diagnostic methods.

Among the cases diagnosed in 2020, the most prevalent pathologies were breast cancer, accounting for 12.5% of the cases, lung cancer at 12.2%, and colorectal cancer at 10.7%. These three types alone represented a substantial portion of the global cancer burden, highlighting the importance of targeted research and intervention strategies for these specific cancers.

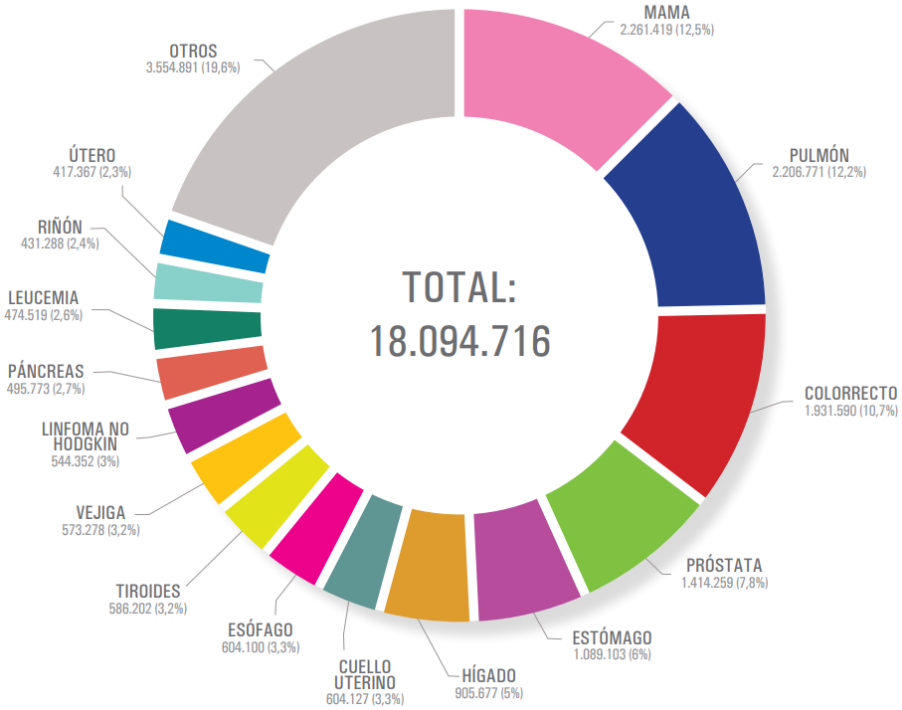

Figure 1.1: Distribution of Cancer Cases Diagnosed in 2020. Extracted from *"Las cifras del cáncer en España 2022"*.

Early detection of polyps can significantly improve the survival probabilities of affected patients, as they can sometimes be removed before they have even turned into tumours. Methods based on *Machine Learning* can aid in this task, assisting doctors and pathologists in making diagnoses. As *Korbar et al.* [2] noted, differentiating between sessile serrated polyps and innocuous hyperplastic polyps, within the realm of colorectal polyps, is a challenging task for pathologists because the diagnosis of these types of polyps is entirely based on morphological characteristics, characteristics that could, in principle, be learned by computer image classification models.

Furthermore, as it will be discussed in the state-of-the-art section, interesting studies are being conducted regarding computer vision methods that could enhance the detection of these pathologies.

Despite the promising advancements in these new models and methodologies, a significant challenge remains: the scarcity of publicly available datasets with labelled cancer images. As of March 2024, a search on *Papers with code* [1] within the *Datasets* section yields only 25 results for the *Images* modality. This limited availability is further complicated by the disparity in labelling practices—both in terms of type, format, and method—and the variety of image types. Consequently and despite these challenges, ongoing research must focus on developing methods and strategies to extract maximum information and utility from the existing data.

Given all these considerations, the primary motivation for this work is to explore and understand the leading approaches for addressing cancer classification and segmentation problems in medical images. Additionally, this work aims to delve deeply into the field of computer vision to comprehend the current research directions and the most recent advancements in methodologies. Beyond the technical exploration, a significant driving force behind this work is the aspiration to apply the knowledge and skills acquired during the degree program to a task that has the potential to positively impact the lives of individuals affected by these kinds of pathologies. Achieving successful outcomes in this area can lead to substantial improvements in the quality of life and prognosis for cancer patients.

1.2 Objectives

The objectives of this work are to understand the main techniques in computer vision and image processing as applied to computational pathology, specifically within the realm of detecting various types of cancer.

Another crucial objective is to identify and comprehend the challenges and difficulties that may arise when dealing with different datasets. This involves understanding how to effectively utilize the available information to address and solve the problem at hand.

Lastly, one of the significant issues encountered in these types of image datasets, particularly those stained with Haematoxylin and Eosin, is the lack of standardization in staining. This variability can affect the consistency and reliability of image analysis. To address this, the present work proposes a discretization approach for the images, whereby each pixel can only assume one of K possible values. These values will be assigned to each pixel of each image through a clustering process. The goal of this approach is to reduce the range of possible values a single pixel can take to a reduced set versus the 2^24 possible colour values, thereby minimizing variations. This reduction in variability aims to help the model perform better in tasks such as classification or segmentation by simplifying the input data and making it more uniform.

[1] https://paperswithcode.com/

This discretization process is intended to standardize the input images, thereby potentially improving the performance of computational models in distinguishing between different types of cancerous tissues. By systematically reducing the colour variations within the images, the model can focus on more consistent features, which may enhance its ability to correctly classify or segment the images.

1.3 Structure of the dissertation

The structure of this dissertation will begin with a comprehensive review of the state of the art in computer image processing for classification and segmentation. This section will delve into the latest advancements, methodologies, and technologies currently being utilized in this field, providing a solid foundation for understanding the context and significance of the subsequent research.

Following this review, a detailed explanation of the datasets used in the various experiments conducted will be presented. This section will describe the characteristics, sources, and preparation of these datasets, as well as any preprocessing steps undertaken to ensure their suitability for the experiments.

Once the datasets have been widely explained, the dissertation will proceed to present the results obtained from the experimentation with these datasets. This section will include an analysis of the performance of different methods and models, discussing their efficacy and any observed trends or patterns.

Finally, the dissertation will conclude with a discussion of the conclusions drawn from this experimentation. This section will synthesize the findings, highlighting the key insights and implications of the research. It will also consider the broader impact of the results on the field of computational pathology and suggest potential directions for future research. Through this structured approach, the dissertation aims to provide a clear and comprehensive understanding of the research conducted.

CHAPTER 2
State of the art

In this section, we will delve into the foundation methodologies for image classification and segmentation within the realm of computer vision. Despite the rapid evolution of deep learning techniques, these conventional approaches persist as fundamental pillars in the field, offering valuable insights and benchmarks for evaluating the efficacy of contemporary methods.

Furthermore, we will explore recent advancements and state-of-the-art architectures that have emerged from cutting-edge research endeavours. This comprehensive overview aims to provide a nuanced understanding of the evolutionary trajectory of image analysis techniques, underscoring the ongoing quest for enhanced accuracy, efficiency, and interpretability in visual recognition systems.

2.1 Convolutional Neural Networks

2.1.1. Clasification

Studies such as *Barbano et al.* [3] advocate for the use of *ResNet* architectures in the classification of various types of colorectal polyps directly from Whole-Slide Images (WSIs). They argue that empirical evidence from works like *Korbar et al.* [2] supports the suitability of residual architectures for this type of task.

To elaborate, the study by *Korbar et al.* involved a comprehensive comparison of *ResNets* of different depths against several well-established architectures, including AlexNet [4], VGG [5], and GoogleNet [6]. The findings of this study highlighted the superior performance of *ResNets*, which the authors attributed to the architecture's ability to address the *Vanishing Gradient Problem*. This problem often hampers the training of deep neural networks by causing gradients to diminish as they propagate through the network layers, thereby impairing learning. *ResNets* mitigate this issue, enabling the development of deeper and more precise models for the characterization of histological images, which was not as feasible with previous architectural approaches.

The *ResNet* architecture, originally introduced by *Kaiming et al.* [7], is distinguished by its use of *residual connections* or *skip connections*. These connections perform identity mapping, allowing the input of one layer to bypass intermediate layers and be directly passed to a subsequent layer, typically the next non-linear layer. This mechanism is crucial because it facilitates the direct propagation of gradients through the network during training. By bypassing one or more layers, the gradient can continue to flow unimpeded, effectively mitigating the vanishing gradient problem that often plagues deep networks.

2.1.2. Segmentation

In the domain of segmentation tasks, which involve assigning a class label to each pixel in an image, one of the most widely adopted architectures is the **U-Net** [8]. This architecture is a specialized type of Convolutional Neural Network (*CNN*) designed to effectively handle the complexities of biomedical image segmentation tasks.

The **U-Net** architecture is characterized by two primary components: the **contracting path** and the **expanding path**. The contracting path, also referred to as the *encoder*, is responsible for capturing the global context of the image. It achieves this through a series of convolutional and pooling layers, which progressively reduce the spatial dimensions of the input while simultaneously increasing the depth, thereby enabling the network to capture and understand features of the image.

Conversely, the expanding path, often called the *decoder*, mirrors the structure of the contracting path but in a reversed manner. Its purpose is to achieve precise localization by reconstructing the segmented image from the features extracted by the encoder. This reconstruction process involves a series of upsampling and convolutional layers, which increase the spatial dimensions of the feature maps while decreasing their depth.

The interplay between these two paths is what gives the **U-Net** its distinctive U-shaped architecture, from which it derives its name.

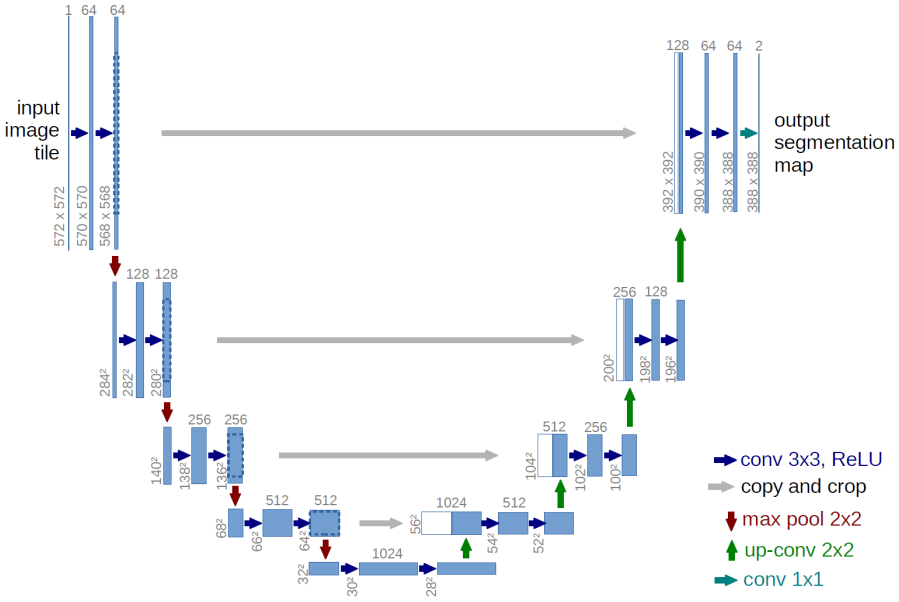

Figure 2.1: U-Net Architecture. Image extracted from *Ronnenberger et al.*

Furthermore, this architecture utilizes *Skip Connections* that concatenate the feature maps from the Encoder with those from the Decoder. These connections serve the following purposes:

- **Assisting in recovering fine details in the prediction:** This is achieved by combining the high-resolution features from the encoder path with the upsampled features in the decoder path.
- **Preventing substantial information loss from layer to layer without these connections:** During the process of encoding in a U-Net, important spatial information can be lost due to downsampling. Skip connections mitigate this issue by directly transferring feature maps from the encoder to the decoder, preserving detailed information.

- **Acting as a strategy against the vanishing gradient problem:** In deep neural networks, gradients can vanish during backpropagation, making the network hard to train. Skip connections allow gradients to flow directly through the network, alleviating the vanishing gradient problem.

- **Allowing the reuse of features from earlier layers with the same dimensionality:** Skip connections enable the network to reuse features from earlier layers that have the same dimensionality. This is beneficial as these features contain rich, high-resolution details that can improve the quality of the output prediction.

- **Facilitating the recovery of spatial information lost during subsampling:** Subsampling operations (like pooling) in the encoding path of a U-Net reduce the spatial resolution of feature maps, leading to loss of spatial information. Skip connections help recover this lost spatial information by providing a shortcut for this information to reach the decoder path. This results in output predictions that better preserve the spatial structures of the input.

2.2 Vision Transformers (ViT)

The **Transformer architecture**, introduced by *Vaswani et al.* [9], was originally designed for natural language processing (NLP) tasks. It revolutionized NLP by leveraging self-attention mechanisms, allowing the model to learn the relevance and context of all words in a sentence. The architecture is based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. This makes the architecture highly parallelizable and efficient to train.

The **Visions Transformer (ViT)**, as proposed by Dosovitskiy *et al.* [10], applies the Transformer architecture to the field of computer vision. The ViT model treats an image as a sequence of patches, similar to how a sentence is treated as a sequence of words in NLP. Each patch from an image is linearly projected, and position and classification embeddings are added. This sequence of image patches is then fed to a Transformer encoder.

Recent studies, such as that of *R. J. Chen, C. Chen, Y. Li et al.* [11], have applied the ViT model to biomedical image classification problems. They introduced a new ViT architecture called Hierarchical Imaged Pyramid Transformer (HIPT), which leverages the natural hierarchical structure inherent in whole-slide images using two levels of self-supervised learning to learn high-resolution image representations.

While it is true that the work starts from the WSI (Whole Slide Image), this method divides the image into cuts at different levels to apply the attention mechanisms, as applying these mechanisms pixel by pixel (*naive self-attention*) would be computationally unfeasible. The work of the article can be seen as a formulation of *Multiple Instance Learning (MIL)* which pre-trains not only the step of extracting fine-grained features from the 16×16 image, but also the subsequent aggregation layers that extract coarse-grained morphological features.

Something to keep in mind about ViTs is that they perform better when pre-trained on large datasets, such as ImageNet21k [12], and then fine-tuned on specific tasks. This is because Transformers do not have strong inductive biases towards certain features of images like CNNs do, and therefore, they need more data to learn useful features.

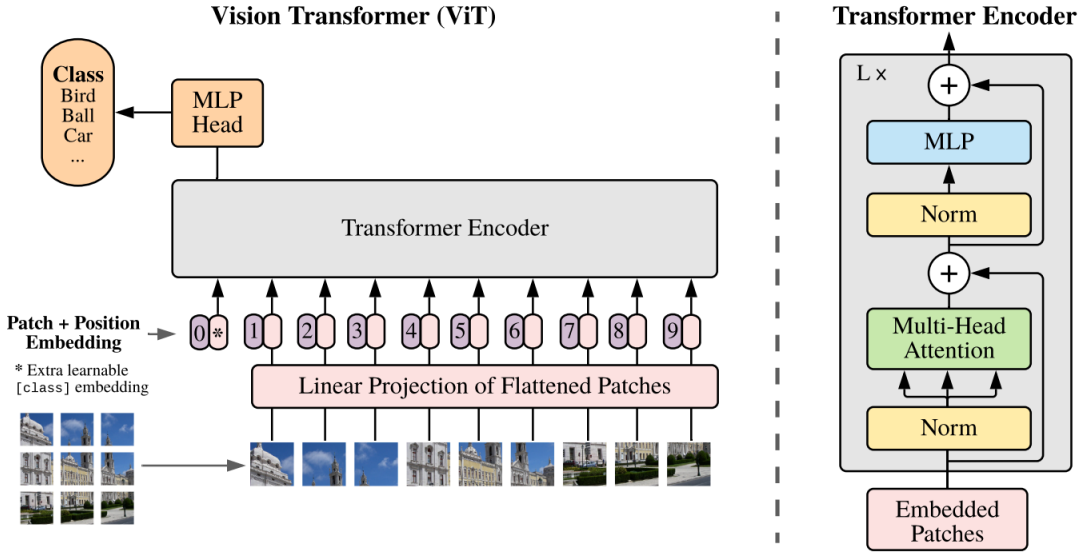

Figure 2.2: Summary of the Vision Transformers (ViT) Model. Image extracted from *A. Dosovitskiy, L. Beyer, A. Kolesnikov et al.* (2021).

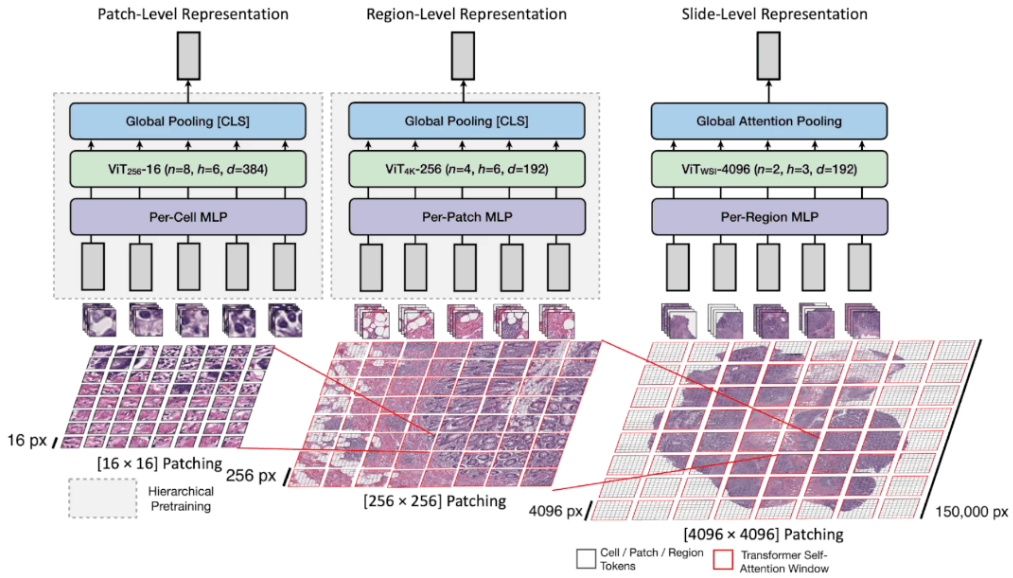

Figure 2.3: HIPT Architecture. Image extracted from https://github.com/mahmoodlab/hipt .

2.3 Hybrid Methods: CNN and Transformers

2.3.1. Clasification

Despite the widespread acclaim achieved by Vision Transformers (ViTs), which swiftly outpaced Convolutional Neural Networks (ConvNets) as the leading models for image classification in research studies, their application encounters notable challenges when extended to broader computer vision tasks such as object detection and semantic segmentation. While ViTs have showcased remarkable performance in image classification benchmarks, their efficacy diminishes when confronted with tasks requiring nuanced understanding of spatial relationships and contextual information within images. This limitation highlights the difference nature of computer vision challenges, where differ-

ent tasks demand tailored architectural considerations and algorithmic approaches for optimal performance.

In works such as that of *Liu et al. (2021)* [13] Swin Transformers are introduced, which integrate various previous key principles of Convolutional Neural Networks (CNNs), thereby rendering the Transformer architecture viable as a foundational element in hybrid methodologies that amalgamate these two paradigms. However, the effectiveness of these models is widely attributed to the intrinsic superiority of the Transformer architecture, rather than to the inherent inductive biases of convolutions.

This has led to the revisiting of the aforementioned Convolutional Neural Network-based classification models, resulting in the ConvNeXt architecture [14] (*Convolutional Neural Networks Extensible*), where a "modernization of pure CNN-based methods is explored, competing with Transformers in terms of accuracy and scalability while maintaining the simplicity and efficiency of standard CNNs.

ConvNeXts have emerged as a significant evolution in the field of convolutional neural network architectures, drawing inspiration from Vision Transformers (ViTs) and demonstrating considerable accuracy across a variety of vision tasks. This success has been evidenced even in comparison with the aforementioned Swin Transformers, which until recently were considered the state-of-the-art in many computer vision applications.

A key feature that distinguishes ConvNeXts is their focus on depth-wise separable convolution. While traditional convolutional networks apply multiple filters to each input channel, ConvNeXts choose to apply a single filter per channel. This approach not only simplifies the network architecture, but also significantly reduces computational complexity. As a result, ConvNeXts-based models tend to be more resource-efficient and more scalable compared to their conventional counterparts.

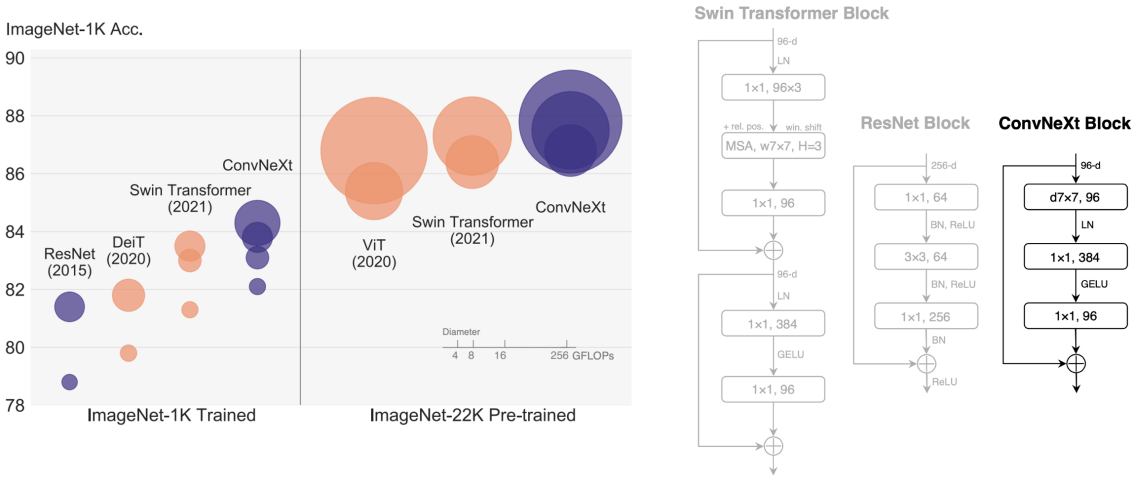

Figure 2.4: Precision of CNN, ViT and ConvNeXt over ImageNet.
Image extracted from *Liu et al.* (2022).

In terms of performance, this category of networks has consistently showcased outstanding accuracy, particularly concerning assessments conducted on the benchmark dataset, *ImageNet*.

2.3.2. Segmentation

Within the field of general-purpose image segmentation, the SAM (Segment Anything) model [15], developed by Meta AI Research and the FAIR (Fundamental AI Research) laboratory in the year 2023, stands out. One of the most prominent features of this model

lies in its ability to segment images with notable precision without requiring tailored, specific pre-training protocols. This capability has sparked significant interest in the research community and positioned SAM as a reference in the field of image segmentation.

It is noteworthy to emphasize that SAM lays the groundwork for the development of more specialized, sophisticated and advanced models within the field. An exemplary instance is MedSAM (Segment Anything in Medical Images) [16], a derivative of SAM specifically designed for the segmentation of medical images. Notably, this model has undergone training utilizing a substantial dataset comprising in excess of 1.5 million image-mask pairs.

In particular, MedSAM has demonstrated remarkable ability to identify and segment over 30 different types of cancer in medical images. This achievement represents a significant advancement in disease diagnosis and treatment, as it provides medical professionals with a tool to consider for diagnostic assistance.

The article authored by Lee et al. (2024) [17] delves into an extensive exploration of SAM's adaptability and extensions across diverse medical imaging domains. It examines how SAM's foundational principles can be adeptly leveraged to address the diverse challenges encountered in various biomedical imaging scenarios, thereby amplifying the efficacy of image segmentation methodologies. The study explains the versatility of SAM, showcasing its capacity to be adapted to the complexity of different medical imaging modalities and pathology types.

Moreover, the integration of SAM with open source annotation tools, such as Label Studio [1], emerges as an important facilitator in the annotation process for segmentation tasks. This synergy not only expedites the labelling process but also ensures the generation of high-quality annotations, thereby enhancing the robustness and reliability of the segmentation models trained on annotated datasets.

[1] https://labelstud.io/blog/exploring-the-powerful-segment-anything-model-integration/

CHAPTER 3
Datasets Definition

Below a detailed exposition and explanation of the three datasets used:

- **Unitopatho**: a dataset consisting of colorectal cancer images with 6 different classes.
- **BCSS**: a dataset comprising breast cancer images with 22 different classes.
- **BRACS**: a dataset of large-scale breast cancer images with 8 different classes (7 tumor types and background).

3.1 Colorectal Cancer

3.1.1. Unitopatho

The Unitopatho dataset, as detailed in the study by Barbano et al. [3], consist of 292 labelled images portraying stained Hematoxylin and Eosin (H&E) patches. It is intended for training deep neural networks for the classification of colorectal polyps and adenomas.

The images were captured using a Hamamatsu Nanozoomer S210 scanner operating at a 20x magnification, corresponding to a spatial resolution of 0.4415 $\mu m/px$. Notably, each image in the dataset has been labelled by expert pathologists and belongs to a different patient and one of six different classes:

Table 3.1: Unitopatho dataset labels

Label	Description
NORM	Normal tissue
HP	Hyperplastic Polyp
TA.HG	Tubular Adenoma, High-Grade dysplasia
TA.LG	Tubular Adenoma, Low-Grade dysplasia
TVA.HG	Tubulo-Villous Adenoma, High-Grade dysplasia
TVA.LG	Tubulo-Villous Adenoma, Low-Grade dysplasia

While the article discusses conducting tests with various scales, ranging from patch scales (σ) in the range [100, 8000], it is important to highlight that the publicly available dataset only provides images in two sizes: $800\mu m$ and $7000\mu m$. Particularly, the former exhibits a smaller area of the original image compared to the latter.

The experimentation data presented in this article indicates that having a single classifier that discriminates between different tumour classes and their variations with accept-

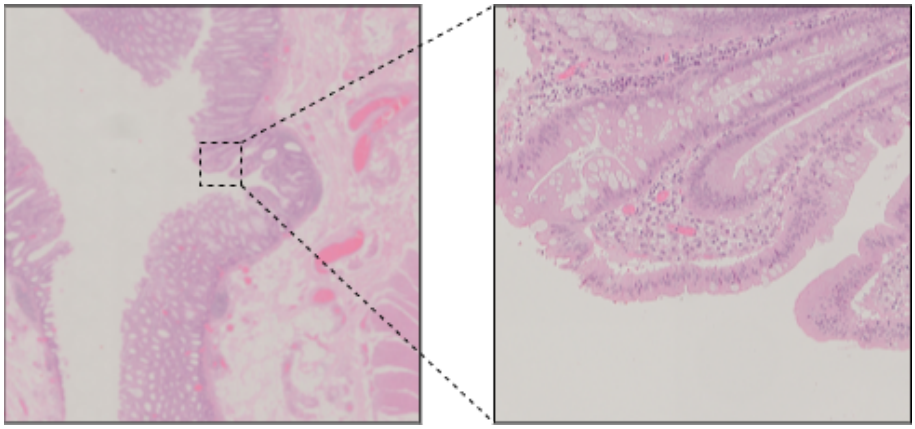

Figure 3.1: Resolution relationship in Unitopatho. On the left, an image of 7000μm, and on the right, a region of the same image at 800μm.

able confidence does not yield satisfactory results. The proposed method is an *ensemble* of cascade classifiers operating at different resolutions.

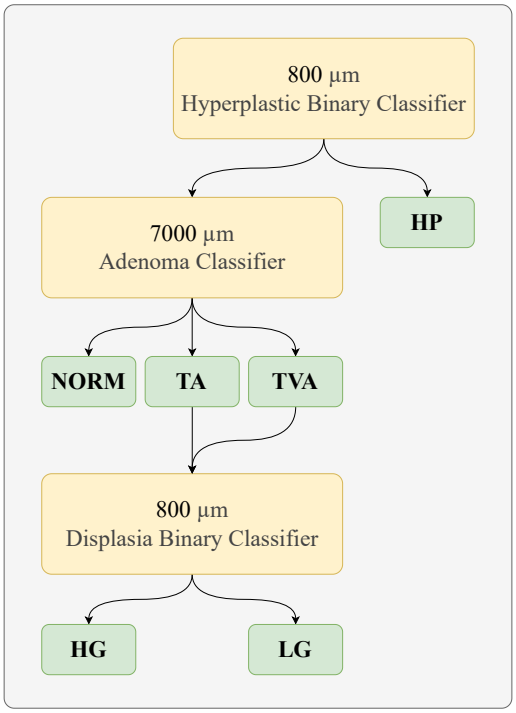

Figure 3.2: Original classification model diagram for the Unitopatho dataset.

The original study demonstrates that Hyperplastic Polyps (HP) and the grades of dysplasia are better classified by images at 800 μm, indicating that the patterns for classifying these types are clearer at the microstructure level, whereas the Tubular Adenoma and Tubulovillous Adenoma groups appear to be better classified at the tissue macrostructure level.

A limitation of this dataset is that only these extracted images from WSIs are available, without direct access to the original WSIs and their annotations. In fact, from these extractions, only the label at a general level is available, and in the CSV files containing the annotations, there are coordinates as well as the width and height of the image. However, regarding the original WSI, the exact tumour coordinates within these crops are not available. This makes it impossible to segment these images. Instead of resiz-

ing the images to 224×224 to fit the input of architectures such as ResNet18, ResNet50, or DenseNet121, it would be preferable to extract images of those sizes to avoid losing potentially useful information in this resizing process.

Additionally, due to only having the label at a general level, the use of the dataset for other tasks, such as semantic segmentation of different types of polyps in the images, is not feasible.

Furthermore, the number of images is not balanced across the different classes. For this reason, Barbano et al. [3] utilize a metric called Balanced Accuracy (BA), which refers to balanced precision. This BA is the average of true positives (sensitivity) and true negative rate (specificity) for each class.

Table 3.2: Class distribution in the Unitopatho dataset images

	HP	**NORM**	**TA**		**TVA**		**Total**
			HG	LG	HG	LG	
Slides	41	21	26	146	20	38	**292**
$\sigma = 7000$	59	74	98	411	93	132	**867**
$\sigma = 800$	545	950	454	3618	916	2186	**8699**
Total	604	1024	552	4029	1009	2318	**9536**

3.2 Breast Cancer

3.2.1. BCSS

The BCSS dataset (Breast Cancer Semantic Segmentation) comprises over 20,000 annotations for image segmentation of breast cancer tissue regions obtained from TCGA (The Cancer Genome Atlas). These annotations were conducted by 25 pathologists, pathology residents, and medical students using the Digital Slide Archive [1].

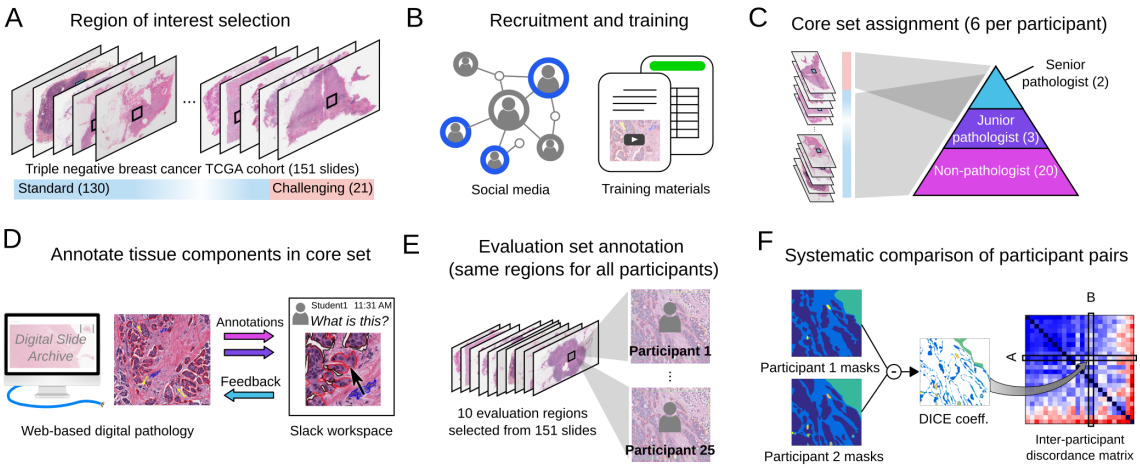

Figure 3.3: BCSS dataset generation process. Image extracted from *Amgad et al. (2019)*.

This dataset consists of images from 151 confirmed cases of triple-negative breast cancer and includes annotations for 22 classes, the codes of which are specified in Table 3.3. However, information regarding the proportion of these labels in the dataset was not found. Nonetheless, there is an online image viewer where the corresponding labels of each WSI image can be viewed, as well as the ability to zoom in, download the

[1] https://digitalslidearchive.github.io/

viewed area, activate or deactivate labels, etc. This viewer is available on the website https://demo.kitware.com/histomicstk/histomicstk. The original WSIs can also be downloaded from this website.

Additionally, there is a public GitHub repository [2] for this dataset, from which the necessary tools for downloading the ROIs with their respective segmentation masks are provided.

Table 3.3: *Ground Truth Codes* for the 22 classes of the BCSS dataset

Label	Code
Outside Roi	0
Tumor	1
Stroma	2
Lymphocytic Infiltrate	3
Necrosis Or Debris	4
Glandular Secretions	5
Blood	6
Exclude	7
Metaplasia Nos	8
Fat	9
Plasma Cells	10
Other Immune Infiltrate	11
Mucoid Material	12
Normal Acinus Or Duct	13
Lymphatics	14
Undetermined	15
Nerve	16
Skin Adnexa	17
Blood Vessel	18
Angioinvasion	19
Dcis	20
Other	21

For more detailed information on the dataset extraction process, refer to *Amgad et al. (2019)* [18].

Regarding the labeling of this dataset, as depicted in Figure 3.4, only a small area of the WSI is being labeled. However, as shown in Figure 3.5, once the area is delineated, the labeling is done exhaustively.

This labelling approach theoretically produces high-quality labels for segmentation tasks. However, it excludes a lot of information from the rest of the image. While the most important classes, those related to tumours, may be present in these labelled areas, models should have enough information to learn where the tumour is not and also recognize areas where nothing should be labelled.

3.2.2. BRACS

The BReAst Carcinoma Subtyping (BRACS) dataset, presented in [19], comprises Hematoxylin and Eosin (HE) stained images, divided into 547 Whole-Slide Images (WSIs) and 4539 Regions of Interest (ROIs) extracted from the WSIs. Each WSI, along with its corre-

[2] https://github.com/PathologyDataScience/BCSS

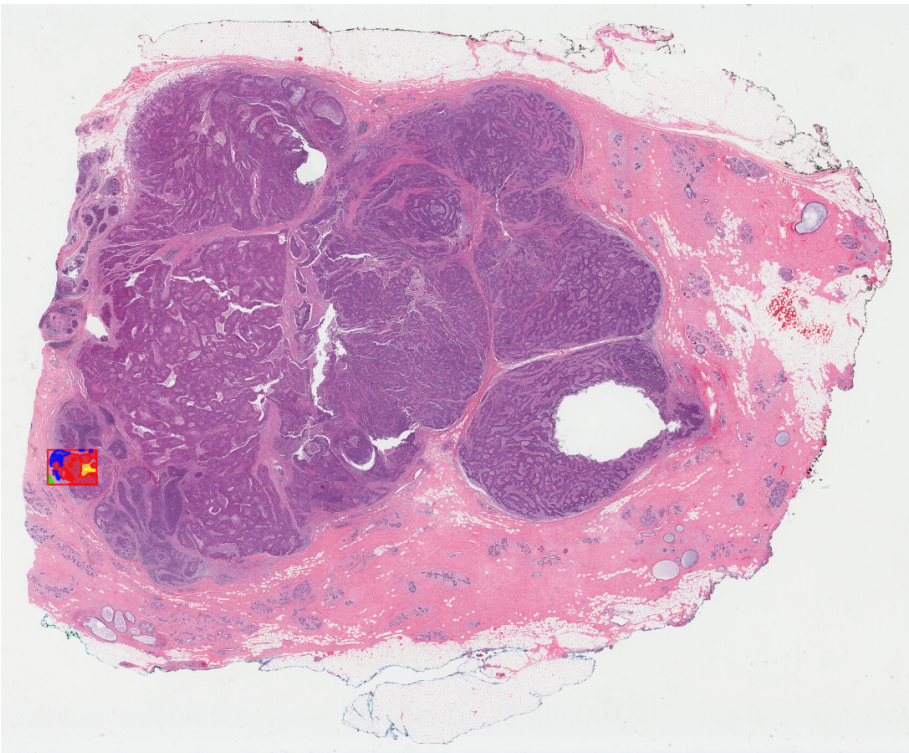

Figure 3.4: BCSS WSI labelling example.

sponding ROIs, is annotated by the consensus of three certified pathologists in different lesion categories. Specifically, BRACS includes three types of lesions: benign, malignant, and atypical, further subdivided into a total of seven categories.

The WSIs are stored in `.svs` format as a multi-resolution pyramid, where the maximum resolution can easily exceed 100,000×100,000 pixels. On the other hand, the ROIs correspond to a region with a 40× magnification, and their resolution can exceed 4,000×4,000 pixels.

Both groups of images have the same type of grouping, where the images are separated into three main groups: BT - benign tumours, AT - atypical tumours, and MT - malignant tumours.

The BT group contains samples of the N type for normal samples, PB for pathologically benign samples, and UDH for samples with Usual Ductal Hyperplasia.

The AT group is further subdivided into two groups. On one hand, the FEA group for Flat Epithelial Atypia and ADH for Atypical Ductal Hyperplasia.

Finally, the MT group is divided into two subsets, including samples annotated as Ductal Carcinoma in Situ (DCIS) and Invasive Carcinoma (IC).

In Table 3.4 the number of WSI images is specified by typology and dataset split (training, validation, and test) for this dataset.

On the other hand, in Table 3.5 the same description can be seen regarding the Regions of Interest.

Of these two subsets of images, only the annotations for the WSIs are available for download. These annotations can be used in segmentation tasks, as they label different classes within the same image and outline the region of interest.

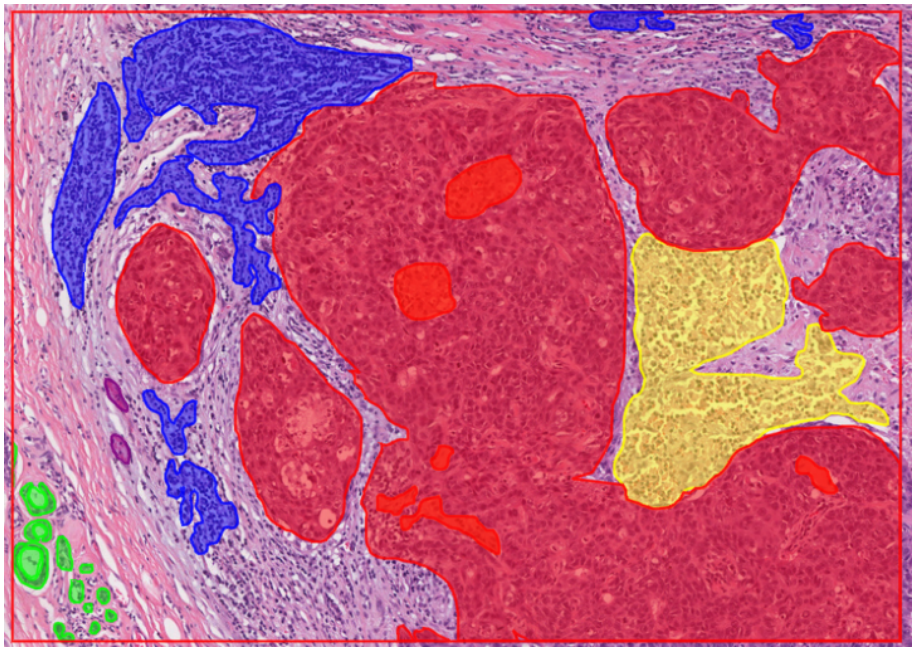

Figure 3.5: BCSS WSI labelling example. Expansion of the image from Figure 3.4.

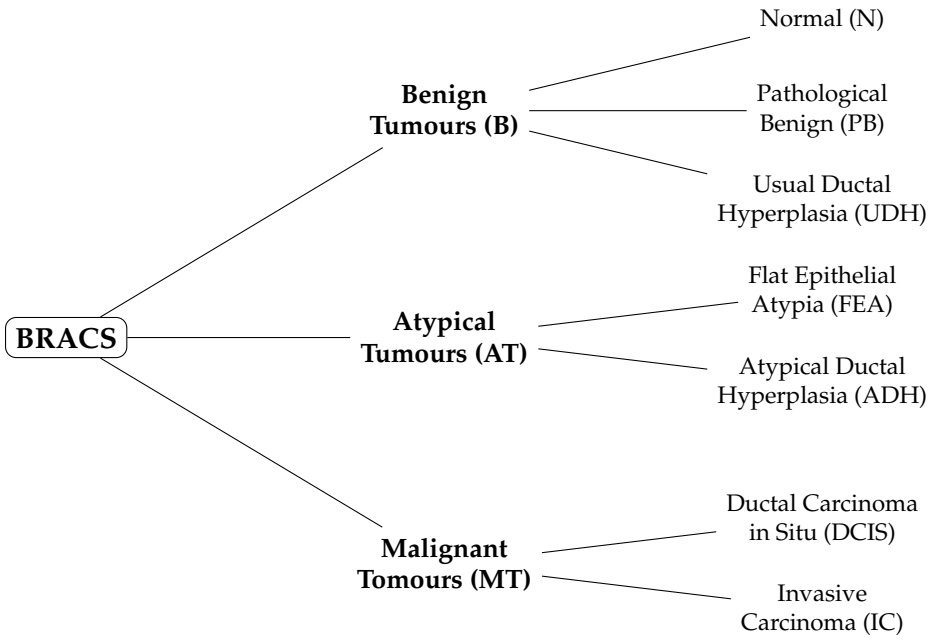

Figure 3.6: BRACS dataset label grouping.

Table 3.4: Structure of WSI images in the BRACS dataset

	Group_BT			Group_AT		Group_MT	
	N	PB	UDH	FEA	ADH	DCIS	IC
Training	27	120	56	24	28	40	100
Validation	10	11	9	6	8	9	12
Testing	7	16	9	11	12	12	20

Table 3.5: Structure of ROI images in the BRACS dataset

	Group_BT			**Group_AT**		**Group_MT**	
	N	PB	UDH	FEA	ADH	DCIS	IC
Training	357	714	389	624	387	665	521
Validation	46	43	46	49	41	40	47
Testing	81	79	82	83	79	85	81

CHAPTER 4
Approach to the problem

In this work, we will address the classification and segmentation of cancer images using various biomedical image datasets.

Cancer image classification involves assigning a label to each image that accurately indicates the presence or absence of different pathologies that may be present in the available images. Segmentation, on the other hand, refers to the task of delineating and labelling, pixel by pixel, the regions of interest within an image, providing detailed information about the location and extent of potential anomalies.

To tackle the problem, different architectures will be utilized to evaluate the performance of each with the available images. This includes the use of Convolutional Neural Network architectures such as ResNet or ConvNeXt, and for segmentation tasks, the U-Net architecture.

4.1 Classification models

The models were used directly from their Pytorch implementation or through the `timm` library. This library, whose name is short for PyTorch (torch) Image Models, is a collection of image models, layers, utilities, optimizers, and schedulers. The `timm` library offers an extensive range of pre-trained models, which simplifies the process of model selection and integration for various image processing tasks. Additionally, it provides numerous utilities and tools that facilitate model training, evaluation, and deployment.

These models, once imported, cannot be used directly for the problem at hand. To enable their use, the output layer must be modified to match the number of classes that need to be classified, and, if necessary, the input layer must also be adjusted. The modification of the input layer is required to ensure that the input channels correspond with the image channels. For instance, greyscale images require a single input channel, whereas colour images necessitate three input channels.

According to the Pytorch pre-trained models documentation[1] all utilized models have been pretrained using the ImageNet-1k dataset [20].

Among the models tested in classification tasks, the first ones belong to the ResNet family; specifically, the ResNet18 and ResNet50 models have been tested. In this family of models, the basic block is known as the Residual Block(Figure4.1). Each Residual Block consists of several layers, including convolutional layers and an activation function. The key feature of these blocks is the skip connection, also known as shortcut connections or residual connections. These connections enable the input signal to bypass certain layers

[1] https://pytorch.org/vision/stable/models.html

and be directly added to the output. This mechanism helps in maintaining the flow of information and also addresses to solve the vanishing gradient problem.

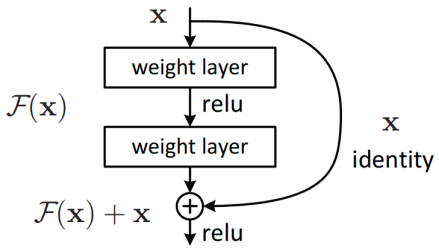

Figure 4.1: Residual Block. Image extracted from [7].

ResNet18 and ResNet50 models differ in the number of residual blocks and the complexity of these blocks. ResNet18 has a simpler architecture and fewer residual blocks compared to ResNet50, making it faster and more computationally efficient, but potentially less accurate.

Regarding DenseNet121, it is a deeper model than the previous two. Unlike ResNets, which combine skip connections through additive identity transformations, DenseNet uses concatenation to combine the outputs of each layer with the feature maps of previous layers.

Figure 4.2 shows a 5-layer block of a DenseNet where each layer captures the feature maps of all preceding layers.

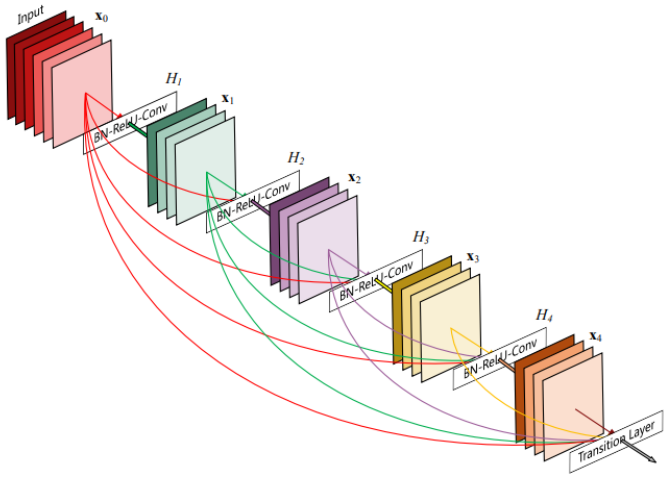

Figure 4.2: Dense Block. Image extracted from [21].

The forthcoming architectural framework to be examined is the ConvNeXt, which will undergo evaluation in three distinct configurations, each varying in size and parameter count (Figura 4.1). This investigation aims to assess the performance capabilities of ConvNeXt across its different scales. As discussed in the chapter dedicated to the state of the art, ConvNeXt architectures have shown promising results, potentially surpassing the performance of both ResNet and Vision Transformers (ViT).

In the case of the Visual Transformer (ViT) utilized, the model is available in different sizes: small, base, and large. Regardless of the model size, it requires input images with dimensions of 224×224. The ViT model operates by dividing these input images into smaller patches, specifically of size 16×16 pixels.

Table 4.1: Characteristics of the ConvNext models

Model Variant	No. Parameters
small	50.223.688
base	88.591.464
large	197.767.336

These patches are then flattened and linearly projected into a 1D vector to form a sequence of tokens. These tokens are used by the transformer encoder, which applies self-attention mechanisms to capture dependencies between the patches.

Table 4.2: Characteristics of the ViT models

Model Variant	Embed. dim.	Depth	No. Heads	MLP ratio
small	384	12	6	4
base	768	12	12	4
large	1024	24	16	4

4.2 U-Net architecture approach

A custom implementation of the U-Net architecture has been carried out, and various iterations have been conducted on it to find a topology that best suits the problem.

Most of these iterations have focused on the network's input layer, experimenting with different values for the kernel size of the convolutional layers and the spacing between kernel points (dilation) (Figure 4.3), as well as determining whether to add layers and transformations in the Skip Connections.

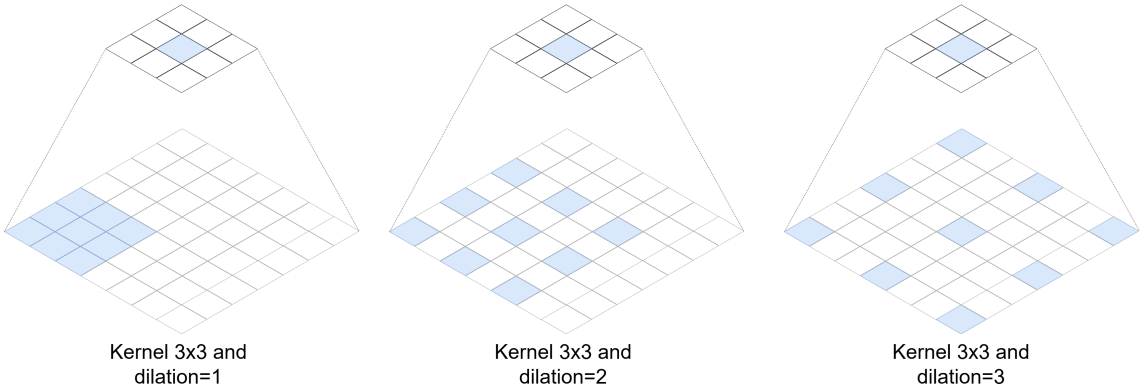

Figure 4.3: Conv2D Kernel Dilation.

The fundamental components of the various iterations include the Input Block, the Encoder Block, the Bottleneck, the Decoder, and the Skip Connections. For RGB images, the input channels of the Input Block will always be three, whereas for greyscale images, the input channels will be only one. Furthermore, all convolutional transformations have the padding parameter set to "same".

4.2.1. U-NetV1

This first iteration is very similar to the U-NetV2 that will be widely explained in the next section, but the skip connections were not correctly implemented.

This first iteration is named here only to maintain the correct correspondence with the later architectures used in the experiments.

4.2.2. U-NetV2

In the second iteration of the U-Net architecture, the Input Block consists of three 2D convolutions applied to the input images. Each of these convolutions generates an output with 32 channels, all sharing a kernel size of 3×3 and a stride of one. The difference between these three convolutions lies in the dilation parameter, which is set to 1 for the first convolution, 2 for the second, and 3 for the third.

These transformations are all applied to the same input; that is, a single batch of images is processed through this first block, producing three different outputs as a result of these transformations. These three outputs are then concatenated, forming an output with 32×3=96 channels.

Once the data have been processed by this Input Block, four Encoder Blocks are added with the following input and output channel configurations:

- Encoder Block 1 with 96 input channels and 128 output channels.
- Encoder Block 2 with 128 input channels and 256 output channels.
- Encoder Block 3 with 256 input channels and 512 output channels.
- Encoder Block 4 with 512 input channels and 1024 output channels.

Each Encoder sequentially applies the following transformations to the input batch:

1. MaxPool2d with a kernel size of 2.
2. Convolutional 2D transformation with the Encoder Block input channels as both input and output channels, a kernel size of 3 and the parameter groups equal to input channels.
3. Convolutional 2D transformation, with the input and output channels being the same as the Encoder Block and a kernel size of 1.
4. Batch Normalization 2D with the number of features parameter set equal to the Encoder Block output channels.
5. ReLu function.

Upon completing the processing through this sequence of Encoder Blocks, the data reach the Bottleneck Block, where four transformations analogous to those in the Encoder Block are applied first, following the same order and configuration.

After the ReLU corresponding to the fifth step of the encoder, the behaviour of the Bottleneck Block would be equivalent to returning to step 2 of the Encoder Block sequence after the ReLU. The Bottleneck Block in this first iteration has 1024 as both input and output channels.

After the Bottleneck, the next stage for the data in this architecture involves passing through four Decoder Blocks, with their input and output channels configured in the reverse order of the Encoder Blocks:

- Decoder Block 4 with 1024 input channels and 512 output channels.

- Decoder Block 3 with 512 input channels and 256 output channels.
- Decoder Block 2 with 256 input channels and 128 output channels.
- Decoder Block 1 with 128 input channels and `last_dec_out_ch` output channels.

In Decoder 1, the output channels are determined by a network configuration parameter called `last_dec_out_ch`, which is configurable and initialized to a default value of 128.

In this setup, each Decoder Block receives input from the previous block and from its corresponding Skip Connection. Specifically:

- Decoder Block 4 receives inputs from the Bottleneck and Encoder Block 4.
- Decoder Block 3 receives inputs from Decoder 4 and Encoder Block 3.
- Decoder Block 2 receives inputs from Decoder 3 and Encoder Block 2.
- Decoder Block 1 receives inputs from Decoder 2 and Encoder Block 1.

These Skip Connections do not apply any additional transformations to the given data; they solely facilitate communication of output features between the Encoder blocks and the Decoders blocks.

These Decoder blocks all have the same six transformations to the given inputs:

1. Apply an Upsampling Nearest 2D transformation with a scale factor of 2 to the input that comes from the previous block.
2. Concatenate the channels of the data from the previous step with the data coming from the corresponding Encoder Block.
3. Apply a Convolutional2D transformation with two times the Decoder Block input channels as the input channels for this convolution; and the Decoder Block Input Channels as output channels. This multiplication by 2 is applied to match the sum of the output channels from the previous block and the corresponding Encoder Block. This convolutional transformation has a kernel size of 3 and the parameter `groups` set to match the input channels of the Decoder Block.
4. Apply a second Convolutional2D transformation with input and output channels matching those of the current Decoder Block, using a kernel size of 1.
5. Batch Normalization 2D with the number of features parameter set equal to the Decoder Block output channels.
6. ReLu function.

The last component of this architecture is the Output Block. This block has input channels configured by the variable `last_dec_out_ch` from Decoder 1, and output channels equal to the number of classes in the dataset. This block applies the following transformations:

1. An Upsampling Nearest 2D transformation with a scale factor of 2.
2. Batch Normalization 2D with the number of features parameter set equal to the Output Block output channels.

3. Convolutional 2D transformation with the Output Block input channels as both input and output channels, a kernel size of 3*times*3.

4. Convolutional 2D transformation, with the input and output channels being the same as the Output Block and a kernel size of 1×1.

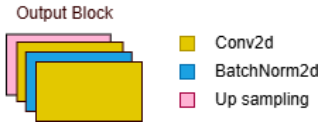

Figure 4.4: U-Net Architecture Output Block.

Summarizing, this output block is depicted in Figure 4.4, and comprises an Up Scaling layer with a scale factor of 2, followed by a convolutional layer with a kernel size of 3 and no dilation, a batch normalization layer, and finally, a second convolutional layer with a kernel size of 1 and no dilation.

4.2.3. U-NetV3

In this second iteration, a new block is added that applies transformations to the Skip connections.

These blocks consist of a single convolutional transformation with input and output channels identical to the Skip block, using a kernel size of 3, followed by Batch Normalization with the same number of features as the output channels of the current Skip block, and finally a ReLU activation.

In this second iteration, a total of four Skip blocks are instantiated with the same number of input and output channels, which correspond to the output channels of the Encoder and the input channels of the related Decoder, as follows:

- Skip Block 1 with 128 input and output channels, connecting Encoder Block 1 and Decoder Block 1.
- Skip Block 2 with 256 input and output channels, connecting Encoder Block 2 and Decoder Block 2.
- Skip Block 3 with 512 input and output channels, connecting Encoder Block 3 and Decoder Block 3.
- Skip Block 4 with 1024 input and output channels, connecting Encoder Block 4 and Decoder Block 4.

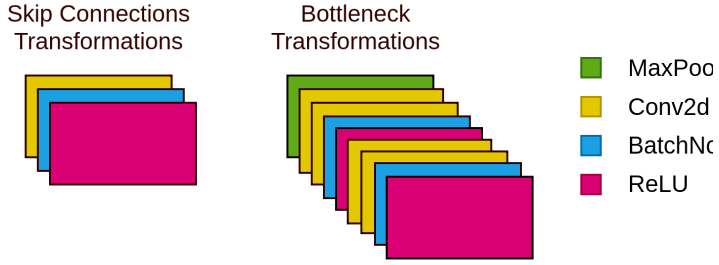

Figure 4.5: U-Net Transformations in the *Skip* and *Bottleneck* blocks.

The rest of the architecture remains exactly the same as in the first iteration. This second iteration serves to verify whether applying transformations to these connections benefits the model or not.

This modification in the skip connections is not further used in the two following versions.

4.2.4. U-NetV4

Is this version, the dilation configuration of the Input Block is removed. Instead, the three convolutions make use of three different kernel sizes (5×5, 7×7, 9×9) but with a dilation of 1.

These convolutions use the same input and output channels as in the previous versions, meaning 3 and 32, respectively.

The remaining changes applied in this iteration pertain to the configuration of the convolutional transformations and the final configuration of the network.

Firstly, for the Encoder Block, Decoder Block, and the Bottleneck, the convolutional transformations all use the same kernel size of 3×3.

Secondly, the final configuration of this version remains as follows:

- Input Block with 1 or 3 input channels (whether the image is greyscale or RGB) and 32×3 output channels.
- Encoder Block 1 with 32×3 input channels and 128 output channels.
- Encoder Block 2 with 128 input channels and 256 output channels.
- Encoder Block 3 with 256 input channels and 512 output channels.
- Encoder Block 4 with 512 input channels and 512 output channels.
- Bottleneck with 512 input channels and 512 output channels.
- Decoder Block 4 with 512 input channels and 512 output channels.
- Decoder Block 3 with 512 input channels and 256 output channels.
- Decoder Block 2 with 256 input channels and 128 output channels.
- Decoder Block 1 with 128 input channels and `last_dec_out_ch` output channels.
- Output Block with `last_dec_out_ch` input channels and `n_classes` output channels.

4.2.5. U-NetV5

In this iteration, modifications are applied exclusively to the Input Block, ensuring that all other components of the architecture remain consistent with those of version V4.

This final architecture employs an input block that concatenates four distinct convolutional transformations of the provided input, all without dilation. The input block concatenates four different convolutional transformations of the given input with no dilation. This input block comprises three input channels and produces 128 output channels, which are the concatenation of the output channels of the four transformations. These output channels have been weighted based on the kernel size of each transformation, as follows:

- Transformation 1 has a kernel size of 1 and 13 (10%) out channels. This kernel size focus on the colour of the pixel.
- Transformation 2 has a kernel size of 3 and 26 (20%) out channels.
- Transformation 3 has a kernel size of 5 and 38 (30%) out channels.
- Transformation 4 has a kernel size of 7 and 51 (40%) out channels.

The final architecture is shown if Figure 4.6. In this architecture, convolutional transformations are not applied within the Skip Connections, and the network's depth extends to 512 channels.

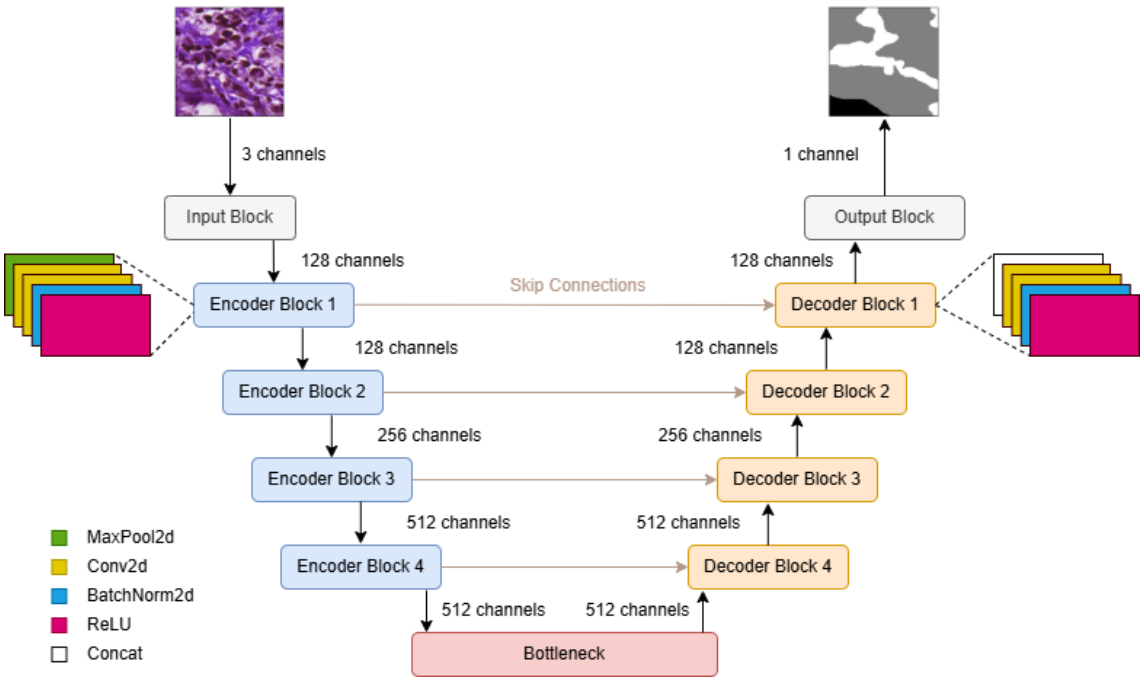

Figure 4.6: U-NetV5 Architecture.

4.3 Image Discretization

The various datasets used all come from histology images. One of the issues encountered with this type of patch-stained H&E images is the lack of heterogeneity in this staining [22]. To address this problem, image discretization is proposed. This discretization is performed using a K-means algorithm on the *RGB* values of the image pixels.

After this discretization process is completed, an image that was originally in the *RGB* colour space is converted into a greyscale image in which only K distinct values are represented.

This discretization will be tested at different levels to experiment whether this method helps solve the problem, worsens it, or ultimately does not alter the initial result. In particular, a series of tests are performed using images that have been discretized to various levels, specifically 4, 6, 8, 10, 12, and 14 levels. This methodological approach allows for an analysis of how different discretization levels affect the performance and accuracy of the image processing system. By examining a range of discretization levels, it is possible to determine the optimal level of granularity required for achieving the best balance between computational efficiency and the fidelity of the processed images.

CHAPTER 5
Experimentation and Results

5.1 Experiment Control and Management

The *Weights & Biases* tool is used [1] to oversee and manage the experimentation process. This tool serves as a software platform offering solutions for tracking and supervising experimentation in Machine Learning tasks, enabling efficient and effective experiment monitoring.

One of the primary advantages of *W&B* is its capability to track and visualize metrics in real-time. This encompasses a huge spectrum of model performance indicators, including but not limited to accuracy, loss, convergence rates, and gradient dynamics, among others. By providing real-time visualization, it facilitates researchers to swiftly identify and address any emerging experiment-related issues or anomalies.

Furthermore, in addition to real-time visualization, *W&B* enables the tracking of the hyperparameters adjusted in experimentation. By monitoring the interplay between hyperparameters and experimentation outcomes, researchers gain valuable insights into the complexities of model optimization.

Moreover, project administrators have control over experiment execution and can halt it if an internet connection is available on the machine where the experiment is running. This allows for stopping experiments if results are deemed unsatisfactory, whether due to model configuration or other reasons, freeing up the machine for subsequent experiments if automated in a queue, thereby saving time and computational resources.

Another feature is the default monitoring of system resource usage, enabling real-time tracking of metrics such as RAM usage, GPU usage and temperature, processor usage, etc.

Additionally, this tool allows for unlimited hours of experimentation tracking as long as the total size of files sent to it does not exceed 100 GB. Hence, if models themselves are not uploaded there and only metrics and experiment evolution are saved, this storage suffices for automatic result logging without capital investment.

In subsequent experiments, for each epoch of each trial, both the best model and the last model are saved. This enables access to the best model for testing on one hand, and availability of the last model to resume training from where it left off in case of training failure. If any failure occurs, the *Weights & Biases* tool allows for continuing to save metrics from the last checkpoint, specifying the identifier of the failed or interrupted training.

[1] https://wandb.ai/

Finally, a very interesting feature of this tool is the ability to monitor the metrics of new runs by forking from previous runs. This is useful, for instance, in the case where a model has been trained for a certain number of epochs and based on the metrics it may be possible for the model to learn better if trained for a longer period. With the fork, the metrics from a previous run are taken and different experiments can be launched from that point by varying the hyperparameters of the initial execution. For example, after having trained with a certain learning rate, different experiments can be launched with various reductions of this learning rate.

5.2 Experimentation for Classification

The following experiments were conducted on a computer equipped with an Intel i7-7700K CPU, 32GB of RAM, and an Nvidia GeForce GTX 1080, all running Python interpreter version 3.10.13, CUDA Version 12.4 and Pytorch 2.2.1.

In order to assure the reproducibility of all the experimentation, the function shown in the Figure 5.1 was called with the integer value 42.

```
def set_seed(seed: int):
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
    np.random.seed(seed)
    random.seed(seed)
```

Figure 5.1: set_seed function.

Furthermore, all the runs have applied data augmentation to the training subset. This augmentation was conducted by applying the following transformations to the input images:

- Horizontal flip with a 50% probability.
- Vertical flip with a 50% probability.
- Rotation of ± 90 degrees with a 50% probability.
- Color Jitter only for RGB images (not greyscale) with a 50% probability.

These transformations are the same as the applied to the experiments done in the Barbano *et al.* (2021) paper.

5.2.1. Experimentation with Unitopatho dataset

The primary objective of the experimentation was to replicate the results of the original paper [3], where this dataset is introduced, to establish a baseline.

This article details the experimentation on which the results are presented.

First, two subsets are established, specified by the accompanying `.csv` files included with the data download. These files divide the dataset into training and testing subsets, representing 70% and 30% of the total, respectively.

The results presented for the different image scales are the outcome of training a ResNet-18 pretrained with ImageNet for 50 epochs, using an SGD optimizer with an initial learning rate of 0.01, reduced by a factor of 0.1 every 20 epochs.

The images used in this training are resized to a dimension of 224×224 pixels.

These initial tests are conducted on images of 800μm and 7000μm, as these are the only available, using the "top-label" as the target, which includes the six classes defined in Figure 3.1.

Table 5.1: Performance metrics of Unitopatho

Original metrics

	Patch scale σ [μm]	
Type	800	7000
BA (6-class)	0.45	0.37
NORM	0.66	0.78
HP	0.92	0.60
TA	0.66	0.76
TVA (HG+LG)	0.67	0.84

Metrics of our first experimentation

	Patch scale σ [μm]	
Type	800	7000
BA (6-class)	0.45	0.35
NORM	0.68	0.78
HP	0.93	0.60
TA	0.71	0.69
TVA (HG+LG)	0.75	0.73

As shown in the Table 5.1, the results are quite similar, although in some cases, this initial experimentation yields better outcomes for images of 800μm than for those of 7000μm. This discrepancy may be attributed to the only parameter not specified in the article, which is the seed for the random processes.

Building upon this initial experiment, various architectures and hyperparameter variations have been tested to explore potential enhancements to the original data. The architectures investigated include traditional ones such as *ResNet18*, *ResNet50* and *DenseNet121*.

Subsequently, experimentation extended to more modern architectures such as ConvNeXt [14] and Vistual Transformers (ViT) [10].

The ViT models have different versions that depend on the model size, input image size, and patch size. Specifically, in the experimentation conducted for this work, the ViT model that accepts 224×224 pixels images as input was used and makes patches of 16×16 pixels.

For all the models employed, and to adhere to the methodology, pre-trained models on ImageNet1k are utilized. Additionally, these experiments will deal with both colour and greyscale images, meaning our input images will possess either three or one channel depending on their colour type. Greyscale images will result from discretizing the input images into K levels. Specifically, discretization have been tested over 4, 6, 7, 10, 12 and 14 levels of greyscale.

For these two reasons, the utilized networks require modification of both the first and the last layer. The modification of the first layer will pertain to the input channels and will be adjusted to correspond to the colour channels of the input images. On the other hand, the modification of the output layer refers to the number of classes targeted.

In the Unitopatho dataset, this output depends on how we configure the labels of the images. Three different approaches are distinguished:

- **Top Label**: In these tests, the classifier is trained to correctly distinguish between the 6 different types of polyps.
- **Type**: The number of classes to classify is reduced from 6 to 4, i.e., attempting to classify polyp groups without considering their level of dysplasia.
- **Grade**: In this type of test, a binary classifier is trained to simply discern the level of dysplasia between *high* or *low*.

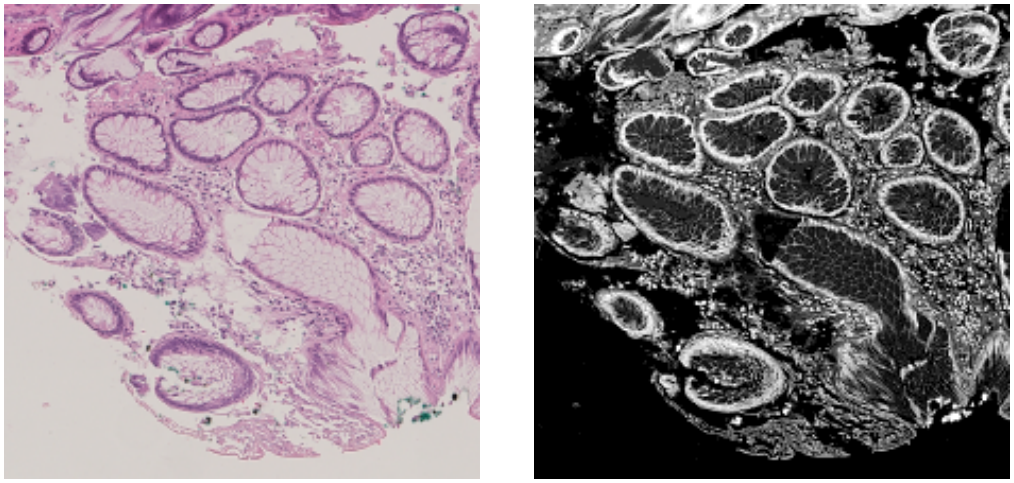

Figure 5.2: Example of an original RGB image and its K14 discretization.

Finally, in the experiments, the metric called balanced accuracy will be mentioned numerous times. This metric is calculated using its implementation from the `scikit-learn` package, which mathematically depends on sensitivity and specificity, calculated as follows, where TP stands for True Positives, TN for True Negatives, FP for False Positives and, lastly, FN for False Negative:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

Unitopatho experimentation on *Grade* target

In these experiments, a binary classifier has been trained to differentiate between the two possible grades of dysplasia, high and low. The graph in Figure 5.3 provides a summary of the results. The bars represent the average accuracy achieved by the models overall across the different versions of the experimentation. The lines depict the maximum accuracy achieved by each architecture.

In this graph, it can be observed that, regardless of the architecture, the 800μm images consistently outperform those of 7000μm in discriminating between high and low grades of dysplasia. Furthermore, it is also evident that, on average, all tested architectures, with the exception of ConvNeXt small and base, maintain test accuracy around 75%. The small ViT model achieves the highest accuracy at 82%, 5% higher than the ResNet18 architecture. While this improvement is not substantial, it is noteworthy that, in the case of ViT, this maximum accuracy was achieved at epoch 2, whereas for ConvNeXt, it was at epoch 10, indicating overfitting beyond this point. The overfitting was so pronounced that in the case of ViT, by epoch 10, the training accuracy surpassed 95%, and by epoch 30 it exceeded 99%.

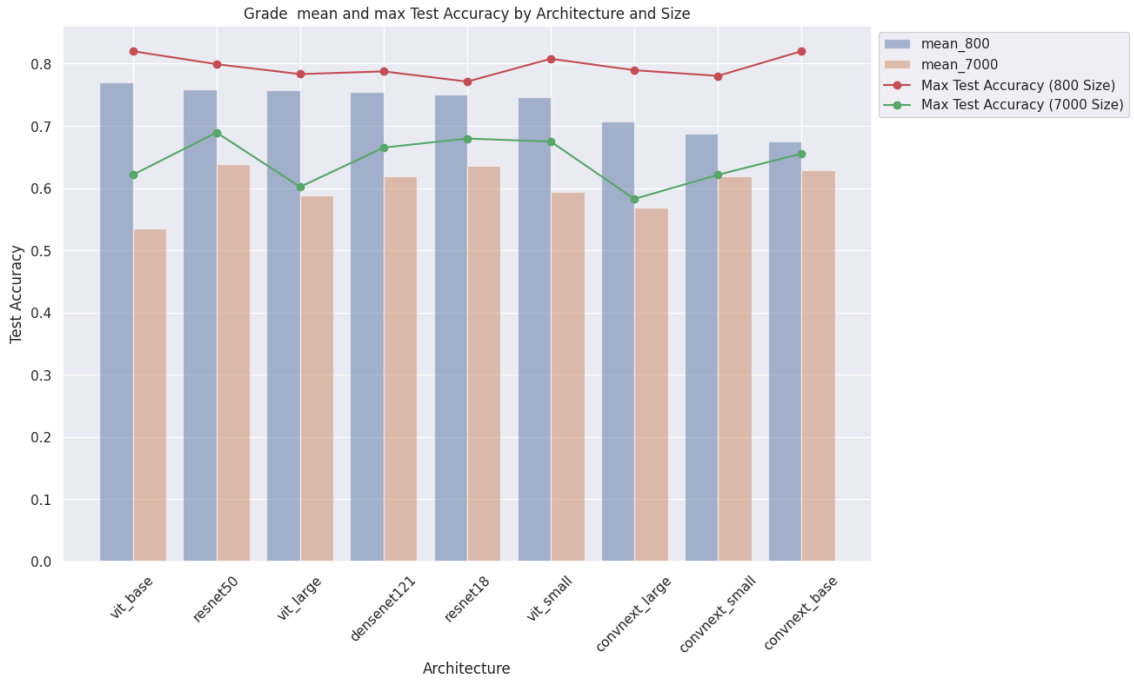

Figure 5.3: Grade Mean and max test accuracy by architecture and size.

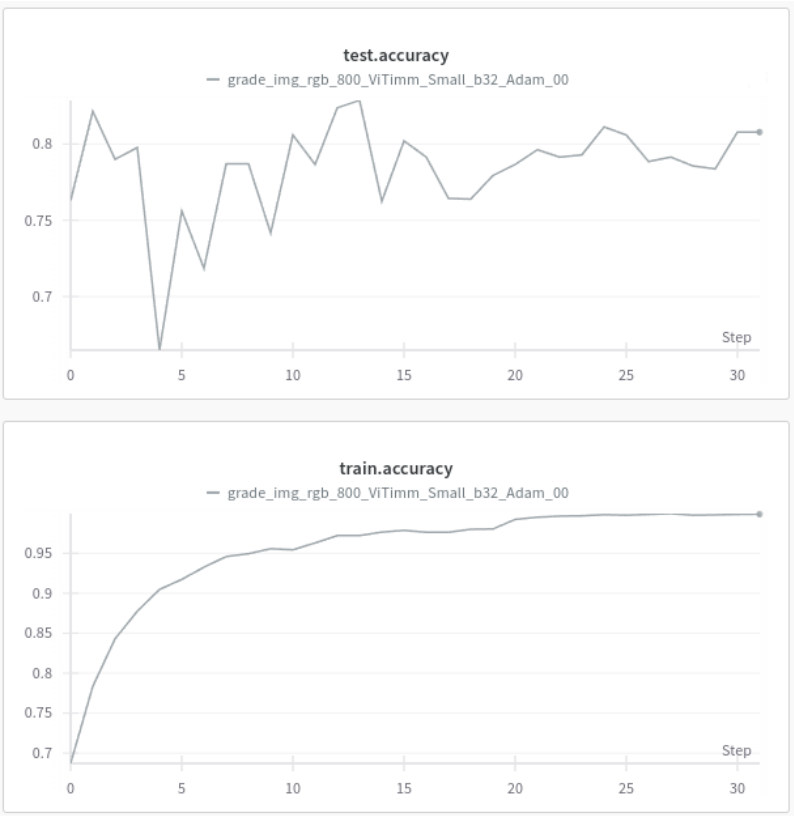

Figure 5.4: Unitopatho Target Grade, ViT Small training plot.

Another significant difference observed in the experimentation is the superiority of ViT over ConvNeXt in terms of computational time. While a ConvNeXt Base takes 3 hours and 17 minutes to complete 50 epochs, the ViT Base completes the same epochs in just 34 minutes, resulting in a training process that is 4.79 times faster in this case.

Table 5.2: Summary Data from Unitopatho Grade experimentation

Architecture	Mean (800)	Max (800)	Mean (7000)	Max (7000)
vit_small	0.75	0.82	0.59	0.67
convnext_base	0.67	0.82	0.63	0.66
resnet50	0.76	0.80	0.64	0.69
convnext_large	0.71	0.79	0.57	0.58
densenet121	0.76	0.79	0.62	0.67
vit_large	0.77	0.78	0.59	0.60
convnext_small	0.69	0.78	0.62	0.62
vit_base	0.77	0.77	0.54	0.62
resnet18	0.75	0.77	0.64	0.68

With regard to the discretized images, no improvement has been observed in their usage compared to ViT and ConvNeXt models with RGB images. However, ResNet architectures exhibit very similar performance to that achieved with ViT and ConvNeXt models. Specifically, the best model trained with discretized images at four levels (K4) achieves a test accuracy of 81%, which is the highest among all different groups of discretized images. Following the ranking by test accuracy, the next best training using discretized images has been with 14 levels (K14) achieving 79.5%, both runs trained with a ResNet50 architecture with the Adam optimizer, a learning rate of 0.001 and a weight decay of 0.01.

Unitopatho experimentation on *Type* target

In this experimentation, the aim is to classify the four different types of tumours (or absence thereof), namely, Normal, Hyperplastic Polyp, Tubular Adenoma and Tubulo-Villous Adenoma.

In this section, the rationale behind Barbano et al. (2021) focusing the first classifier on Hyperplastic Polyp with 800μm images has been confirmed. The Balanced Accuracy for this class alone is around 95%, whereas for the rest of the classes, the values of this metric range between 60 and 75%.

On the other hand, the Hyperplastic Polyp, when analysed with the ViT Large model and a resolution of 800μm, achieved the highest balanced accuracy among the target classes at 92.13%. The remaining classes were classified with a balanced accuracy below 75%.

For the remaining classes, similar to the modelling approach in Barbano et al. (2021), they are better classified with 7000μm images. In this experimentation, discretized images have played a more prominent role, discerning Normal tissues.

Table 5.3: Best Results for Type NORM with 7000μm images

Architecture	K	NORM BA
ResNet50	14	0.82
ResNet18	10	0.82
ResNet50	4	0.81
ResNet18	8	0.79
ResNet50	6	0.78
ResNet18	RGB	0.78

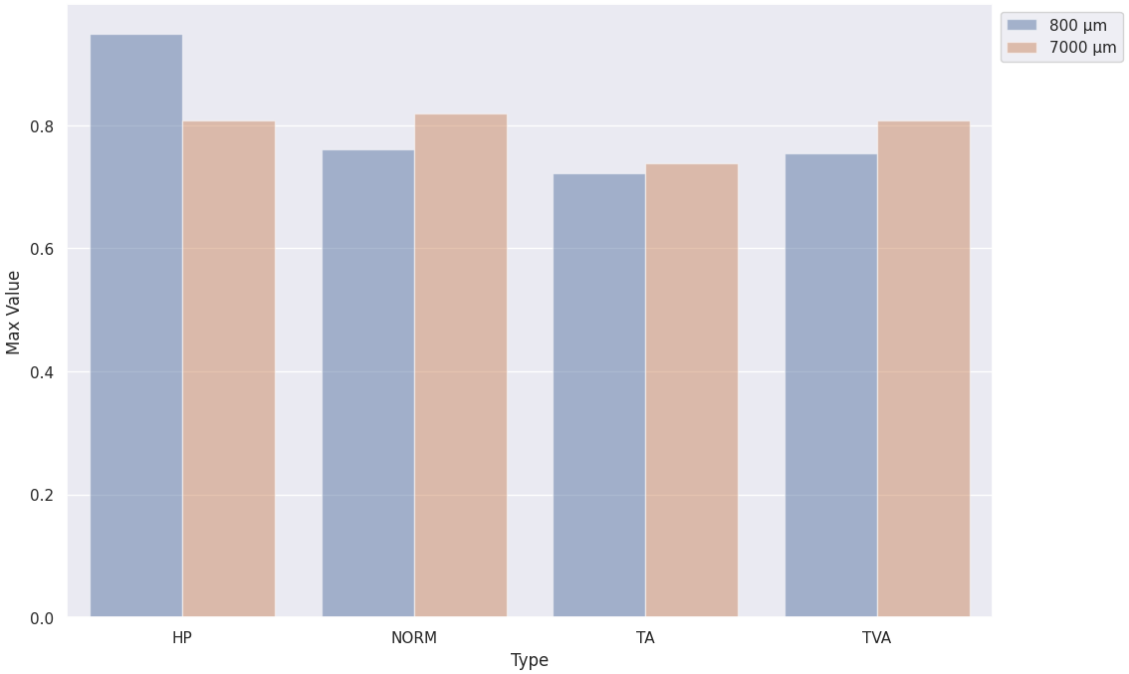

Figure 5.5: Type max test balanced accuracy by input images resolution

In the results for the best class-wise classification, convolutional models appear to outperform the Vision Transformer (ViT) models. As Dosovitskiy et al. (2021) [10] suggest, ResNets exhibit more inductive biases than the transformer architecture applied to images, and the latter requires more data to learn useful representations.

Table 5.4: Best Results for Type TV with 7000μm images

Architecture	K	TV BA
Densenet121	14	0.74
ResNet18	RGB	0.74
ResNet18	6	0.73
ResNet18	4	0.72
ResNet50	8	0.72
DenseNet121	12	0.72

As illustrated in Figure 5.5, the model achieves superior performance in distinguishing tubular tumour types when utilizing images with a resolution of 7000μm. This enhanced performance can be attributed to the inherent characteristics of tubular tumours, which could be displaying more distinguishable structures at a macroscopic level. The larger field of view provided by these images captures these morphological features more effectively.

On the other hand, tubular tumours are the most highly represented. It is worth noting that for the Tubular Adenoma type, there are 509 images at 7000μm and 4072 at 800μm, and for the Tubular-Villous Adenoma, there are 225 and 3102 images at the respective resolutions. This is well above the representation of NORM and HP. Thus, in principle, with more data, the model should be able to better learn the characteristics of this type of tumour, although this is not the case. This could suggest that this type is difficult to classify based on the provided images.

Table 5.5: Best Results for Type TVA with 7000μm images

Architecture	K	TVA BA
ResNet18	RGB	0.81
DenseNet121	4	0.80
ViT Base	RGB	0.78
ResNet18	14	0.78
ResNet50	10	0.77
ResNet18	8	0.77

Unitopatho experimentation on *Top Label* target

The objective of this experimentation is to verify the performance of the model in attempting to classify the six different classes, specifically the variations in cancer and its subtypes by degree of dysplasia. This task is the most complex, with test accuracy not exceeding 50% in any of the conducted tests, and the maximum balanced accuracy in the test is 51.36%. Figure 5.6 depicts these metrics for the best training based on Test Balanced Accuracy; training with a ResNet18 and images discretized at 800μm into 6 levels (K6).

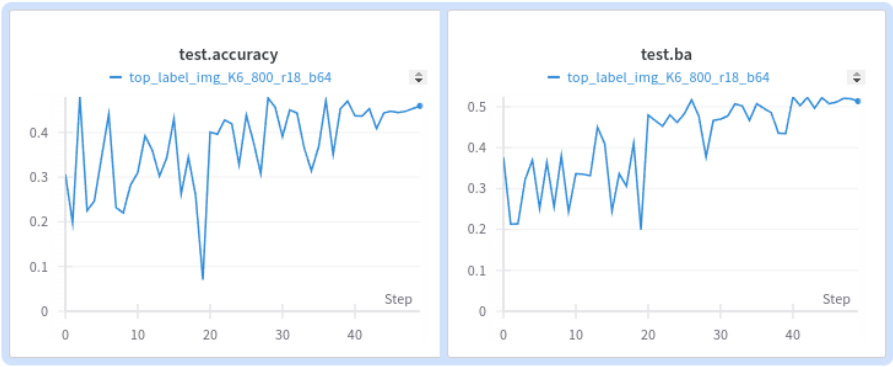

Figure 5.6: Test Accuracy and Test Balanced Accuracy from best Top Label experiment.

The decrease in performance is primarily caused by the inclusion of subtype classification based on the degree of dysplasia, although these metrics alone can be misleading. If we examine Figure 5.7, the balanced accuracy per class, for class 0 (HP), shows similar results to the previous test with balanced accuracy results above 90%. The rest achieve results above 50%, although classes 4 and 5 (corresponding to TVA.HG and TVA.LG) are the ones where the model struggles the most to differentiate accurately. This can be observed in the significant fluctuations in the metric during training.

About the experimentation with Visual Transformers

To enable the utilization of pretrained models, the Python package `timm` was employed. This package facilitates the loading of Vision Transformer (ViT) models pretrained on ImageNet1k across various sizes. Nonetheless, training the entire model sometimes led to overfitting within a few epochs, compounded by the issue that it did not perform adequately during testing; rather, the test loss increased instead of decreasing.

In an attempt to address this, experiments were conducted by solely training the classifier while keeping the remainder of the network fixed, and by training both the classifier and the first convolutional layer responsible for Patch Embedding, while still freezing

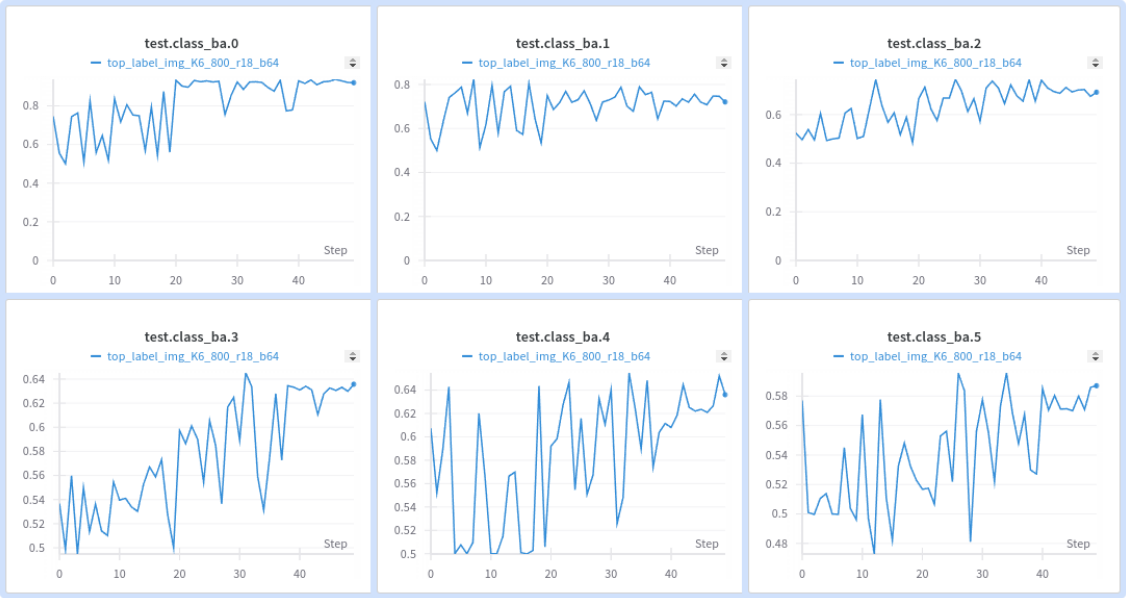

Figure 5.7: Test Balanced Accuracy per class from best Top Label experiment.

the rest of the network. This approach resulted in more consistent training, but it did not achieve results as good as those obtained with the full model training.

The "Type" target benefited the most from this approach; however, it did not achieve better performance compared to the ResNets, attaining only 69.34% on the balanced accuracy metric.

These results, as well as those from training the full network, could potentially be enhanced by using pretrained ViT models on an even larger dataset. According to the findings of Dosovitskiy et al. (2021) [10], the best results for this architecture were achieved by pretraining the model with ImageNet-21k [12], a dataset consisting of two versions:

- **Fall 11**: with 11,221 classes, 11,797,632 training images and 561,052 test images.
- **Winter 21**: with 10,450 classes, 11,060,223 training images and 522,500 test images.

The dataset size exceeds one Terabyte of storage, and according to Dosovitskiy et al. results, training on this dataset using a TPUv3 with 8 cores would take approximately 30 days. Given hardware and time constraints, pretraining on this dataset is not feasible within the scope of the current study.

5.3 Segmentation Experimentation

For the following experiments, semantic segmentation was conducted on two breast cancer pathology datasets. As described in the 'Approach to the problem' chapter, several versions of the U-Net architecture were used.

In Figure 5.8 shows how the use of skip connections affects the model. In this example the model is being trained with the BCSS dataset. It is clear that these connections provide the model with valuable information, helping it to learn to distinguish the characteristics of different classes.

That figure shows that without skip connections, the model only achieves a 7% validation IoU metric within a few epochs and appears to face a plateau in the IoU, precision, and loss metrics.

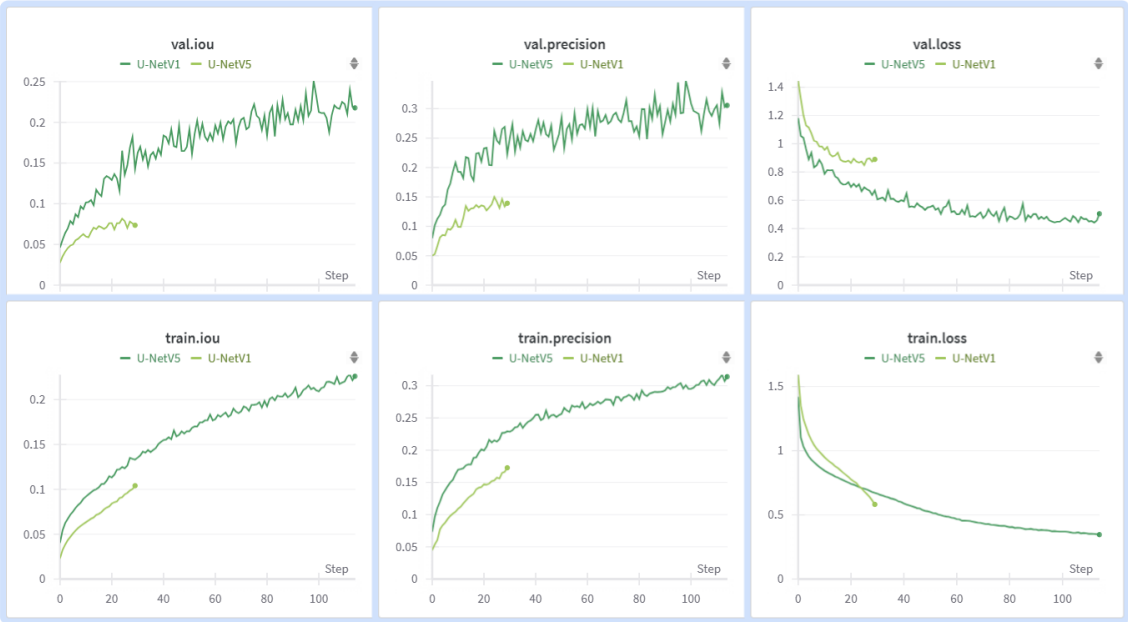

Figure 5.8: Metrics from experiments with and without skip connections. *As referred to in the 'Approach to the Problem' chapter, the U-NetV1 does not have skip connections implemented, while the U-NetV5 does.*

For the segmentation runs, the metrics employed include loss, Intersection Over Union (IOU) per class, precision per class, and recall per class. The IOU, precision, and recall are calculated as shown in Algorithm 5.1.

5.3.1. Experimentation with BCSS dataset

The authors of this dataset provide images along with their corresponding masks for easy download at a resolution of 0.25 MPP (Microns Per Pixel). The download size is suitable for processing on any machine, with a total weight of 4.9 GB. These images range in size from approximately 2000 to 5000 pixels on each side and do not exhibit uniform dimensions.

The challenge associated with this dataset lies in attempting to discern as accurately as possible among the 22 classes present in its labelling.

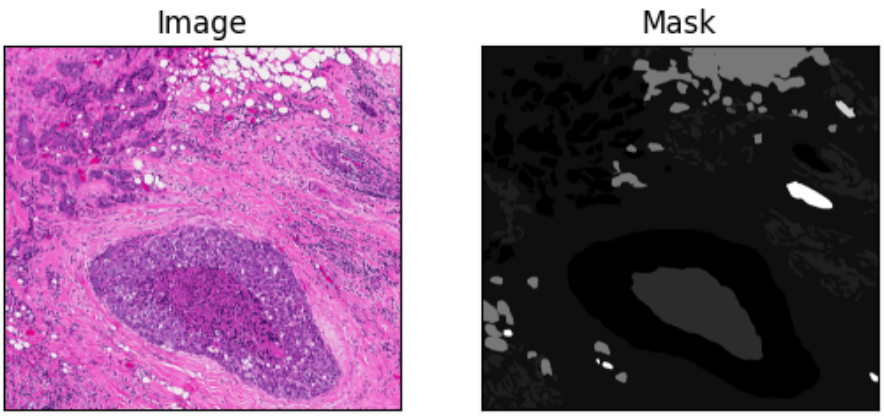

Figure 5.9: Image and Mask example from BCSS dataset.

Algorithm 5.1 Segmentation Metrics

```
for each epoch do
    iou_per_class ← 0⃗
    precision_per_class ← 0⃗
    recall_per_class ← 0⃗
    counter_per_class ← 0⃗ + ϵ
    for each batch do
        for each n from 0 to the length of batch size do
            for each i from 0 to the number of classes do
                true_count ← sum of elements in masks[n] == i
                pred_count← sum of elements in predicted[n] == i
                intersection ← sum of elements where predicted[n] and masks[n] == i
                union ← pred_count + true_count
                if union > 0 then
                    counter[i] += 1
                    iou_per_class[i] += intersection / (union - intersection + epsilon)
                    recall_per_class[i] += intersection / (true_count + epsilon)
                    precision_per_class[i] += intersection / (pred_count + epsilon)
                end if
            end for
        end for
    end for
    iou_per_class ← iou_per_class / counter
    precision_per_class ← precision_per_class / counter
    recall_per_class ← recall_per_class / counter
end for
```

To work with these images in as much detail as possible, and given the non-uniformity of their sizes, which prevents them from being rescaled without distorting their content, they have been divided into non-overlapping fragments of 256×256 pixels. This cropping method yields a total of 37,141 images.

Another characteristic of this dataset is that both images and masks are stored in the same directory without specifying which images are used for the training, testing, and validation subsets. Therefore, during training, the data is partitioned, with 80% of the data allocated for training and the remaining 20% evenly split between validation and testing. Subsequently, a text document is stored containing the paths of the images used in each subset, facilitating independent validation of the model if necessary, separate from the training process.

The initial experiments were conducted without applying any form of data augmentation. These tests revealed that the network was capable of learning from the provided images and masks but struggled to generalize (see Figure 5.10).

In an attempt to address this issue, data augmentation techniques were applied to the training images. These augmentation techniques involve applying a range of random transformations to the original images and their associated masks.

To ensure that the mask information was not corrupted during certain transformations, the following precautions were taken:

- Horizontal flip with a 50% probability.
- Vertical flip with a 50% probability.
- Rotation of ± 90 degrees with a 50% probability.

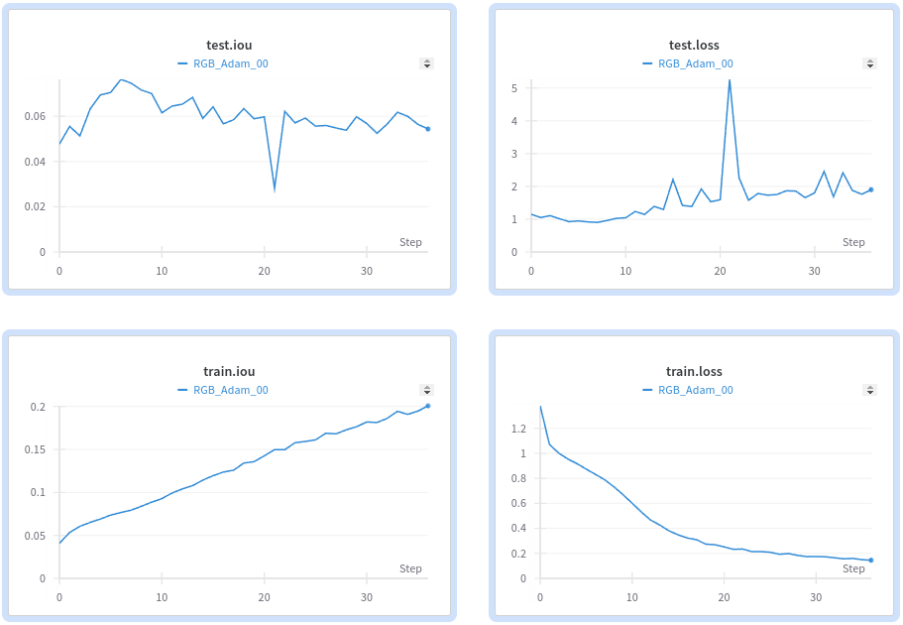

Figure 5.10: Rapid overfitting in training without data augmentation.

These transformations, if employed, are applied both to the image and the mask at the same time. By introducing randomness into the augmentation process, the model is exposed to a more diverse set of scenarios, helping it to better generalize to unseen data. The improvement in the training phase is shown in Figure 5.11.

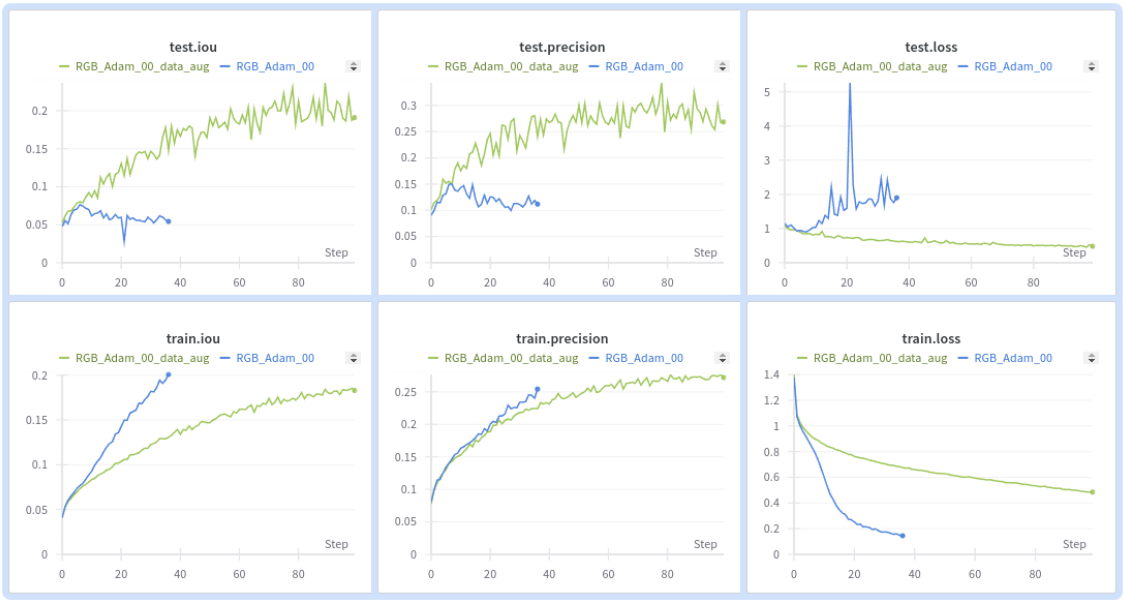

Figure 5.11: Training without data augmentation vs training with data augmentation.

By applying the data augmentation techniques, the U-NetV3, V4 and V5 were trained for 100 epochs and the results showed that the V5 had a slightly better performance. The validation IoU metric for this training is shown in Figure 5.12.

Given this better performance, the U-NetV5 model was trained for 200 total epochs. The training process lasted for 3 days and 15 hours, ultimately achieving an Intersection over Union (IoU) score of 30%.

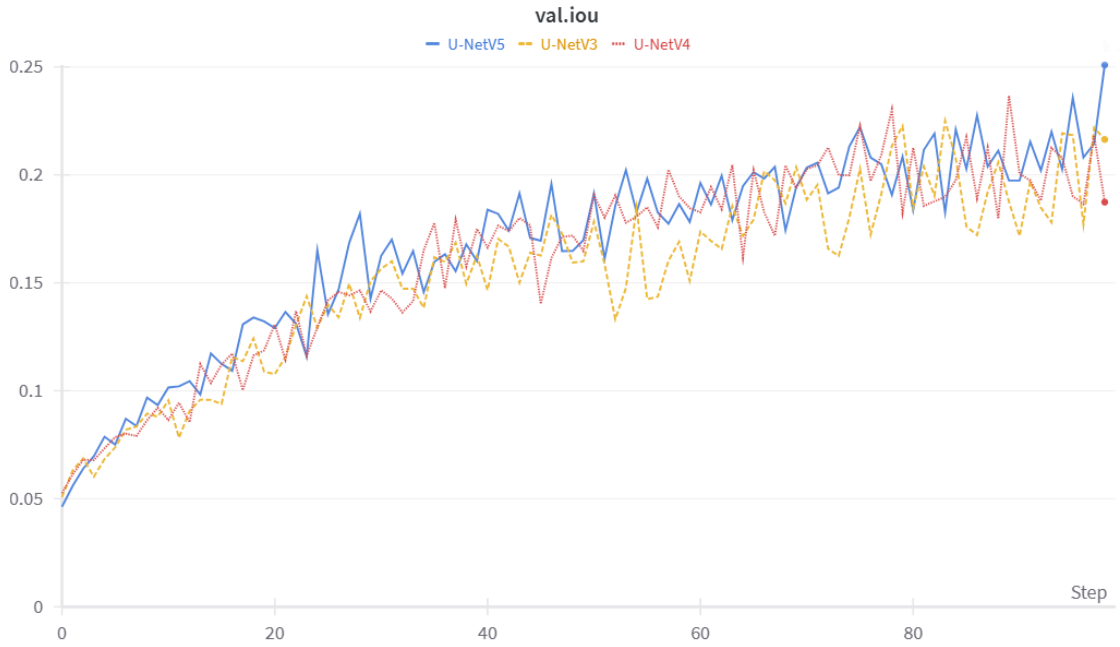

Figure 5.12: U-Net versions validation metrics comparison.

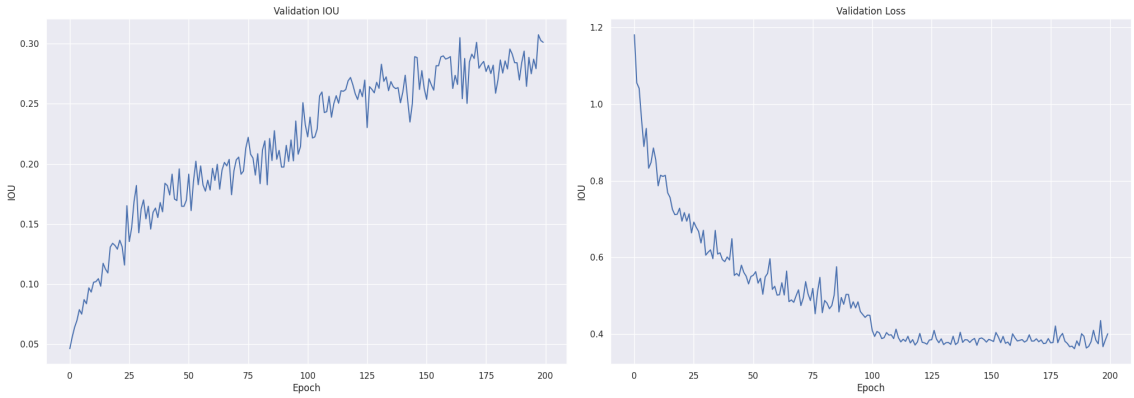

Figure 5.13: Validation IOU and loss from BCSS Training with U-NetV5.

After these 200 epochs, the model ceased to improve and appeared to have reached its peak learning capacity, with the loss exhibiting minor oscillations towards higher values in the final epochs. It should be noted that the IoU metric represents the mean IoU across all classes.

Out of the 22 classes present in the dataset, the model failed to classify the "Nerve" and "Other" classes, achieving an IoU of 0% for both. The poor performance on the "Nerve" class is likely due to its very low representation in the data. The "Other" class, being a generic category, encompasses a wide variety of elements, making it difficult for the model to identify consistent patterns.

Another class that adversely affects the metrics is "Undetermined" which, like the "Other" class, includes a very broad category.

An important point is that the highest IoU score, 67%, was achieved with the "Tumour" class, followed by "Necrosis Or Debris" with 59.88%. These are perhaps the most relevant and interesting classes for a pathologist assisting in cancer detection.

Given these circumstances, the predictions on the test set were examined, and the generated masks largely reflect the original masks. An example of these masks predicted by the model can be seen in Figure 5.15.

A small test with the U-NetV5 was conducted with discretized images for K8, K10, K12, and K14. This test showed that this type of image can perform similarly to RGB images, but with slightly worse performance.

Figure 5.14: BCSS Test with discretized images using U-NetV5.

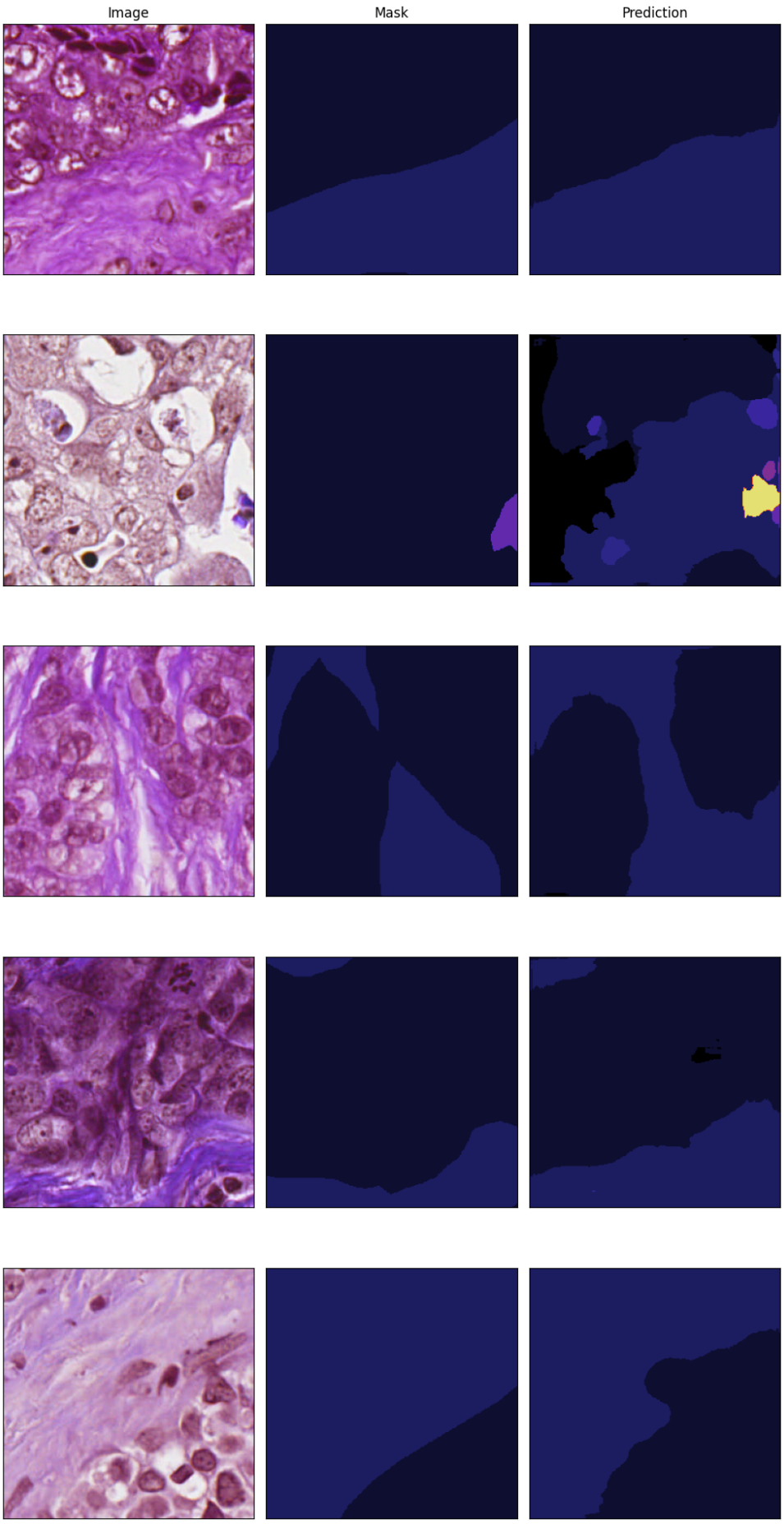

Figure 5.15: Mask predictions for images from the test set of the BCSS dataset.

5.3.2. Experimentation with the BRACS dataset

For this dataset, access was provided to a machine equipped with hardware suitable for handling this data. The machine was outfitted with a 13th Gen Intel Core i7-13700K, an NVIDIA GeForce RTX 4090 GPU, and a 4TB hard drive.

In contrast to the previous dataset, BRACS only provides Whole Slide Images (WSIs) and their annotations (Regions of Interest (ROI) are also available, but without annotations), necessitating a different approach to the problem. The primary issue posed by such a dataset is the storage space required for the images and their labels, which can demand up to 1TB of disk memory.

The annotations for this dataset are provided in the .qpath format, which needs to be read using QuPath[2], an open-source software for biomedical image analysis.

These labels cannot be directly read from Python. There are several alternatives for working with these labels; one is to use Jython, a Python implementation for the Java Virtual Machine. The limitation of this implementation is that it cannot work with extensions written in C, so tools like Numpy [3] cannot be used.

Another alternative is to use **QuPath's** own scripting language to export these labels to the `.geojson` format, an open standard format designed to represent geographical features along with their non-spatial attributes. This is the only format to which **QuPath** allows exporting.

The chosen option is the latter, as it offers greater flexibility and provides access to the entire Python ecosystem and its libraries without relying on a Java environment.

With the annotations available in the appropriate format, the next step is to create the image masks. At this stage, the first issue with the data arises. The annotations are not standardized correctly among the group of pathologists who performed the labelling, both in terminology and form.

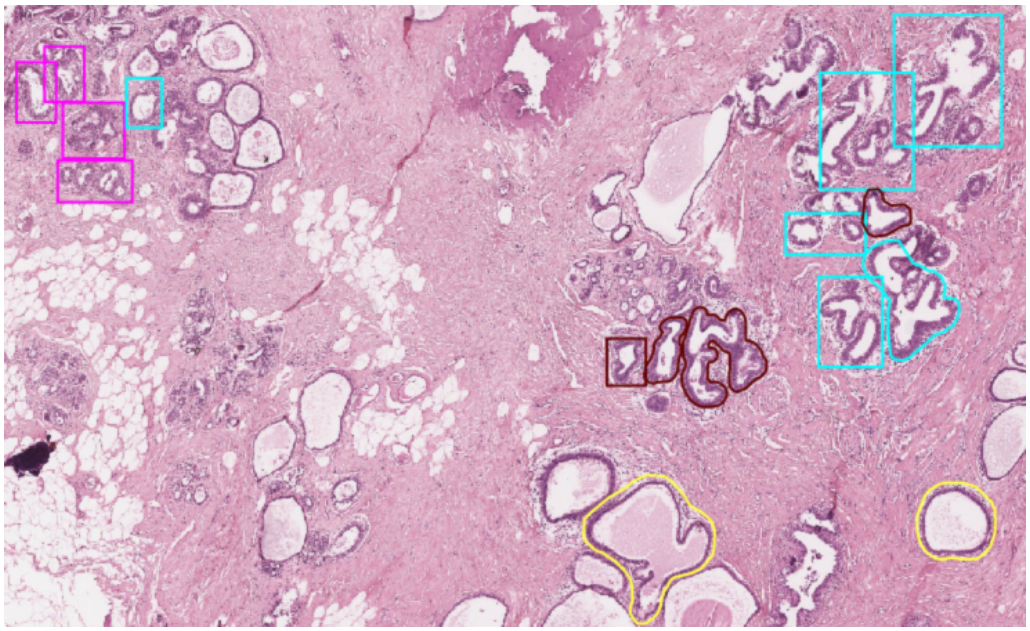

Figure 5.16: Annotations example from BRACS dataset visualized with QuPath.

[2] https://qupath.github.io/
[3] https://numpy.org/

In terms of form, there are annotations suitable for segmentation tasks, which correctly delineate the area of the specified class, while other annotations are simple rectangles. These rectangles can introduce noise into the model by indicating, within the same area, pixels that belong to a certain class and pixels that potentially do not belong.

These disparities in form and annotation method can be observed in Figure 5.16, which pertains to the image with identifier 1271 within the validation subset of this dataset; specifically, it belongs to the subgroups `Group_AT/Type_ADH`.

Regarding terminology, out of the eight classes (the seven specified in Figure 3.6 plus the background) indicated in the dataset, there are up to 17 different terms for these classes. These terms have been assigned an identifier to group them according to the original seven categories. As shown in Table 5.6, these identifiers range from 1-7, leaving identifier 0 for sections that are of no interest or background.

Table 5.6: Assignment of identifiers to the different terminology of the annotation set

Term	Identifier
ADH	5
ADH-sure	5
BENIGN	1
Benign sure	1
Benign-sure	1
DCIS	6
DCIS-sure	6
FEA	4
FEA-sure	4
MALIGNANT	7
Malignant	7
Malignant-sure	7
Pathologica benign	2
Pathological-benign	2
Pathological-benign (Benign-sure)	2
UDH	3
UDH-sure	3

These identifiers will be used as values for the pixels that contain the corresponding label.

Two approaches were followed for the creation of the masks. First, for each image, the coordinates of its annotations were checked, and the minimum and maximum x and y points were saved. The area delimited by these coordinates will be used as our image, discarding the rest; it is important to note that the WSIs in this dataset can easily exceed 100,000×100,000 pixels.

To ensure the actual correspondence between the original image and its mask, the mask is created over the original dimensions of the image in a two-dimensional *array*, where each element represents a label as specified in Table 5.6, initialized with all values set to 0. The coordinates of this array corresponding to areas that need to be annotated are assigned the value of the corresponding label.

Once this mask is created, 256×256 pixels crops are made, without overlap, within the area delimited by the aforementioned minimum and maximum x and y points. These coordinates are adjusted to ensure that the crops are precise and all have the desired dimensions.

This approach, even when using only the portion of the image that encompasses all its labelled areas, generates over six million crops. This large number of images presents several challenges. Among them is the time limitation; training the implemented architecture for 50 epochs can take more than a week, so training for 200 epochs as with the previous dataset would require a month of computation.

Additionally, these images greatly unbalance the dataset, resulting in the vast majority of images lacking labels of interest. To quantify, only 355,429 out of the more than six million images contained labels other than zero.

Using this approach, tests were conducted by training the network with all the images combined. This did not yield good results due to the severe imbalance between images with labels of interest and those that could be categorized as "background", resulting in an Intersection over Union (IoU) close to 0%.

To address this, another perspective was adopted: for each batch of each epoch, the model was fed using images with non-zero labels and images with all mask pixels set to zero in a 50%-50% proportion. The idea was that the model should also learn to distinguish areas where nothing should be labelled from those that should be labelled. Unfortunately, this did not introduce significant improvement either. The possible explanation for this is that, since these crops were derived from large images, they might represent such microstructures that they were not useful for the model's learning.

The second approach involved working with the entire WSI and its different magnification levels. Each WSI file in this dataset offers four magnification levels, numbered from 0 to 3, with level 0 corresponding to the original dimensions of the WSI and level 3 providing a general, lower-dimension image. Working with the latter level yields images approximately $2000px$ per side. This level was chosen for further processing, and 256×256 pixels crops with overlap were made from it. This overlap involved moving the cropping window by $128px$ instead of $256px$ after each crop. The reason for this is that, given the smaller source image, it is necessary to increase the number of generated images. Even so, the resulting images are significantly fewer, thus reducing the training time per epoch considerably.

Additionally, upon analysing the masks, an additional issue becomes apparent, as previously mentioned and reflected in Figure 5.16, regarding the non-homogeneous and possibly inconsistent annotation method. The non-homogeneity stems from some labels accurately reflecting clear areas by delineating their contours, while other annotations cover the area of interest with a rectangle, which is not as precise as outlining the exact area of the labelled element.

The mentioned possible lack of consistency arises from observing masks where a specific area is delimited by a particular colour in the image, but it is not fully selected. This might be appropriate for an expert pathologist, although it raises doubts at first glance. Furthermore, when viewing the crops and their masks, it becomes more evident that the labelling is either incomplete or has not been thoroughly reviewed. As seen in Figure 5.17, a type of structure in the image is labelled, but not all similar structures within the same image are labelled.

Again, while an expert pathologist might argue that this labelling is correct, it does not appear so at first glance, and it is likely that this type of annotation introduces noise into the model.

Using the crops from this second approach, various experiments were conducted with different configurations of the implemented U-Net. In these initial tests, it was observed that while the network was capable of learning, it failed to generalize. That is, the training metrics showed positive progression, which was not reflected in the validation metrics.

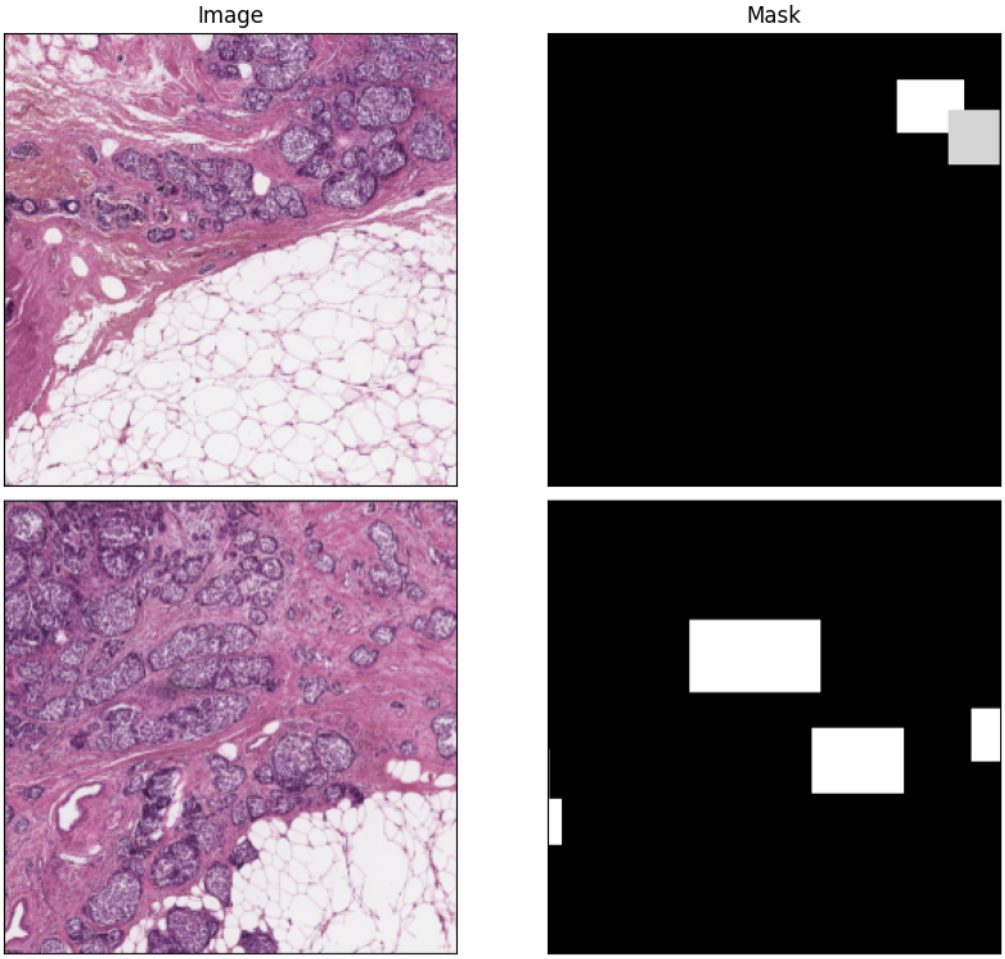

Figure 5.17: BRACS crops example.

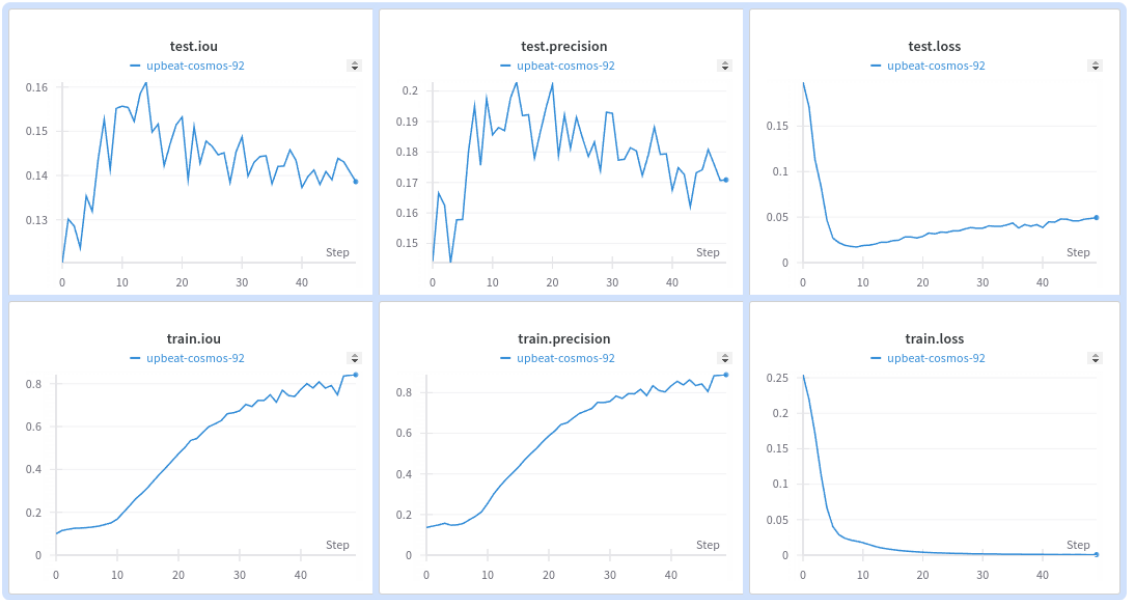

Figure 5.18: BRACS Train vs. Validation metrics with no data augmentation.

As a potential solution, data augmentation was introduced for the training data. To avoid compromising the masks and to prevent small-angle rotations from corrupting the labels, horizontal flip, vertical flip, and rotation of ± 90 degrees were applied. Each of these transformations was applied with a 50% probability, allowing for all, some, or none of them to be applied. These transformations are the same as those applied in the experimentation with data augmentation in the previous dataset. With this data augmentation, an apparent improvement was observed in training, in the sense that the validation metrics did not deteriorate as abruptly.

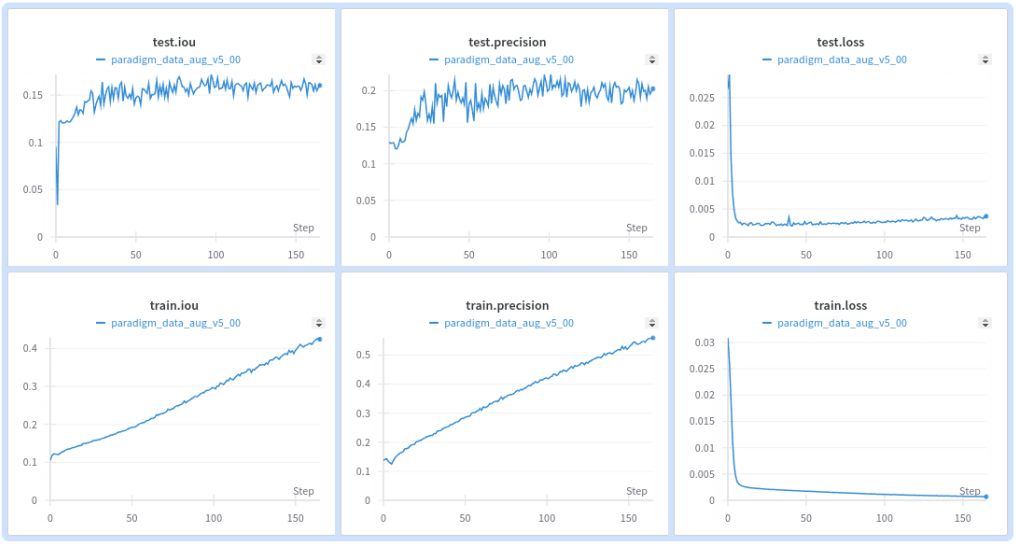

Figure 5.19: BRACS Train vs. Validation metrics with data augmentation.

However, examining the IoU metrics per class, as shown in Figure 5.20, it is evident that the model is not capable of learning. This figure illustrates the low scores on this metric achieved by the different classes, with values jumping between zero, or values very close to zero, and low values in a seemingly random or chaotic manner.

Given these metrics and the aforementioned quality issues of the annotations in this dataset, it does not seem suitable for segmentation tasks. However, a pipeline has been developed to process WSI images with annotations without masks, addressing the challenge of dealing with a large data corpus.

Figure 5.20: BRACS Validation IoU metrics per class

CHAPTER 6
Conclusions

6.1 About the experimentation with Unitopatho

Experimental results with this dataset demonstrate that the approach proposed by Barbano et al [3] yields improved classification of the six considered typologies, indicating that a model composed of specialized sub-models for specific pathologies and various image augmentations can adequately classify colorectal cancer images. Moreover, it has been observed that some of these typologies benefit from image discretization into K levels of grayscale when pathology classification relies on more macro-level structures, whose patterns appear to be more easily recognized by convolutional models in this type of images.

A slight improvement in the classification performance of ViT and ConvNeXt networks compared to convolutional networks such as ResNet or DenseNet has also been observed. Regarding ViT, it would be reasonable to anticipate an even greater improvement with larger datasets than ImageNet-1k, such as ImageNet-21k or JFT-300M, both datasets mentioned in Dosovistky *et al.* [10] paper where this architecture is introduced.

The pretrained models available in common Python Machine Learning packages do not include these weights, and according to the same paper, training on these datasets poses a significant time limitation. Due to this paper, using a typical cloud TPUv3 with 8 cores, it takes around 30 days to do the training with the mentioned datasets.

If indeed models trained on larger datasets were to achieve much better accuracy, the need for a modular model with small specialized models would be eliminated, requiring less image processing to achieve different magnification scales. This would improve inference time and presumably enhance the work of healthcare professionals who could utilize these tools more efficiently and accurately, thereby potentially improving diagnoses and facilitating earlier detection of such pathologies.

6.2 About the experimentation with BCSS

Despite the initial challenges posed by this dataset, a segmentation model has been successfully trained to generate masks that can assist pathologists in diagnosis. The model's success in performing this segmentation is likely due to the high quality of the annotations.

Although the labelled area constitutes only a small fraction of the original and complete Whole Slide Image (WSI), this annotated region is nearly comprehensive, providing detailed information for training the model.

In this experiment, no post-processing of the images was performed to preserve the original masks. To further improve the model's performance, one could consider modifying the masks by consolidating the "Exclude," "Undetermined," and "Other" labels into a single identifier. This approach would enable the calculation of metrics using the remaining classes while instructing the loss function to ignore the consolidated identifier. Such a strategy could reduce noise introduced by ambiguous categories and enhance the model's ability to focus on more clearly defined classes.

Finally, the test conducted with colour-discretized images shows that, for this task and with the applied image resolution, no better results were achieved, nor did it help distinguish any particular class more effectively. However, it is also evident that the model's performance with this type of image is only slightly inferior and still allows the model to learn. Therefore, the use of this type of image could be considered for model prototyping in environments where storage is a limitation. A pipeline could be developed to download data from a large dataset and convert the images to discretized images, which occupy significantly less disk space and serve as a basis for evaluating a model's potential performance.

6.3 About the experimentation with BRACS

Despite efforts to address a segmentation problem using this dataset, neither satisfactory results nor a useful model have been achieved. My conclusion is that much of the issue stems from the data itself. Firstly, the labels do not appear to be thoroughly reviewed, and suffixes such as "sure" cast doubt on the reliability of labels lacking this suffix.

Additionally, the delineated areas lack consistency; sometimes they strictly outline the element to be labelled, while other times they encompass the area with a rectangle. These inconsistent annotations introduce significant noise to the model, complicating its learning process.

However, these labelling issues do not render the dataset entirely unusable for machine learning tasks. Specifically, it might still be useful for classification tasks. The dataset includes global labels for the images that correspond to the directories where they are stored, reflecting seven different typologies. Furthermore, since the labels are organized in ascending order of severity, one could investigate whether the highest value identifier in the masks corresponds with the label indicated by the image name and directory, discarding those that do not match.

By employing this approach, it may be feasible to train a model specifically tailored for classification tasks, thereby maximizing the utility of the dataset and enhancing its effectiveness in supporting various machine learning applications.

Lastly, these tests underscore the critical importance of data quality in training models. Labelling datasets for segmentation tasks is time-consuming and labor-intensive, but it is essential that this labelling is of high quality to effectively support medical professionals, particularly in oncology. As discussed in the chapter on the state of the art, models like SAM can be integrated with labelling tools to streamline the process and reduce the required time.

6.4 Final conclusions

The problem of cancer detection and segmentation through images requires well-labeled and curated data. Such data are not abundantly available publicly, as the publication

of these annotated images involves privacy and security issues due to the sensitivity of medical data.

Moreover, more traditional models should not be dismissed in favour of newer architectures. As observed in the experimentation for classification, these newer architectures do not always outperform more modest architectures. This nuanced perspective underscores the importance of maintaining a diverse toolkit of models, recognizing that the latest innovations in neural network design may not always offer the best solution for every scenario.

Furthermore, no matter how well these models perform, they must always be accompanied by the opinion and judgment of a physician; in other words, these models should act as assistants to the work of a professional pathologist. Otherwise, one would have to consider another type of problems of a more ethical and moral nature.

Finally, tests conducted demonstrate that, at least for the evaluated types, the patterns, and structures can be appreciated at different image resolutions. This suggests that the solution to this issue should involve interdisciplinary work between computer scientists, radiologists, oncologists, and regulatory bodies to advance this field.

CHAPTER 7

Relationship of the Work Developed with the Studies Undertaken

This final degree project is related to subjects that have covered the entirety of the degree program, from the first to the fourth year.

Firstly, and unsurprisingly, introductory computer science and programming courses (IIP), as well as programming (PRG), laid the foundations of this discipline and introduced the principles of Object-Oriented Programming (OOP) that have been applied in code implementation and its structure.

Secondly, Data Structures and Algorithms (EDA) from the second year, along with Algorithmics (ALT) from the fourth year, introduced algorithms and approaches to problem-solving that have always made me wonder whether what I am designing or programming could be done in a better and more optimal way.

This project is more directly related to subjects specific to the chosen branch, Computing. Among these, the subject of Information Storage and Retrieval Systems (SAR) was the first to introduce us, through its practices, to the treatment and processing of large volumes of data. While this subject focused on text data, its principles and teachings are applicable to any data type.

Moreover, various subjects have paved the way to more specialized ones in machine learning. In the first year, the statistics course laid the foundations upon which later courses, such as Intelligent Systems (SIN) in the third year and Perception (PER) in the fourth year, were built.

The subject most directly related to this project is Machine Learning (APR by its acronym in Spanish), where a brief introduction to neural networks and convolutional networks was provided in the laboratory. In this subject, we worked with the Keras library [1], a high-level library that allows for easy implementation of models. While using this library provided us with initial tools for working with these models, when it came time to carry out this project, I opted for *PyTorch* [2], which operates at a somewhat lower level, providing more control over the process. In my opinion, this allows for a deeper understanding of the architecture being constructed and its training process.

Lastly, although these subjects have added significant value, what we are taught represents just the tip of the iceberg of what we may encounter. This is inevitable, as the

[1] https://keras.io/
[2] https://pytorch.org/

university must, within a four-year period, hopefully equip us with enough knowledge to be self-sufficient upon completing the degree. What we see in class and in practices are nothing more than constrained and introductory problems, allowing us, as students, to “get our hands dirty”.

During the completion of this project, I often recalled a phrase uttered by a professor during one of those last-year practices attended by few, where conversations extended beyond the subject: “In the real world, everything is an exception”.

CHAPTER 8
Future Work

This study has focused on exploring the principal architectures underpinning the fields of image classification and segmentation within Deep Learning.

Building on the exploration, experimentation, and insights gained, future work will focus on investigating more complex architectures. One such architecture is HIPT, which was mentioned in the chapter dedicated to state-of-the-art technologies.

Furthermore, this research has enabled a better understanding of the foundational aspects of this field, the construction of models, and the interpretation of articles necessary for implementing these architectures. Consequently, future work should not only aim to enhance existing models but also propose new architectures that can address the limitations and challenges currently faced in this domain.

Another point of study, considering the original article presenting Unitopatho [3] and experimentation with the BCSS and BRACS datasets, is to determine the optimal resolutions at which the necessary features for effectively distinguishing different pathologies are best learned. It is also crucial to investigate whether there are shared patterns within each pathology that could be generalized across other pathologies. This would enable the development of models that are not only accurate but also versatile, potentially serving as valuable tools for assisting doctors and pathologists in the oncology field.

The aforementioned HIPT architecture directly addresses the challenge of working with different resolutions or identifying the optimal resolution by working from the complete Whole Slide Image (WSI) and generating patches at various levels. However, this approach often requires high-performance hardware, which is typically beyond the reach of university students. Therefore, more modest architectures should not be overlooked, as they can provide valuable insights into the problem and help develop better potential solutions.

Regarding the studies and articles consulted for this undergraduate thesis, the role of pathologists and doctors has primarily been limited to labelling datasets, without mention of conducting experiments in collaboration with them. It would be beneficial to work closely with medical professionals and/or delve into more specialized medical literature to extract knowledge that can be applied to the developed models.

Bibliography

[1] S. E. de Oncología Médica, "Las cifras del cáncer en españa 2022," 2022.

[2] B. Korbar, A. M. Olofson, A. P. Miraflor, K. M. Nicka, M. A. Suriawinata, L. Torresani, A. A. Suriawinata, and S. Hassanpour, "Deep-learning for classification of colorectal polyps on whole-slide images," 2017.

[3] C. A. Barbano, D. Perlo, E. Tartaglione, A. Fiandrotti, L. Bertero, P. Cassoni, and M. Grangetto, "Unitopatho, a labeled histopathological dataset for colorectal polyps classification and adenoma dysplasia grading," in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sept. 2021.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, p. 84–90, May 2017.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[11] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, R. G. Krishnan, and F. Mahmood, "Scaling vision transformers to gigapixel images via hierarchical self-supervised learning," 2022.

[12] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," 2021.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[14] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022.

[15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[16] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, Jan. 2024.

[17] H. H. Lee, Y. Gu, T. Zhao, Y. Xu, J. Yang, N. Usuyama, C. Wong, M. Wei, B. A. Landman, Y. Huo, A. Santamaria-Pang, and H. Poon, "Foundation models for biomedical image segmentation: A survey," 2024.

[18] M. Amgad, H. Elfandy, H. Hussein, L. A. Atteya, M. A. T. Elsebaie, L. S. Abo Elnasr, R. A. Sakr, H. S. E. Salem, A. F. Ismail, A. M. Saad, J. Ahmed, M. A. T. Elsebaie, M. Rahman, I. A. Ruhban, N. M. Elgazar, Y. Alagha, M. H. Osman, A. M. Alhusseiny, M. M. Khalaf, A.-A. F. Younes, A. Abdulkarim, D. M. Younes, A. M. Gadallah, A. M. Elkashash, S. Y. Fala, B. M. Zaki, J. Beezley, D. R. Chittajallu, D. Manthey, D. A. Gutman, and L. A. D. Cooper, "Structured crowdsourcing enables convolutional segmentation of histology images," *Bioinformatics*, vol. 35, pp. 3461–3467, 02 2019.

[19] N. Brancati, A. M. Anniciello, P. Pati, D. Riccio, G. Scognamiglio, G. Jaume, G. D. Pietro, M. D. Bonito, A. Foncubierta, G. Botti, M. Gabrani, F. Feroce, and M. Frucci, "Bracs: A dataset for breast carcinoma subtyping in h&e histology images," 2021.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2015.

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[22] N. Marini, M. Atzori, S. Otálora, S. Marchand-Maillet, and H. Müller, "H&E-adversarial network: a convolutional neural network to learn stain-invariant features through hematoxylin & eosin regression," 2022.

APPENDIX A

A.1 Sustainable Development Goals

Degree of the work's relation to the Sustainable Development Goals (SDG).

Sustainable Development Goals	**High**	**Medium**	**Low**	**Does not apply**
SDG 1. **No poverty.**				X
SDG 2. **Zero hunger.**				X
SDG 3. **Good health and well-being.**	X			
SDG 4. **Quality education.**				X
SDG 5. **Gender equality.**				X
SDG 6. **Clean water and sanitation.**				X
SDG 7. **Affordable and clean energy.**				X
SDG 8. **Decent work and economic growth.**				X
SDG 9. **Industry, innovation and infrastructure.**				X
SDG 10. **Reduced inequalities.**		X		
SDG 11. **Sustainable cities and communities.**				X
SDG 12. **Responsible consumption and production.**				X
SDG 13. **Climate action.**				X
SDG 14. **Life below water.**				X
SDG 15. **Life on land.**				X
SDG 16. **Peace, justice and strong institutions.**				X
SDG 17. **Partnerships for the goals.**		X		

Reflection on the relationship of the Final Degree Project (TFG) with the Sustainable Development Goals (SDGs) and with the most related SDGs:

This work is related to three of the Sustainable Development Goals (SDGs), specifically:

- Goal 3: Good Health and Well-being.
- Goal 10: Reduced Inequalities.
- Goal 17: Partnerships for the Goals.

Of these goals, the most intimately related is “Good Health and Well-being” since the lines of research that this work follows align with people’s right to quality healthcare services.

Among the targets of this goal, in correspondence with this work, 3.4 stands out, which will be about the reduction of premature mortality from non-communicable diseases, and target 3.8 on universal health coverage and access to essential health services.

Research in artificial intelligence applied to cancer detection leads to the ability to detect these pathologies more quickly without implying an overload in the daily work of these professionals. On the other hand, these investigations can lead to the reduction of the use of these types of technologies and, therefore, they can be easily implemented in a wide variety of medical centers with less capital allocated to new technologies. This rapid detection would directly influence the survival rate, also influencing another point mentioned in target 3.8, mental health. In addition to the physical health benefits, early detection can also have a positive impact on the mental health of both patients and their relatives. A terminal diagnosis can be emotionally devastating. By reducing the likelihood of such a diagnosis through early detection, we can alleviate some of the psychological stress associated with cancer.

For the relatives of the patient, the early detection of cancer means that they are less likely to face the sudden loss of their loved one. The anticipation of loss can cause significant emotional distress, so reducing this possibility can also contribute to better mental health for the relatives.

Research to make this type of technology more affordable is directly related to target 10.2 of the “Reduced Inequalities” goal, which aims to empower and promote the inclusion of people in different socioeconomic strata, regardless of their condition or other determinants such as age, sex, or race. The democratization of these types of technologies should be able to give access to people in unfavourable situations to better quality health services by being able to assist doctors and health professionals in their daily tasks and duties.

In relation to the "Partnerships for the Goals" objective, this work is specifically related to the targets that mention technology; these are 17.6 which promotes the development, dissemination, and diffusion of technologies to developing countries and, on the other hand, target 17.8 on the creation of a support mechanism for capacity building in science, technology, and innovation.