CentraleSupélec

Master of Science in Artificial Intelligence

MSc THESIS

Cross-Scanner Breast Adenocarcinoma Image Segmentation with Deep Learning Algorithms

Maria Kontaratou

maria.kontaratou@student-cs.fr

Research Center: Archimedes Research Unit, Athena RC

Supervisors

Prof. Maria Vakalopoulou (CentraleSupélec, Archimedes-Athena RC)
Prof. Timos Sellis (Archimedes - Athena RC)
Prof. Myriam Tami (CentraleSupélec)

maria.vakalopoulou@centralesupelec.fr timos@athenarc.gr
myriam.tami@centralesupelec.fr

Athens

November 2025

Abstract

In recent years, deep learning has started to change the way pathologists look at tissue samples and diagnose cancer. It's made analysis faster, more consistent, and more precise than ever before. But one issue keeps coming up - models that work well on one scanner or staining style often struggle when used on another. This thesis focuses on that challenge by developing a scanner-agnostic nuclei segmentation framework, built around the large vision transformer UNI2-h.

Several decoder strategies were tested to find the best way to turn transformer features into detailed segmentation masks. These included classical convolutional designs such as DeepLabV3, multi-scale approaches like PixelFPN + ASPP and PixelFPN + 1×1 head, as well as a Lite Query decoder inspired by transformer attention. A Dual-Resolution Fusion + DeepLabV3 model was also introduced, combining high- and low-resolution feature paths to preserve fine details while maintaining global context. All models were trained on the multi-scanner COSAS dataset (180 training images, each approximately 1500×1500 px). The proposed Dual-Fusion model reached a final score of 0.8217, followed closely by PixelFPN + ASPP and DeepLabV3 baselines.

Overall, this work shows that transformer-based foundation models, when paired with carefully designed decoders, can achieve strong, domain-robust segmentation in histopathology - even with limited data. The results suggest real potential for clinical use and pave the way for future studies on whole-slide image (WSI) learning, domain adaptation, and hybrid attention-convolution architectures to further improve consistency across different scanners.

Contents

Chapter 1

Introduction

1.1 Context and Motivation

The medical world has been heavily impacted and influenced by AI as well as medical imaging. When it comes to professionals and clinical tasks, AI has arguably made their jobs easier for tasks such as disease detection and personalized treatments.

Medical imaging in particular has been of utmost importance, as technologies such as CTs, X-rays and MRIs offer very specific information on tissue structures and organ details. Through Deep Learning, the information is then analyzed and creates a more accurate diagnosis, even better at times than the one made by a clinically trained pathologist [1].

In digital pathology, a huge step has been taken: scientists used to need microscopes to examine tissue slides stained with hematoxylin and eosin (H&E). After the introduction of whole-slide imaging (WSIs), these slides are scanned at high resolution and analyzed digitally. This shift has introduced room for AI models to automatically segment, classify, and highlight regions of interest more accurately and avoiding human error [2].

In cancer care, the role of these technologies becomes even more critical. Breast cancer is still the most common cancer among women worldwide and one of the leading causes of death [3]. Early detection and accurate diagnosis are the key to improving survival rates, and histopathology remains the gold standard in this process. With the help of AI-based segmentation methods, tumor regions can be identified more consistently, reducing the workload for pathologists while also minimizing the risk of human error [4].

1.2 Research Problem

One of the main challenges when developing AI models is making sure that they can perform well not only on the data on which they were trained but also on new unseen data. In theory, most learning algorithms assume that both training and testing data come from the same distribution, known as the independent and identically distributed (i.i.d.) assumption. However, this rarely

holds true in practice.

In the real world, data often come from different sources, collected under slightly different conditions, which can cause a problem known as domain shift. Domain shift occurs when a model encounters data that do not match the characteristics of its training data, often leading to a noticeable drop in performance. These invisible changes are referred to as out-of-distribution (OOD) data and they have the potential to seriously impair deep learning algorithms dependability [5].

In digital pathology, this issue is especially common. The way images appear can be changed by even minor differences between scanners such as variations in color resolution, contrast or illumination. Although these changes may seem minor to the human eye, for AI models that rely on pixel-level consistency, they can have a major effect on performance [6, 7].

In mathematical terms, let $\mathcal{X}$ represent the input images and $\mathcal{Y}$ represent the corresponding labels(like "tumor" or "no tumor"). A model learns a function

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

based on data drawn from one or more source domains $p_s(x, y)$(training data). When the same model is tested on a new target domain $p_t(x, y)$(new data), even small differences between p_s and p_t can make it to can cause it to misread features or perform in a poor way [5].

This problem is especially relevant in actual clinical settings. The scanners used in hospitals and labs come from various manufacturers and the color and texture of the images they generate differ slightly. A segmentation model trained on images from one scanner may perform worse when tested on images from another, making it unreliable for diagnostic use. To address this problem, researchers have focused on domain generalization (DG), developing models that can handle data from different scanners or domains without needing access to those examples during training. The objective is to learn representations that hold up well even when staining techniques or different scanners cause changes in the appearance of the image [8, 5].

Task 2 of the Cross-Organ and Cross-Scanner Adenocarcinoma Segmentation (COSAS) 2024 Challenge focuses on this issue [9]. In this task models are assessed on histopathology images that have never been seen before and trained on images from multiple scanners. Each scanner presents different color statistics and trends, making it a realistic dataset to test how well segmentation models adapt to new conditions. The objective is to create algorithms independent of the scanner that can precisely segment adenocarcinoma regions [7].

Chapter 2

State of the Art

2.1 Essential Concepts

2.1.1 Convolutional Neural Networks (CNNs)

The origins of Convolutional Neural Networks date back to the late 1980s, when Yann LeCun and his colleagues introduced LeNet-5, one of the first convolutional networks trained to recognize handwritten digits [10]. While the idea was ahead of its time, computational power and available data were still very limited, which slowed down progress. It wasn't until 2012 that CNNs gained global recognition when AlexNet, a much deeper and larger architecture, won the ImageNet Large-Scale Visual Recognition Challenge by a wide margin [11].

The main idea behind CNNs lies in the convolution operation. This operation uses small filters, also called kernels, that slide over the image and detect specific patterns. At first, the network focuses on simple details such as edges or color intensity. As it goes deeper, it begins to recognize more abstract and complex structures, like the arrangement of cells or tissue patterns in medical images. Each filter specializes in recognizing a particular feature, and the model automatically learns these filters through training, it doesn't need to be told what to look for.

A typical CNN is built using several key components. The convolutional layers handle feature extraction, applying multiple filters to the input image. After each convolution, activation functions such as ReLU introduce non-linearity, allowing the model to capture complex visual relationships. Pooling layers then reduce the spatial resolution of the data, keeping only the most relevant information and making the model faster and more robust to small changes in the image. Finally, fully connected layers or decoder blocks combine all the learned features and produce the output, such as a classification or a segmentation mask.

One of the main reasons CNNs became so popular is because they can learn directly from raw image data. Before deep learning, most computer vision methods used hand-crafted features, meaning manually designed image descriptors that required experts and at times failed to generalize. CNNs completely changed this. By learning hierarchical feature representations automatically, they became truly effective in complex domains like medical imaging, where

visual details can be subtle and vary across patients [12].

In digital pathology, CNNs have been used to identify cell nuclei, segment tumor regions, and classify different tissue types [13]. CNNs are great at picking up small details in images, which makes them useful on spotting small cancerous areas. But as they mostly focus mostly on local detals, they miss the broader spatial context. They can also be thrown off by changes in things like color or texture, which often happen when images come from different scanners or use slightly different staining methods [7].

2.1.2 Vision Transformers (ViTs)

In 2020, Vision Transformers (ViTs) [14] were introduced, marking a big shift in how visual information is processed. Instead of relying on convolutions, ViTs use a self-attention mechanism that can capture relationships across an entire image, not just within small regions. The idea came from the Transformer architecture, originally built for language tasks [15]. In this case, it was adapted for images, thus allowing models to understand global structures and context, not just local details like traditional CNNs.

In a Vision Transformer, the input image is divided into fixed-size patches, typically 16×16 pixels, which are flattened and linearly projected into an embedding space of constant dimension. Each of the visual tokens created by these patch embeddings represents a distinct area of the image. To ensure that the model retains awareness of spatial organization, positional embeddings are added to each token in order to preserve the spatial links lost during patch flattening. Additionally, a learnable class token (CLS) is presented, which is intended to compile data from the full image while self-attention is being used.

At its core, a Vision Transformer (ViT) is built from encoder blocks that each have two main parts: multi-head self-attention and a feed-forward network (FFN). The self-attention part lets every patch of the image "look" at every other patch and figure out which ones are important to focus on. This helps the model understand how different regions of an image connect; both the close-up details and the bigger picture. After that, the feed-forward layer takes what the attention mechanism has learnt and turns it into richer, more meaningful features. To keep training smooth and stable, layer normalization and residual connections are used.

ViTs learn spatial relationships directly from the data, unlike CNNs, which rely on built-in assumptions about things like position and movement in an image. This makes ViTs more flexible and better at generalizing to new conditions, but it also means they need more data and computing power to train. This feature becomes especially useful in digital pathology, where CNN performance is sometimes limited by domain variability resulting from variations in scanners, staining, or illumination. Thanks to their global attention mechanism, ViTs are less dependent on precise pixel-level details and are therefore more resilient to these kinds of changes. [5].

The Vision Transformer architecture has recently been expanded to include self-supervised

learning and extensive pretraining. ViTs have been able to develop rich visual representations from huge and diverse datasets using techniques including contrastive learning and masked image modelling, which has improved their robustness to changes in color, texture, and acquisition settings [16, 17, 18]. These advances have led to the rise of foundation models - large pretrained networks that can be adapted to many different tasks, including digital pathology. Their ability to generalize across domains makes them especially useful for addressing the problem of domain shift, which is discussed in the following sections.

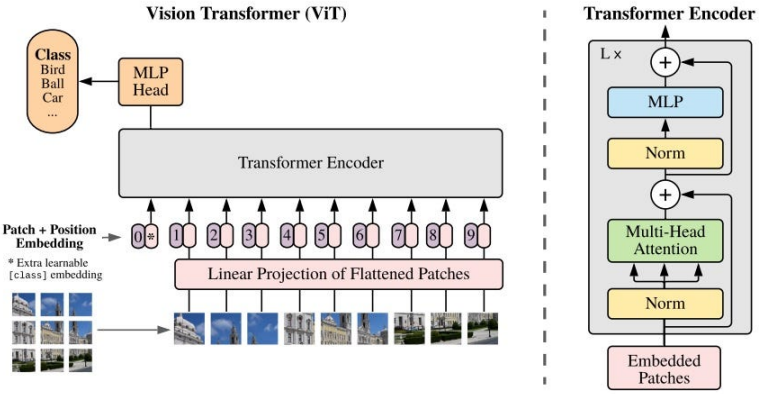

Figure 2.1: An overview of the ViT architecture

2.2 Related Work

2.2.1 Domain Generalization in Digital Pathology

A central challenge in computational pathology is ensuring that models trained on one dataset continue to perform well when applied to new data from different conditions. This issue, known as domain shift, arises from variations in image acquisition, such as differences in scanner type, magnification, or staining intensity [7, 6]. While these variations may appear subtle to the human eye, they can significantly change the low-level pixel distributions that deep learning models depend on, often leading to reduced performance when the model is exposed to unseen data.

Most segmentation models are trained under the assumption that the training and testing data come from the same distribution. However, this assumption is seldom true in digital pathology. Each scanner manufacturer uses its own color correction, lighting, and compression methods, which can change the overall appearance and statistical properties of the final whole-slide images. As a result, models that perform well on validation sets struggle to generalize to new clinical settings. Recent studies have repeatedly pointed this out, showing that achieving real robustness in histopathological image analysis requires more than just basic data augmentation or normalization techniques. [5, 8].

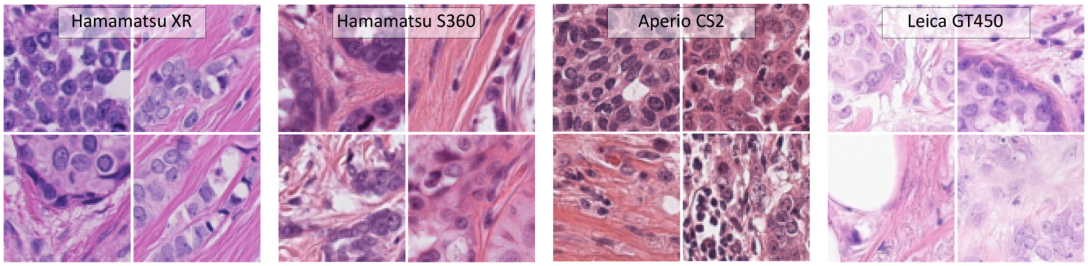

Figure 2.2: Examples of H&E image variations induced by different scanners. Even for the same tissue type, scanner-specific color and contrast differences cause noticeable domain shifts [19].

Domain generalization (DG) is an area of research focused on building models that can handle unseen domains without having access to their data during training. The main goal is to make the model less sensitive to appearance changes (like color or texture) and instead help it learn features that stay consistent across different domains. Mathematically, this can be seen as learning a function $f : X \rightarrow Y$ that performs reliably across multiple source domain distributions $p_s(x, y)$ and still works well on a new targeted domain $p_t(x, y)$, where $p_s(x) \neq p_t(x)$.

Several strategies have been investigated in order to accomplish this. Color augmentation, stain normalization, and style transfer are examples of data-centric methods that try to mimic the diversity caused by different scanners and staining processes [20, 21, 22]. These methods help the model stop relying so much on color differences by showing it a wider mix of examples during training. In contrast, representation learning focuses on improving the model itself so it can pick up features that stay stable across different domains. For example, domain-adversarial training helps the model learn representations that look similar no matter which scanner the data comes from. Meanwhile, attention-based networks and multi-scale encoders teach it to focus more on the actual structure and shape of the tissue instead of just color or texture [8].

In digital pathology, the focus has shifted from task-specific convolutional networks to more flexible self-supervised and transformer-based architectures. These newer models can learn broad visual representations from large and diverse histopathology datasets. Unlike traditional supervised methods, they use learning objectives such as contrastive learning and masked image modelling, which help them capture meaningful tissue structures and color patterns that stay consistent even when staining or scanner conditions change [16, 17].

Frameworks like DINO, MAE, and DINOv3 have shown that large-scale pretraining can produce strong, transferable representations that often work well on new data with little or no fine-tuning [18]. Still, domain generalization in pathology remains a real challenge. Many existing models rely heavily on data augmentations or multiple source domains during training, which limits their reliability when faced with entirely new clinical data. To overcome this, newer foundation models (such as UNI2-h [23]) have been developed to learn domain-robust features that generalize naturally across different scanners, organs, and tissue types [23].

2.2.2 Pathology Foundation Models

Task-specific networks have given way to more general, data-driven architectures that can directly learn from vast and varied picture collections as digital pathology has developed. The goal of foundation models is to train a single, general-purpose visual encoder that can be modified to many downstream applications with little fine-tuning, rather than developing a new model for each dataset or scanner.

Building a foundation model starts by dividing whole-slide images (WSIs) into patches that are processed by a neural network and then pretrained on a vision encoder (usually ViT) on unlabelled data. Then the model uses self-supervised learning to learn general-purpose representations [16, 17] and once it's pretrained, the encoder can be fine-tuned for tasks such as classification or segmentation with limited labelled data, ensuring the reduction of cost annotation and leading to stronger generalization across scanners and institutions.

Several large-scale pathology foundation models have been introduced in recent years, each taking a slightly different approach to the same goal:

UNI

The Universal Pathology Foundation Model (UNI) [23] was one of the earliest models created by Mahmood and his team. Over 100 million image tiles from over 100,000 H&E-stained whole-slide images covering 20 major tissue types were used to train it. The model was able to acquire universal visual characteristics that capture both broad morphological patterns and fine-grained tissue details thanks to this quantity of data. UNI was fully trained inside the pathology domain using a self-supervised technique, in contrast to previous models that depended on transfer learning from real pictures. In addition to reaching cutting-edge results on 34 computational pathology benchmarks, UNI unveiled new features including resolution-agnostic tissue categorization, few-shot learning for slide-level diagnosis, and cross-cancer subtyping across 108 tumor types in the OncoTree system. We will be further analyzing the model in the next chapter.

Virchow2

Virchow2 [24] is a large vision transformer model, trained through self-supervision that leverages mixed magnification information. Instead of relying on a single resolution, the model simultaneously processes tiles from multiple magnification levels (for example, 5×, 10×, and 20×), allowing it to learn both fine-grained cellular structures and broader tissue organization. A student-teacher architecture based on the principles of self-distillation is given picture tiles of the same tissue location at varying magnifications during pretraining. While the student network learns to align its representations across scales, the teacher network, which is updated using an exponential moving average, offers stable feature targets. In pathology, where diagnostic signals

emerge at several spatial levels, this technique helps the model detect the same tissue pattern even when examined at various resolutions.

For the model to be more robust, its creators have applied domain normalization and heavy color augmentations that mimic different staining, lighting and scanner conditions. The self-attention part of the backbone helps capture relationships with distant regions of an image, while the magnification-aware embeddings make tissue patterns interpretable, even at different zoom levels. The model was trained on a dataset of over 3.1 million whole-slide images, spanning multiple organs, institutions, and staining techniques. After pretraining, the encoder can be fine-tuned with minimal effort for downstream applications such as whole-slide analysis or tile-level classification.

Midnight

A different approach to developing pathology foundation models is taken by Midnight [25], which prioritizes training efficiency over extensive data collection. Midnight uses significantly less data to achieve comparable results to most foundation models, like UNI and Virchow2, which are trained on millions of whole-slide images. A new regularization technique that encourages the model to learn a variety of stable image features and enhanced color augmentations in the hematoxylin-eosin-dab (HED) color space are two of the pathology-specific modifications made to the model, which is based on the DINOv2 self-supervised framework. By processing tiles from various magnifications during training, the model is able to learn both larger tissue patterns and minute cellular details. Even with fewer examples thanks to this design, Midnight is able to capture pertinent visual information.

What makes Midnight especially notable is its data efficiency. Only 12,000 WSIs from the TCGA dataset were used to train the smallest version, Midnight-12k, which produced results on par with or better than models trained on hundreds of thousands of slides. The TCGA and Netherlands Cancer Institute combined datasets were used to train larger versions, like Midnight-92k and Midnight-92k/392. The latter was refined on higher-resolution images to better capture morphological details. Performance was enhanced by this fine-tuning step, especially in segmentation tasks in well-known datasets like MoNuSAC and CoNSeP.

2.2.3 DeepLabv3

DeepLabV3 [26] is one of the most popular architectures for image segmentation and has been widely used in both natural and medical imaging [27]. It was developed by Google as part of the DeepLab family models. DeepLabV3 introduced Atrous Spatial Pyramid Pooling (ASPP), a method to capture image details at different scales, improving the production of segmentation results.

The main idea behind DeepLabV3 is to let the model see both small details and larger context at the same time. The ASPP module does this by applying several filters with different spacing

(called dilation rates). This helps the model recognize objects of different sizes, which is a very useful feature for pathology images, where cells and tissues can vary a lot in shape and scale.

DeepLabV3 also adds a global pooling layer, giving the model a view of the total image while keeping track of local information. This balance helps it produce smoother and more accurate segmentation maps, making it one of the most stable and reliable decoder designs available.

In digital pathology, DeepLabV3 is often used as a benchmark decoder, especially when testing transformer-based encoders like BEiT [28], It's efficient, easy to train, and doesn't need huge amounts of computational power, which makes it ideal for large whole-slide images.

Still, DeepLabV3 has some limitations. Because it's based on convolutions, it sometimes struggles to capture long-range relationships or very fine boundaries when the data changes between scanners or staining styles. However, when combined with strong transformer encoders or domain-robust backbones like a pathology foundation model, it performs consistently well across different datasets and imaging conditions.

2.3 Critical Discussion

As we have examined in this chapter, deep learning methods and foundation models have pushed computational pathology forward, but this comes with issues when it comes to the implementation in real-life settings. UNI, Virchow2 and Midnight indeed work powerfully, but their strength depends on factors such as huge datasets, high-end hardware and GPUs as well as thorough domain curation.

One of the biggest challenges is still data. Hospitals and labs do produce thousands of whole-slide images, but very few have detailed annotations, as pathologists rarely have the time to mark regions by hand. So even though models can be pre-trained on large datasets with no labels, fine-tuning - especially for WSIs - remains an issue, particularly for pixel-level tumor borders or segmentation tasks in general.

Another major issue is scanner variability. Different hospitals own different scanners, thus slides tend to look surprisingly different even if they come from the same tissue type. Differences that occur in brightness, color balance and texture do not help the model generalize well across all scanners. As a result, a model that performs well on one dataset can drop significantly if tested on slides from another lab, making domain shift a huge barrier to clinical deployment.

Then there's the matter of image size. Whole-slide images can be as massive as several gigapixels, so to make them manageable, we split them into tiles at the cost of removing spatial context. Transformers do capture global structures but are heavy on the memory, while CNNs are lighter but lose fine details. A good trade-off still remains an open issue, especially for segmentation tasks.

We also have to think about interpretability. Pathologists need to understand why a model made a certain decision, not just what that decision was. However, foundation models, and especially transformers, often act like black boxes. Attention maps and saliency tools help a

bit, but they rarely give clear, human-understandable explanations of what the model is seeing. Without that, clinical trust is still limited.

Lastly, there's the question of efficiency. Many of the top-performing models in the literature are enormous; trained on millions of images using clusters of GPUs for weeks. That's great for research, but not realistic for most labs or hospitals. It also makes experimentation much slower. What's needed are models that can perform well without depending on extreme scale, models that are not only accurate but practical.

Chapter 3

Research Center Presentation

3.1 Host Organization

This thesis was conducted as part of a research internship at Archimedes, a unit of the Athena Research Center in Athens, Greece [29]. Archimedes is dedicated to advancing artificial intelligence research and its applications across science, technology, and medicine.

The center hosts teams of researchers and engineers working in computer vision, biomedical imaging, robotics, and machine learning. The ecosystem is highly collaborative, combining expertise in mathematics, physics, biology, computer science, and medical research.

The internship took place under the supervision of Dr. Maria Vakalopoulou at the MICS Laboratory, CentraleSupélec [30], who is also lead researcher in Archimedes.

Daphne Tsolissou and Andreas Lolos, both PhD students affiliated with the group, are actively contributing to the field of computational pathology and medical AI. They both provided incredible help and guidance throughout my internship. Their recent works on multimodal carotid risk assessment using vision–language models [31] and sparse Gaussian process-based multiple instance learning [32], show great commitment to advancing interpretable and data-efficient machine learning in medicine.

3.2 Ongoing Research and Thesis Context

At Archimedes, people from very different backgrounds - computer scientists, engineers, biologists, and mathematicians - work together to push artificial intelligence closer to real healthcare. The team's research is mainly centred around computational pathology, foundation models for medical imaging, and domain generalization. Another key direction is clinical collaboration, where the group works closely with specialists to ensure their methods are not only accurate but also interpretable and practical for doctors to use.

My thesis fits naturally within this research effort, as the goal of the project was to explore how deep learning models can remain accurate when dealing with histopathology images captured by

different scanners.

To address the domain shift problem, the work included:

- Analyzing the COSAS dataset, understanding how each scanner's images differ in terms of color, brightness, and tissue appearance
- Evaluating performance on both familiar and unseen scanner domains
- Visualizing and interpreting the results, summarizing key findings that contribute to the lab's ongoing research on robust and foundation-based pathology models

Naturally, the project contributes to Archimedes' goal of developing AI systems that can work reliably across hospitals and devices. Personally, it was an opportunity to combine my technical background with biomedical research and to experience how AI can directly support medical diagnosis and research.

Chapter 4

Problem Description

4.1 Formal Problem Statement

The goal of this work is to develop a deep learning model that can accurately segment adenocarcinoma regions in hematoxylin and eosin (H&E)-stained breast histopathology images. More importantly, the model must generalize well across across microscope scanners that differ in color, texture, and image quality.

To describe this, let:

- $\mathcal{X} \subset \mathbb{R}^{H \times W \times 3}$ be the space of RGB histopathology image patches,
- $\mathcal{Y} = \{0, 1\}^{H \times W}$ be the space of corresponding binary masks, where 1 denotes adenocarcinoma tissue and 0 denotes non-tumorous regions

Each imaging domain D_k (for example, a specific scanner model) follows a joint probability distribution $P_k(X, Y)$, and we assume access to n source domains:

$$\mathcal{S} = \bigcup_{k=1}^{n} \left\{ (x_i^{(k)}, y_i^{(k)}) \right\}_{i=1}^{m_k} \sim P_k,$$

but no access to the unseen target domain D_T.

The task of domain generalization for semantic segmentation is therefore to learn a function $f_\theta : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected segmentation loss on the unseen domain:

$$\theta^* = \arg\min_\theta \ \mathbb{E}_{(X,Y)\sim P_T} \left[\mathcal{L}(f_\theta(X), Y) \right].$$

In practical terms, f_θ is implemented as a deep neural network built around a pretrained Vision Transformer (ViT-H/14) backbone, extended with a segmentation decoder. The network predicts tumor probabilities for each pixel, which are subsequently thresholded to form the final binary mask.

The difficulty of the problem is upscaled as the dataset is quite small, with 180 labelled images available for training and validation. Each of the six scanners has its own visual style

with different domain shifts. Strong color imbalance also arises as an issue beacause tumor regions are usually a small fraction of each patch. An image size of 1500×1500 pixels also pushes the limits of GPU memory.

Our contribution is to build a model that holds segmentation accuracy with the aforementioned difficulties when applied to new unseen images from scanners without additional fine-tuning.

4.2 Hypotheses

H_1 - Domain shift mainly affects appearance, not meaning

We assume that the tissue structure and relationship between $\mathcal{X}$ and $\mathcal{Y}$ remain the same across scanners, even though the appearance seems different. In other words, the problem can be treated as a covariate shift, where $P(X)$ varies but $P(Y \mid X)$ remains constant. This aligns with the covariate shift assumption used in domain generalization papers [33, 5]. If the model learns features based on morphology rather than color or scanner-specific texture, it generalizes well across domains.

H_2 - Foundation model features are transferable to pathology

Despite being trained on large datasets, foundation models like UNI2-h [23] acquire important low and mid-level features like texture and structure. After fine-tuning with specific data augmentations, representations can be morphed into pathological tasks.

H_3 - Strong augmentation and balanced sampling help overcome data scarcity

Since there are only 180 images in the training data, we expect that data augmentation and methods like color jittering, rotations, and domain-balanced batches increase sample variety and decrease overfitting.

4.3 Data

4.3.1 Dataset Overview and Source

The dataset used in this thesis is from the Cross-Organ and Cross-Scanner Adenocarcinoma Segmentation (COSAS 2024) challenge [9], hosted on the Grand-Challenge platform. Specifically, the work focuses on Task 2 - Cross-Scanner Adenocarcinoma Segmentation.

All images were collected from the histopathology archive of Ruijin Hospital, Shanghai Jiao Tong University School of Medicine, and depict invasive breast adenocarcinoma tissue stained with hematoxylin and eosin (H&E) at 20× magnification. COSAS was specifically designed to model real-world clinical variation, combining data from several scanning systems that differ in optics, illumination, and color calibration.

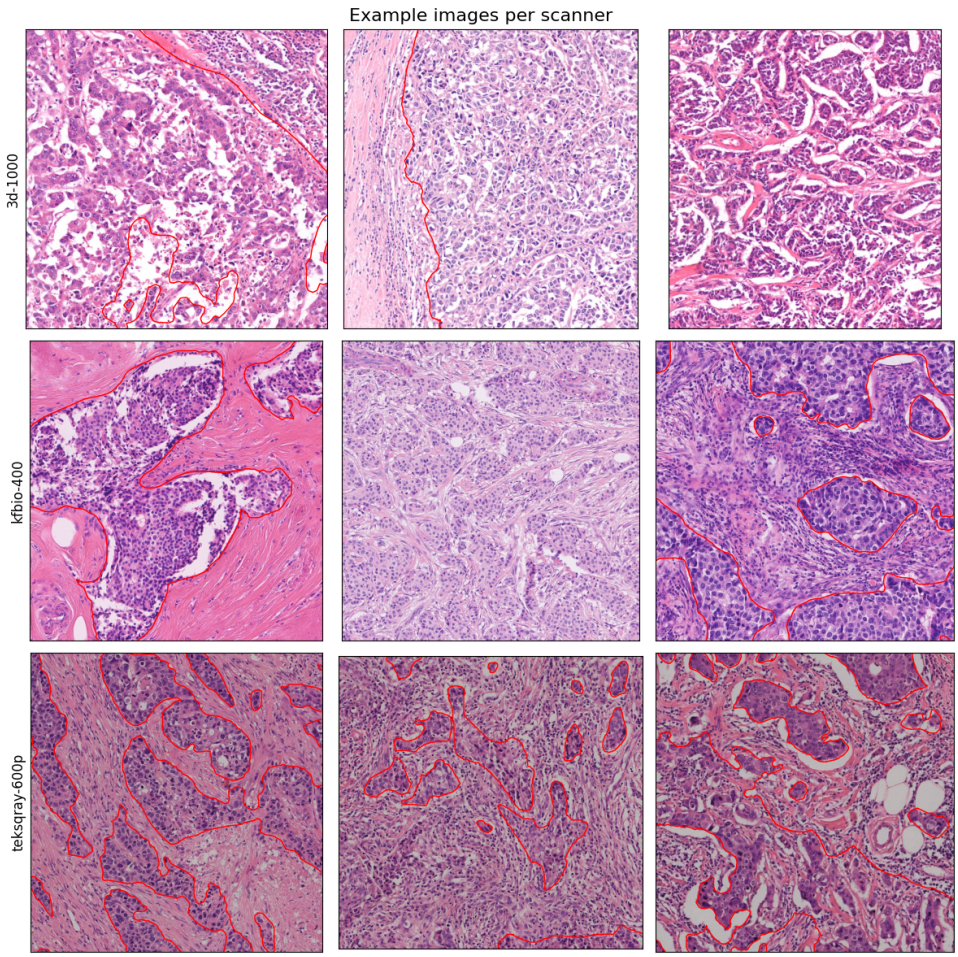

Figure 4.1: Representative image patches from the COSAS training set. Each row corresponds to a different scanner domain: 3d-1000, kfbio-400, and teksqray-600p. The red contours indicate the tumor boundaries.

4.3.2 Dataset Composition

The dataset is divided into a labelled training set and a hidden test set managed by the challenge organizers, who uploaded the test set shortly after the end of the challenge. Each image tile measures roughly 1500 × 1500 pixels and is paired with a binary segmentation mask manually annotated and verified by expert pathologists. Each pixel in the segmentation masks represents either 0 (normal tissue) or 1 (adenocarcinoma). There were boundary inconsistencies - which were expected from manual histopathology annotations - but the dataset remains trainable.

Table 4.1: Composition of the COSAS 2024 dataset

Split	# Patches	# Scanners	Scanners	Description
Training	180	3	teksqray-600p, kfbio-400, 3d-1000	Labeled set used for model development
Test (hidden)	90	6	seen + leica-450, motic, 3d-250	Used for blind evaluation of cross-scanner generalization

4.3.3 Data Characteristics

Each sample is a 2D RGB histopathology image stored as an 8-bit PNG file. Images are large (approximately 1500×1500 pixels) and represent 20× optical magnification. All patches have a nearly square aspect ratio and uniform spatial resolution.

The most important aspect that affects model performance is the scanner color and contrast variation, which differ in hues and brightness. For example, some scanners may capture warmer tones while others more bluish hues. Despite these differences, tissue and cellular morphology remain the same, which underpins the assumption of Hypotheses 1.

4.3.4 Preprocessing and Augmentation

Before the training loop, both images and masks were standardized and augmented. The goal was to expose the model to realistic variations with differences in color, illumination and slight deformation. We resized and padded all images to 1512×1512 pixels, as the original dataset contained images ranging from about 1300 pixels to 1600 pixels (1500 pixels on average), and 1512 was chosen because it is divisible by 14, ensuring compatibility with the encoder's input resolution. Reflective padding was applied to the image borders to avoid artificial edges, while masks were padded with zeros to preserve clean background boundaries.

Transformations were implemented using the Albumentations library:

1. Geometric transformations: Random flips, rotations, and scaling were applied. Mild elastic transformations were also added to imitate the small distortions introduced during tissue slide preparation
2. Color and brightness changes: Random variations in brightness, contrast, hue, and saturation
3. Noise and compression artifacts: Gaussian noise, sharpening, and JPEG compression to simulate the imperfect conditions found in digital scanner
4. Cropping and padding: Tumor regions were prioritized during cropping, ensuring that most tiles contained useful features for learning, while reflective padding ensured a consistent image size
5. Normalization: Finally, images were normalized using ImageNet mean and standard deviation values and converted into tensors for model input

A simpler validation pipeline was used, applying only resizing, padding, and normalization. This ensured that evaluation was consistent and unaffected by random augmentations.

Table 4.2: Data augmentation transformations

Transformation	Probability	Hyperparameters / Notes
Horizontal Flip	0.5	Random left-right reflection
Vertical Flip	0.5	Random up-down reflection
Random Rotate 90°	0.5	Random rotation in 90° increments
Shift-Scale-Rotate	0.5	shift_limit=0.03, scale_limit=0.08, rotate_limit=12°
Elastic Transform	0.08	α=10, σ=4, α_{affine}=4
Longest Max Size	1.0	Resize to 1512×1512 px (preserve aspect ratio)
Pad If Needed	1.0	Reflective padding to reach 1512×1512 px
Crop Non-Empty Mask	0.7	Prioritize tumor-containing regions
Random Crop	0.3	Uniform random crop to 1512×1512 px
Brightness / Contrast	0.6	limit=±0.12 for both
Hue / Saturation / Value	0.4	hue=±6°, sat=±14%, val=±10%
RGB Shift	0.2	±6 per channel
Gaussian Noise	0.1	var_limit=(3.0, 10.0)
Sharpen / Unsharp Mask	0.1	alpha=(0.08, 0.15), blur_limit=(3, 5)
Normalize	1.0	mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225)

4.4 Dataset Exploration

4.4.1 Training Set Analysis

The training set contains 180 labelled patches, evenly distributed across three scanners: 3DHISTECH PANNORAMIC 1000 (3d-1000), KFBIO KF-PRO-400 (kfbio-400), and TEKSQRAY SQS600P (teksqray-600p).

Across the dataset, the tumor vs non-tumor regions differ substantially; from patches with almost no cancer-zone tissue to ones entirely filled with tumor regions. On average, about half of each image is annotated as tumor, but variation between scanners exists. This introduces class imbalance, which can bias model training but also encourages the model to learn from diverse tissue contexts.

Color statistics revealed clear scanner-dependent trends:

- 3d-1000 and kfbio-400 images appear brighter and slightly warmer
- teksqray-600p tends to produce darker, cooler tones

4.4.2 Test Set Analysis

The test set contains 90 patches from six scanners: the three seen during training plus three unseen ones: 3DHISTECH PANNORAMIC 250 (3d-250), Leica-450, and Motic. These additional devices introduce new color and brightness profiles. Dataset exploration showed that 3d-250 and 3d-1000 samples had the largest proportion of tumor tissue, while Motic and teksqray-600p samples had smaller tumor areas.

Average RGB intensity analysis revealed distinct color trends:

- 3d-250 and 3d-1000: bright, warm tones
- leica-450: pale, slightly desaturated hues
- motic: darker, cooler colors
- teksqray-600p: darker overall, with high contrast

Table 4.3: Summary of tumor coverage and visual characteristics across scanners in the COSAS dataset

Scanner	**Mean Tumor Ratio**	**Std. Dev.**	**Observation**
3d-1000	0.50	0.26	Balanced tumor coverage
3d-250	0.58	0.29	Unseen domain with higher tumor density
kfbio-400	0.35	0.25	Moderate variation in tumor presence
leica-450	0.40	0.18	Pale staining and uniform tissue structure
motic	0.33	0.21	Cooler tone and darker appearance
teksqray-600p	0.34	0.16	Darker tone, consistent with training distribution

Chapter 5

Methodology

5.1 Model architecture

5.1.1 Foundation Encoder: UNI2-h

The encoder we used is UNI2-h [23], a Vision Transformer (ViT-H/14) created by MahmoodLab, as part of the UNI foundation models. UNI2-h uses eight register tokens, enabling global feature propagation without relying on a single classification token (CLS). Each input is divided into non-overlapping 14×14 patches, which are linearly projected into 1536-dimensional embeddings.

The embeddings then go through 24 transformer blocks, each containing multi-head self-attention with 24 heads and a SwiGLU-packed MLP. The SwiGLU activation (Swish-Gated Linear Unit) [34] improves how neural networks learn by using a gating mechanism. It first creates two parallel transformations of the input: one determines how much information should pass through (the "gate"), and the other carries the main signal. The two are then combined through multiplication to a feature embedding. (Figure 5.1)

The images were resized to 1512 × 1512 pixels to fit UNI2-H's patch-based design, where the encoder breaks the image into 14 × 14 patches. At this resolution, the model forms a 108 × 108 grid of tokens (1512 ÷ 14 = 108). This setup helps keep the same patch-to-embedding ratio used during pre-training, while still capturing fine details in the tissue structure.

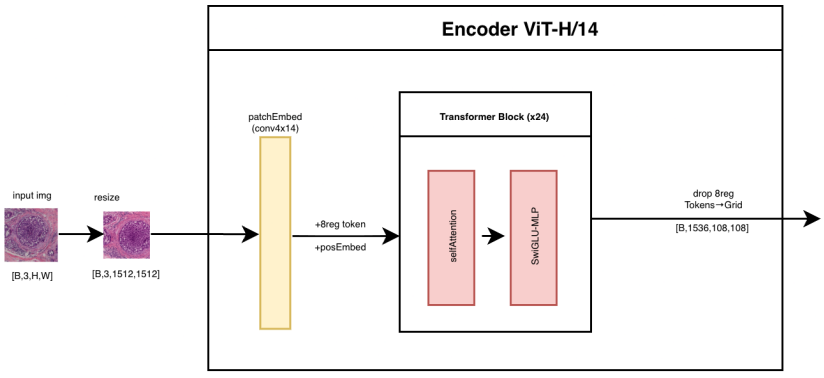

Figure 5.1: The UNI2-h encoder

5.1.2 DeepLabV3-style ASPP Decoder

For the segmentation head, we used a DeepLabV3-style Atrous Spatial Pyramid Pooling (ASPP) decoder, which connects directly to the UNI2-h encoder output (Figure 5.3). This decoder follows the classic DeepLabV3 design but has been simplified into a single-path configuration, focusing on keeping strong contextual awareness.

The Atrous Spatial Pyramid Pooling (ASPP) helps the model capture multi-scale information by applying many convolution layers with different dilation rates, so the model can see both small details (nuclei edges) and general structure (tissue structure). In the implementation, the ASPP block has five parallel operations: a 1×1 convolution, three 3×3 convolutions with increasing dilation rates (12, 24, and 36), and a pooling branch. The outputs of these branches are concatenated through a 1×1 projection layer through group normalization, GELU activation and a small dropout for regularizaton. The final features are processed by a main head, which produces the final segmentation logits and is then upsampled to the input image size (1512×1512 pixels), followed by sigmoid activation and thresholding to generate the mask.

The decoder was chosen as a strong and interpretable baseline for this specific adenocarcinoma segmentation task which integrates information at different levels without losing spatial resolution. Cancerous tissues show glandular boundaries and stromal texture that vary widely across scanner sources, and studies have shown that context modelling through ASPP improves segmentation under stain and scanner variation. Wang and Liu [27] applied DeepLabV3+ to gastric cancer pathology slides and showed that multi-dilation feature aggregation improved boundary precision and robustness across scanners. Similarly, the TransDeepLab framework [35] replaced CNN backbones with pure Transformers and demonstrated that the ASPP concept remains effective in capturing contextual structure across multiple organs in the Synapse Multi-Organ CT dataset.

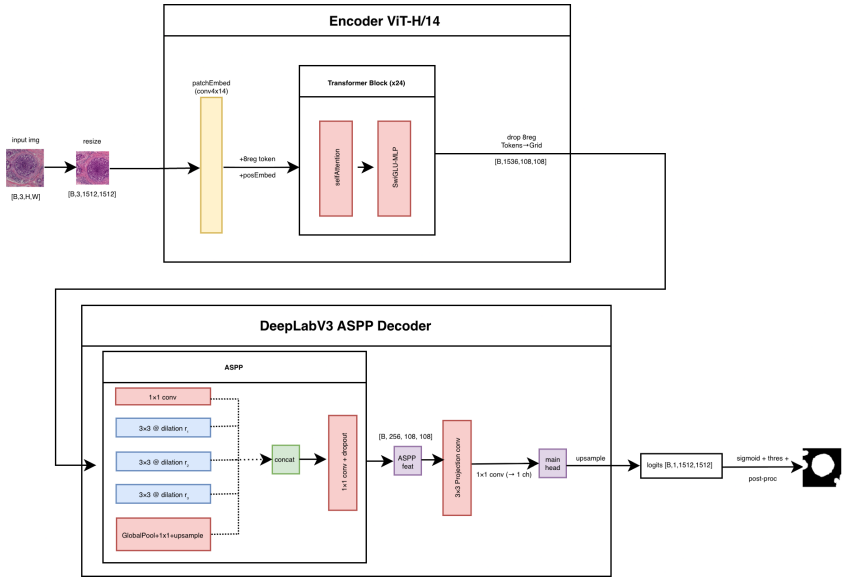

Figure 5.2: Architecture of UNI2-h and DeepLabV3-inspired model

5.1.3 Dual-path fusion and DeepLabV3-style ASPP Decoder

To be able to capture global and local context, we create a dual-path fusion mechanism before the feature map enters the DeepLabV3 decoder. Basically the same input image is processed at two resolutions using the shared UNI2-h encoder through two forward passes: a high resolution path (1512×1512 pixels) that preserves fine nuclei details and a low resolution path (756×756 pixels) that preserves broader information on tissue structure. Both streams produce token grids and then are merged through a cross-attention fusion block, creating a feature map of [B,1536,108,108]. This fused representation is then passed into the DeepLabV3 ASPP decoder, and after passing the decoder stage it is upsampled and produces the final segmentation mask.

The motivation for this dual-resolution design stems from the difficulty of representing nuclei-level detail and tissue-level organization within a single scale. Prior works in medical and histopathology segmentation have shown that cross-scale fusion improves boundary precision and contextual stability across imaging domains [36, 37].

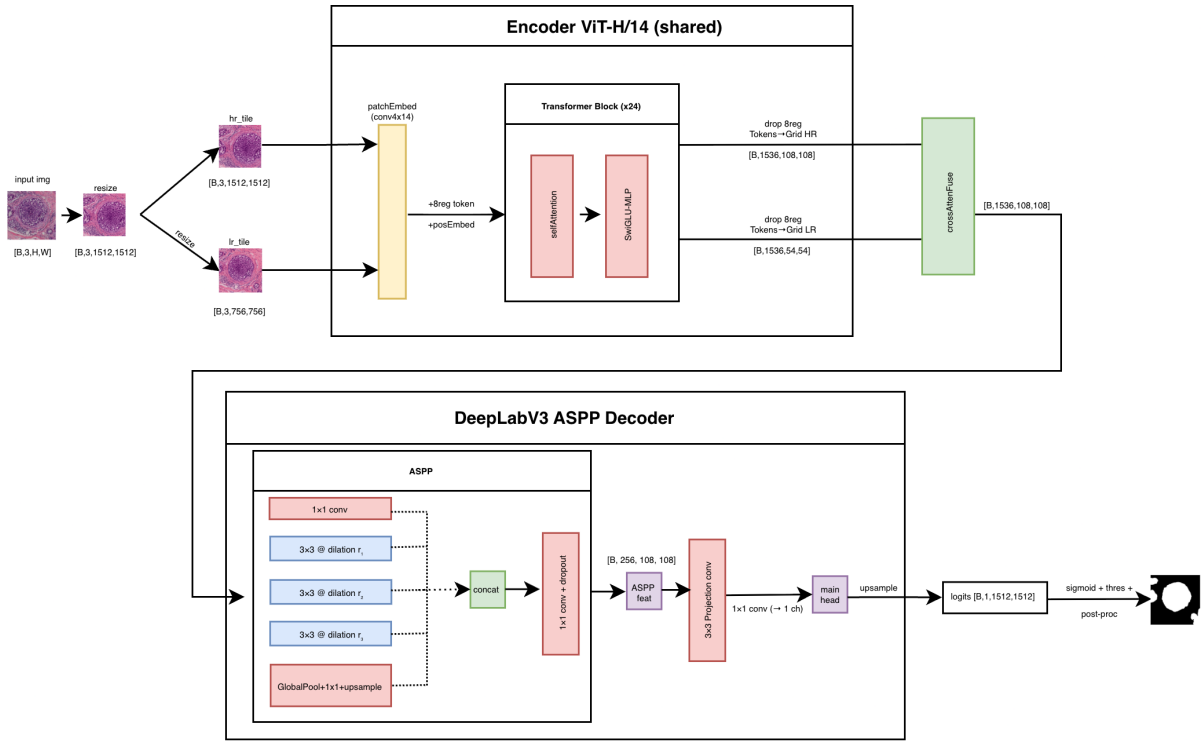

Figure 5.3: Architecture of the DeepLabV3 proposed model

5.1.4 Pixel Feature Pyramid and Query-Based Decoder

After encoding, the output token grid (108×108×1536) is passed through a Pixel Feature Pyramid Network (Pixel FPN) [38], which serves as a bridge between the transformer's token representation and the dense spatial reasoning required for segmentation. The Pixel FPN compresses the channel dimension to 256 using a 1×1 convolution and constructs a small hierarchy of feature maps at three scales:

- P3 (108×108): fine-grained texture and boundary information
- P4 (54×54): mid-level structure patterns
- P5 (27×27): coarse global context

Each feature level goes through Group Normalization (GN) [39] and a GELU activation [34], which helps the model stay stable and efficient without adding much extra computation.

The feature pyramid lets the model look at the tissue from multiple "zoom levels". This means it picks up details such as cell borders while also recognizing more general and high-level structures such as tissue patterns. DeepLab and U-Net work in a similar way, combining information from multiple layers to balance the trade-off between details and global context.

After that, the model passes these multi-scale features into a Lite Transformer Decoder, which is a lighter version of the one used in Mask2Former [40]. The decoder uses learnable queries, and each one acts a bit like a marker that looks for a possible region in the tissue. These queries check all levels of the feature pyramid at once, using positional and scale information to decide where to focus and how much detail to capture.

This setup lets the model look at both the small details and the overall structure of the tissue at the same time, without having to manually combine results from different zoom levels. The approach follows recent trends in universal segmentation frameworks such as Mask2Former and MedSAM [41], which demonstrate that cross-scale feature aggregation with query-based decoding leads to higher accuracy levels and domain robustness.

Each query then gives two things:

1. a classification score that tells how confident the model is that it found a tumor region and
2. a mask embedding, which describes what that region looks like spatially

To build the final segmentation map, the model compares all the query embeddings with the pixel features and selects the top-k most confident ones (Top-K fusion).

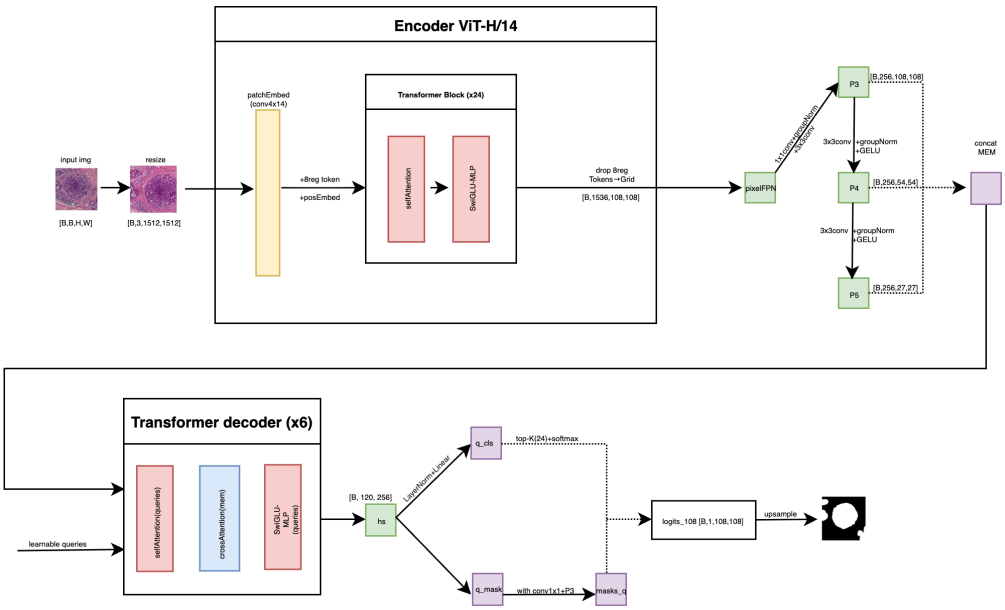

Figure 5.4: Architecture of the UNI2-h + PixelFPN + Transformer Decoder model

5.2 Training Strategy

5.2.1 Data split

The training set of the COSAS 2024 dataset contains 180 images from three scanners: 3DHISTECH 1000, KFBIO 400 and TEKSQRAY 600P.

An 80/20 split was applied to both sets, ensuring each scanner contributed equally to them. In this way, the training process avoids bias towards a specific device.

5.2.2 Loss Design

Segmentation quality in histopathology depends on region overlap, boundary precision and robustness to class imbalance. To address these, we implement the composite loss function:

$$\mathcal{L}_{total} = 0.5 \times \mathcal{L}_{bce} + 0.5 \times \mathcal{L}_{dice} \tag{5.1}$$

Each term serves a distinct role in improving segmentation performance:

- **Binary Cross-Entropy (BCE)** ($\mathcal{L}_{bce}$) [42] ensures stable pixel-level optimization, particularly under severe class imbalance where nuclei occupy a small fraction of the image
- **Soft Dice loss** ($\mathcal{L}_{dice}$) helps the model improve the overlap between its predictions and the ground-truth masks, guiding it to better capture the overall shape of the tumor regions

This hybrid approach is commonly used in biomedical image segmentation, as it provides a practical balance between pixel-level precision and overall shape consistency. In the experiments, assigning equal weights (0.5–0.5) to the BCE and Dice terms resulted in stable convergence across scanners. A number of other losses were tested but did not yield better results; these will be analyzed in the following chapter.

5.2.3 Optimization

AdamW optimizer was used with layer-wise learning rate decay, assigning lower learning rates to early ViT layers and higher ones to the segmentation head. The base learning rate was 1×10^{-4} and the cosine learning rate schedule was applied following a 200-iteration warmup.

Training ran for 50 epochs, using gradient accumulation, mixed precision (FP16), and gradient clipping to maintain stability.

The encoder was frozen for the first 5 epochs, allowing the decoder to stabilize, and then the top 8 transformer blocks were unfrozen for fine-tuning. A CPU-based exponential moving average (EMA) of model weights was maintained and used for validation and checkpointing to reduce the effect of gradient noise.

5.2.4 Computational Resources

All experiments were conducted on an NVIDIA A100 GPU (40 GB VRAM) using PyTorch. The training environment was based on Ubuntu 22.04 with CUDA 12.1 and cuDNN 8.9. Each model was trained using mixed-precision (FP16) to optimize memory efficiency and speed.

5.3 Evaluation Strategy

5.3.1 Metrics

The evaluation followed the COSAS 2024 protocol metrics. The model was trained using images from a subset of scanners and validated on the remaining unseen scanners. Two metrics were used for assessment:

- **Dice Similarity Coefficient (DSC)**, and
- **Jaccard Similarity Coefficient (JSC)**, also known as the Intersection over Union (IoU)

The Dice coefficient measures the spatial overlap between the predicted mask B and the ground-truth A and is defined as:

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

A Dice score of 1 means perfect agreement, while a score of 0 means no overlap.

In the context of adenocarcinoma segmentation, Dice emphasizes correct identification of both tumor and boundary pixels, rewarding predictions that capture the shape and extent of the tumor regions.

The Jaccard coefficient evaluates the intersection-over-union between the prediction and the reference annotation:

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

Compared to Dice, Jaccard is more conservative and penalizes small mismatches more strongly. The combined score was computed to balance both region and boundary accuracy:

$$\text{Score} = 0.5 \times \text{DSC} + 0.5 \times \text{JSC}$$

The challenge consisted of two evaluation phases, the preliminary one and the final test one. The final ranking for each team was determined using a weighted combination of the two phases:

$$\text{Final Score} = 0.2 \times \text{Score}_{\text{Preliminary}} + 0.8 \times \text{Score}_{\text{Final}}$$

5.3.2 Thresholding & Post-processing

After training, the model outputs a probability map, and instead of using random threshold, we fix it and tune it in post-processing of the validation set.

Validation results were saved beforehand and we performed some extra image flips and rotations to make predictions more stable, so that we could try different threshold strategies without rerunning the model each time.

A three-stage search was performed:

Stage 1 - Coarse search:

We test threshold values between 0.50 and 0.78, if for example the model's probabilities were best calibrated at 0.65.

For each value, we tried small morphological operations (light openings and closings of the mask to remove small false positives or fill small nuclei holes). Before binarization, a mild Gaussian smoothing was optionally applied to the probability map, effectively blending global and local thresholding behavior: the global cutoff remains fixed, but smoothing allows local intensity variations to influence the final mask boundaries.

Stage 2 - Refinement:

Next, we focused on the best candidates from the first round by making small adjustments to the threshold and filter sizes and ran a second grid search with numbers close to the Stage 1 results.

Stage 3 - Final selection:

Finally, we test the best setups on the validation set and pick the ones that gave the best score. This final combination of threshold and cleaning steps was then used for all the test images.

The final post-processing pipeline was kept simple:

1. Binarize the probability map using the chosen global threshold (optionally with a light Gaussian smoothing)
2. Apply morphological closing (radius 2) to close small gaps, and optionally a light opening (radius 0-1) to remove small speckles
3. Remove small regions and fill small holes using relative area thresholds, scaled to image size so that it works across scanners
4. Optionally clear edge pixels to eliminate border artifacts

This was purely a fixed inference setup, determined once based on validation results, and then applied in exactly the same way to both test sets.

5.4 Baselines and SOTA

The goal of the thesis was not to outperform the leaderboard solutions of the COSAS competition, but to investigate how foundation models work for domain-generalized tumor segmentation. The comparison with state-of-the-art methods serves to contextualize the setup. The actual comparison is between the model presented in the thesis and the original foundation models. However, it is important to mention the top three best scores of the competition.

5.4.1 Winning competition teams

DeepMicroscopy Team

The winning solution [43] simply used nnU-Net, a fully automated segmentation framework that adapts its preprocessing, architecture, and training parameters to the dataset. Their performance highlights how a well-tuned architecture can thrive in such task.

DeepMicroscopy Team

The second-place team [44]used UperNet with a Visual Attention Network (VAN) backbone. They split training data by scanner type and combine multiple models to improve performance. Their success shows that testing and augmenting data in a real-world way when it comes to scanner differences is needed to build such models.

Zhijian Life Team

The third-place team [45] used a large pretrained vision model (DINOv2) as a frozen backbone and added a few small adapter layers along with a segmentation head on top. Instead of fine-tuning the entire network, they only trained these added components, which made the process much lighter and faster while still delivering strong performance.

Table 5.1: COSAS Task 2 Final Results (September 2024)

Rank	Team	Score
1st	deepmicroscopy	0.8527
2nd	Bio-Totem	0.8354
3rd	Zhijian Life	0.8192
4th	Cross-Domain Adenocarcinoma Segmentation	0.8175
5th	Amaranth	0.8128
6th	Team-Tiger	0.8093
7th	ICT_team	0.7944
8th	SMF	0.7924
9th	DeepLearnAI	0.7597
10th	Sanmed_AI	0.7420

5.4.2 Thunder

Unlike competition teams who optimized for the best leaderboard scores, this thesis took a scientific approach inspired by the THUNDER benchmark [46] by MICS Laboratory, focusing on understanding how large pretrained pathology models behave across datasets.

THUNDER (Tile-level Histopathology Image UNDERstanding) is an open-source benchmark designed to standardize the evaluation of digital pathology foundation models. It assesses models along three key dimensions: downstream performance, feature-space analysis, and robustness to uncertainty and perturbations. By operating at the tile level across 23 foundation models and 16 public datasets, it isolates each model's representational power and promotes transparent, reproducible comparison. This philosophy guides the experimental design of this thesis, emphasizing interpretability, robustness, and fair evaluation over leaderboard optimization.

The COSAS leaderboard served mainly as context, while the core analysis centered on foundation model robustness and domain generalization testing whether these models can truly act as universal, stain-agnostic backbones for pathology segmentation.

Chapter 6

Work Carried Out and Results

6.1 Experiments

Before building the segmentation architecture, we evaluate the performance of the top existing foundation models in medical image segmentation provided by the Thunder paper. This step was essential to get an understanding of the current state-of-the-art models' ability to be scanner-agnostic in the specific segmentation task.

The models were accessed using the Thunder API [46] and UNI2-H, UNI, Virchow2 and Midnight scored among the highest. However, they still scored below any of the top 5 teams.

Table 6.1: COSAS Task 2 Final Results alongside foundation models

Rank	Team / Model	Score	DSC/F1	Jaccard
1	deepmicroscopy	0.8527	–	–
2	Bio-Totem	0.8354	–	–
3	Zhijian Life	0.8192	–	–
4	agaldran (Cross-Domain Adenocarcinoma Segmentation)	0.8175	–	–
5	Amaranth (Nameeta)	0.8128	–	–
6	Team-Tiger (D_prasad)	0.8093	–	–
7	ICT_team (winycg)	0.7944	–	–
8	SMF (hoheon0509)	0.7924	–	–
9	**UNI2-H**	**0.7655**	**0.812**	**0.719**
10	DeepLearnAI	0.7597	–	–
11	**Virchow2**	**0.7555**	**0.804**	**0.707**
12	**Midnight**	**0.7525**	**0.801**	**0.704**
13	**UNI**	**0.6980**	**0.750**	**0.646**
14	Sanmed_AI (ksanmed)	0.7420	–	–

Additionally, as illustrated in Figure A.1, even the best-performing model (UNI2-H) still showed some shortcomings: while it captured the general structure reliably, it tended to over-

smooth boundaries and occasionally merge closely adjacent regions. So, to preserve tissue boundaries, foundation models need improvement for the cross-scanner segmentation task. From these comparisons, UNI2-H was the most promising model and was chosen for this reason.

6.1.1 UNI2-H with DeepLabV3-style Decoder (No Auxiliary Output)

After establishing the thunder baseline, the next stage introduced a DeepLabV3-style Atrous Spatial Pyramid Pooling (ASPP) decoder, designed to enhance spatial consistency and multi-scale reasoning. As presented in the methodology section, it connects directly to the UNI2-h encoder feature map and applies parallel dilated convolutions with rates of 12, 24, and 36, alongside a 1×1 convolution and a global pooling branch. These features are concatenated and projected through Group Normalization and GELU activation.

The model reached early stopping at epoch 47, using 11.63 GB per epoch. After post-processing refinement, the model achieved a combined COSAS score of 0.8209, making it a strong first setup with minimal changes.

6.1.2 UNI2-H with DeepLabV3-style Decoder (With Auxiliary Output)

Following the ASPP setup, the next step integrated the auxiliary output branch, an element used in DeepLabV3 and DeepLabV3+ architectures to stabilize training and improve gradient flow. This head is connected to intermediate ASPP features and produced an extra segmentation prediction during training. So the two outputs are then combined using a weighted loss, allowing earlier layers in the decoder to receive more direct supervision.

The total loss function was defined as a weighted sum of the main and auxiliary losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \lambda_{\text{aux}}\, \mathcal{L}_{\text{aux}}, \tag{6.1}$$

where $\lambda_{\text{aux}} = 0.4$

Early stopping was reached at epoch 43 with a peak GPU memory usage of 11.64 GB. After post-processing, the final COSAS score improved to 0.8204, showing a steady gain over the single-path setup, but still inferior to the simple-loss setup.

6.1.3 UNI2-h with Low-Resolution Fusion and DeepLabV3 Decoder

In this experiment we extended the single-scale DeepLabV3 setup with a dual-path fusion stage. The same input image was sent through the UNI2-h encoder at two forward passes: a high-resolution stream (1512×1512) to preserve fine nuclear textures, and a lower-resolution stream (756×756) to capture larger tissue patterns. Both streams produced feature maps, which were then merged using a cross-attention block. In this step, the high-resolution tokens query the low-resolution ones, allowing detailed features to incorporate broader spatial context. The

fused representation (B×1536×108×108) was then passed to the standard DeepLabV3 ASPP head for segmentation.

Training proceeded smoothly with the same settings and simplified loss (no auxiliary design), with validation metrics stabilizing near the end (Epochs 48–50: DSC ≃ 0.8227–0.8229, JSC ≃ 0.6894–0.6897, combined score ≃ 0.756). As expected, GPU memory usage increased to around 19.4 GB compared to roughly 11.6 GB for the single-scale setup, mainly because of the second encoder pass.

Table 6.2: Validation metrics every 10 epochs

Epoch	**Dice**	**Jaccard**	**Combined Score**
1	0.3329	0.2114	0.2721
10	0.6994	0.6024	0.6509
20	0.7835	0.6738	0.7287
30	0.8168	0.6796	0.7482
40	0.8218	0.6886	0.7552
50	0.8229	0.6897	0.7563

To improve the final predictions, we performed the post-processing explained in the methodology chapter on the validation set. This search explored 3024 combinations of thresholds and morphological parameters in Stage 1, followed by refined combinations in Stage 2. The best configuration was found at a threshold of 0.4, with opening and closing radii of 0 and 2 pixels, respectively. The minimum object area fraction was 0.0006 while the minimum hole area fraction was 0.0024. Finally, the gaussian smoothing was found at 0.6.

Table 6.3: Performance summary across scanners

Domain	**DSC**	**JSC**	**Score**
Preliminary Phase			
3D-1000	0.9029	0.8249	0.8639
KFBIO-400	0.9353	0.8784	0.9069
Leica-450	0.8460	0.7500	0.7980
Motic	0.8854	0.8035	0.8445
PreliminaryMean	0.8817	0.7982	0.8399
Final Test Phase			
3D-1000	0.8785	0.7908	0.8347
3D-250	0.9138	0.8495	0.8817
KFBIO-400	0.8826	0.8013	0.8419
Leica-450	0.8493	0.7557	0.8025
Motic	0.7921	0.6727	0.7324
Teksqray-600p	0.8451	0.7744	0.8097
FinalTestMean	0.8602	0.7741	0.8171
Final Combined Score ($0.2 \times \mathit{Prelim} + 0.8 \times \mathit{Final}$)			0.8217

As seen in Table 6.3, mean scores reached 0.8399 in the Preliminary phase (DSC = 0.8817, JSC = 0.7982) and 0.8171 in the Final Test phase (DSC = 0.8602, JSC = 0.7741), giving a combined final score of 0.8217. Domain-level patterns were as expected: scanners that benefit from larger structural context (such as Leica-450) showed clearer, more coherent predictions (final score $\simeq$ 0.8025), while those dominated by compact, high-contrast nuclei performed about the same as the single-scale version.

As seen in (Figure 6.1), we can confirm the quantity results, as the model has a strong segmentation performance under various stains and across very different scanners. In most domains, particularly in 3d-1000, kfbio-400, and teksqray-600p the predictions align almost perfectly, which is expected because the model was trained on different images of these scanners. Nevertheless, imperfections remain visible. Especially in Leica-450 and Motic scanners, masks are over-segmented at times and the model mixes up nearby areas or smooths out thin tissue lines, so it needs to focus more on fine details.

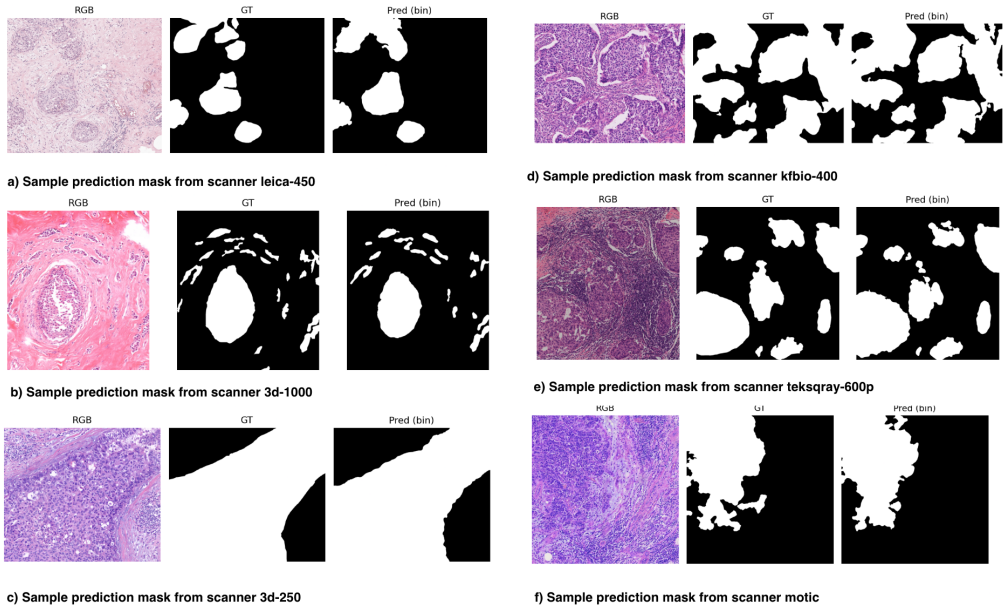

Figure 6.1: Sample prediction masks from multiple scanners showing RGB image, ground truth (GT), and binary prediction (Pred)

Compared to the DeepLabV3 model with an auxiliary head (final = 0.8204), this fusion setup delivered a small but consistent improvement, without the extra supervision or loss terms. This suggests that directly exchanging information between scales is a more reliable way to introduce context than relying on auxiliary gradients, especially in cross-scanner adenocarcinoma segmentation.

6.1.4 UNI2-H with PixelFPN + Simple 1×1 Decoder Head

The goal now was to figure out if the multi-scale context could be achieved by avoiding two forward passes and reducing computational cost. The PixelFPN + 1×1 decoder head design

builds a lightweight feature pyramid directly from the UNI2-h encoder's output. Instead of explicitly combining high and low resolution branches, the model generates a hierarchy of feature maps (P3–P5) by progressively downsampling and processing the encoded features so we have information on different scales. Each level is refined through shallow convolutions and then upsampled back to a shared resolution, where they are summed to restore fine spatial detail. Finally, a single 1×1 convolution projects the fused feature map into the segmentation space, producing the predictions.

Training followed the same configuration as earlier baselines, with the model training smoothly for 50 epochs, stabilizing around a validation Dice of 0.8055 and Jaccard of 0.7041 and a peak memory use of just 11.36 GB. After post-processing, it reached a combined COSAS score of 0.8130, slightly lower than the dual-path fusion result (0.8217) but still competitive given its lower computational cost.

6.1.5 UNI2-H with PixelFPN + Atrous Spatial Pyramid Pooling (ASPP)

After the simple PixelFPN model, we introduce an Atrous Spatial Pyramid Pooling (ASPP) block. The intuition behind it was that this Deeplab-style decoder could yield better results because ASPP can capture features at multiple dilation rates, as previously discussed.

The model constructs its feature hierarchy (P3–P5) directly from the UNI2-h encoder output, generating a coarse-to-fine pyramid of spatial features. These are merged to form a unified multi-scale feature map, which is then passed through a DeepLabV3-style ASPP module. The ASPP block includes parallel atrous convolutions (rates = 12, 24, 36), a 1×1 convolution, and a global pooling branch, all followed by Group Normalization and GELU activation. A final 1×1 convolution projects the fused representation into the segmentation space, keeping computation low while improving contextual coverage.

The model trained for 50 epochs, reaching a validation Dice of 0.8048 and Jaccard of 0.7034, with a peak GPU memory use of 11.41 GB, similar to the simpler PixelFPN experiment. After post-processing, the final COSAS score reached 0.8129, nearly matching the PixelFPN + 1×1 head (0.8130) but not surpassing it.

6.1.6 UNI2-H with PixelFPN + Query-Based Multi-Scale Decoder

After experimenting with the PixelFPN + ASPP model, the next step was to see whether we could keep the same level of multi-scale awareness but make it more adaptive and less hand-crafted. ASPP is effective at expanding local context using several fixed dilation rates, but it's a static, pixel-centred operator, so every location processes information in exactly the same way. In histopathology, the hardest cases usually aren't about simply "seeing farther", they're about structure. Nuclei tend to touch or overlap, gland boundaries can be messy and irregular, and tissue density often changes from one scanner to another. Since ASPP blends features in the same way everywhere, it can easily blur thin edges, overfill crowded regions, and miss subtle

connections between areas that actually belong together. In short, ASPP increases the receptive field but doesn't really organize well what it sees.

To address this, we replaced the ASPP block with a Lite Transformer query decoder (Figure 5.4). The goal was to keep the single encoder pass while giving the model the ability to decide what to focus on and where to look and also test whether the PixelFPN backbone still holds its ground. The input image still goes through UNI2-h encoder, whose output is reshaped into a 2D feature map and reduced to 256 channels. Two downsampling layers with stride = 2 generate the coarser feature levels (P4 and P5), progressively reducing spatial resolution to capture larger tissue patterns. Then these three levels are refined through group normalisation and GELU, and the respective feature maps are concatenated into one sequence of tokens, the "memory" that the transformer decoder reads from when figuring out where to pay attention to.

The transformer decoder with four layers and eight attention heads uses 192 learnable queries to interpret the multi-scale feature maps and generate confidence scores and mask embeddings. For the task, the number is a good balance as it's three times higher than the 64 queries kept at inference, giving the model room to explore different potential regions during training without becoming redundant. The ratio isn't fixed, but it's a safe proposal to avoid overfitting.

For mask formation, the model keeps the pixel space at the finest level (P3). A 1×1 projection maps P3 into a mask basis, and each query combines its embedding with these pixel features through a weighted sum operation to generate a soft mask. The query confidences are normalized using a softmax function and used to blend the per-query masks into a single logit map, which is then upsampled to the original image resolution. A Top-K selection (K=64) step is applied before blending to reduce redundancy and stabilize training by retaining only the most confident queries. This design keeps the efficiency of a single encoder pass while preserving explicit multi-scale evidence through the pyramid. It also adds an object-level grouping mechanism via the queries, leading to sharper boundaries, cleaner suppression of false activations, and more reliable predictions across scanner domains.

The model converged at 50 epochs, reaching a validation Dice score of 0.8029, a Jaccard score of 0.6999, and a combined validation score of 0.7514, with a GPU memory of 15.77 GB per epoch. After post-processing, the model achieved a final score of 0.8098, slightly below the PixelFPN + ASPP result (0.8129) despite being faster and simpler.

Table 6.4: Experiments summary table

Model Variant	Score
UNI2-h + DeepLabV3 (no aux)	0.8209
UNI2-h + DeepLabV3 (with aux)	0.8204
UNI2-h + Low-Res Fusion + DeepLabV3	**0.8217**
UNI2-h + PixelFPN + 1×1 Head	0.8130
UNI2-h + PixelFPN + ASPP	0.8206
UNI2-h + PixelFPN + Lite Query	0.8176

6.1.7 Losses experiments

After having chosen the best-working setup (UNI2-h with Low-Resolution Fusion and DeepLabV3-inspired decoder) we ran an 8-epoch comparison of several loss functions to check whether or not it would stabilize the training. The Tversky and Focal variants provided only marginal gains ($\Delta < 0.004$) compared to the BCE + Dice baseline. These improvements are too small to justify a change in the loss formulation, confirming that BCE and Dice remains the most efficient choice for this setup.

Table 6.5: Comparative results of different loss formulations (UNI2-h + LowResFusion + DeepLabV3).

Loss Variant	Best Score	Best Thr.	Δ vs. Baseline
BCE + Dice (baseline)	0.7811	0.65	–
+ Tversky ($\alpha = 0.3, \beta = 0.7, w = 0.30$)	0.7843	0.50	0.0032
+ Focal ($\gamma = 1.5, w = 0.25$)	0.7842	0.50	0.0031

6.2 Quantitative Benchmarking - Comparison with COSAS Challenge

While the COSAS challenge permitted the use of models pre-trained on large-scale non-medical datasets such as ImageNet and MS COCO, no external pathology data were allowed for model development. In contrast, the UNI2-h backbone used in this work was pre-trained exclusively on medical histology data - over 200 million tiles from 300,000+ H&E and IHC slides collected at Mass General Brigham. As outlined earlier, the aim of this study was to explore how foundation models can be fine-tuned for cross-scanner adenocarcinoma segmentation, so to put things into perspective it's useful to compare our models against COSAS challenge leaderboard.

In the official leaderboard, the top-performing team (DeepMicroscopy) reached a COSAS score of 0.8527, followed by Bio-Totem (0.8354) and Zhijian Life (0.8192), as we have analyzed

in Chapter 5. Among our configurations, the best-performing model was UNI2-h + Low-Res Fusion + DeepLabV3-inspired decoder which achieved 0.8217, positioning it closely with the third-place team despite using no non-medical pretraining and less architectural complexity. Other variants, including PixelFPN + ASPP (0.8206) and UNI2-h + DeepLabV3 (0.8209), also ranked competitively around the top 4-5 range, suggesting strong generalization across unseen scanners.

Although many top COSAS entries used multiple models, heavy augmentations, and non-medical pretraining, our single UNI2-h model - trained only on pathology data - came within about 3–4% of the top score. This shows that UNI2-h already captures the kinds of visual patterns that matter for adenocarcinoma segmentation, making it naturally well-suited for cross-scanner generalization.

Table 6.6: Comparison with official COSAS challenge leaderboard. UNI2-h experimentative models are shown in bold

Rank / Model	**Final COSAS Score**
1st DeepMicroscopy	0.8527
2nd Bio-Totem	0.8354
3rd Zhijian Life	0.8192
– **UNI2-h + Low-Res Fusion + DeepLabV3 (ours)**	**0.8217**
4th Cross-Domain Adenocarcinoma Segmentation	0.8175
– **UNI2-h + DeepLabV3 (no aux)**	**0.8209**
– **UNI2-h + PixelFPN + ASPP**	**0.8206**
– **UNI2-h + DeepLabV3 (aux head)**	**0.8204**
– **UNI2-h + PixelFPN + Lite Query Decoder**	**0.8176**
5th Amaranth	0.8128
– **UNI2-h + PixelFPN + 1×1 Head**	**0.8130**
6th Team-Tiger	0.8093
7th ICT_team	0.7944
8th SMF	0.7924
9th DeepLearnAI	0.7597
10th Sanmed_AI	0.7420
– **UNI2-h (baseline thunder model)**	**0.67655**

Chapter 7

Critical Discussion

7.1 Analysis of the results

7.1.1 Strengths of the developed approach

The main strength of this thesis is in the evaluation pipeline, which is fair across models. Each architecture is trained with the same schedules, augmentations and epochs to ensure consistency and to better showcase the difference that can be based only on architectural variations.

Post-processing optimization is also key, as the multi-stage threshold and morphology sweep in plenty of combinations provided a great way to fine-tune the model without re-running it. This added interpretability point to the pipeline, showing how performance gain can be achieved solely through post-processing.

The UNI2-h encoder turned out to be a great fit for this task. From the large-scale histopathology pretraining to the COSAS challenge, it managed to prove itself with no major architectural changes. The attention layers picked up stain-invariant and consistent tissue patterns, creating features that were fitted for the following decoding process.

Pairing UNI2-h with a Dual-Resolution Fusion and DeepLabV3-inspired decoder was the right choice: the fusion of high and low resolution paths alongside the Deeplab-ASPP inspired decoder helped achieve the highest overall score (0.8217). This was particularly beneficial for specific scanners such as Leica and Motic, where boundaries are less distinct.

Finally, another strength of this study lies in how thoroughly the experiments were explored. Different setups were tested and every round of experiments added a new understanding of what influenced the model's performance and lead to a final configuration that's balanced.

7.1.2 Limitations of the study

The first limitation comes from computational cost. Even though the proposed architectures were lighter than transformer-heavy alternatives, memory requirements still ranged between 11 and 19 GB per GPU.

Another important challenge is domain imbalance. Although the COSAS dataset includes six scanners, there was not even distribution across testing splits. During the preliminary stage, which influences the final score by 20%, scanners such as leica and motic were under-represented, leaving the model with fewer testing examples. This phenomenon was also present in the training splits, where the model is less able to learn as well color and texture variations. This uneven exposure explains the weaker accuracy in certain scanners. In the final test phase, the data were balanced with 15 patches per scanner, which made the evaluation fairer for the remaining 80% of the overall score.

Finally, we used static validation sets and cached probability maps, which made testing quicker but it might have slightly overstated the effects of post-processing. This didn't affect which models performed best, but it could introduce a small difference between validation and test results.

7.2 Sources of Bias and Experimental Considerations

Scanner and Patch Distribution

The COSAS dataset is unevenly distributed across scanners, as we mentioned in the previous section. As shown in Figure A.2, tumor coverage also varied between domains: 3D-Histech and 3D-250 patches contained larger tumor regions (median ≈0.5–0.6), whereas Leica, Motic, and Teksqray had smaller ones (≈0.3–0.4). Because of this, the model saw more examples of "dense tumor" tissue from the dominant scanners, which helped calibration there but slightly reduced accuracy for the less-represented domains.

Stain and Intensity Variation

The mean RGB intensity plots (Figures A.3 A.4) show color and brightness differences between scanners. 3D-Histech and KFBIO images had stronger and more consistent intensities, while Teksqray and Motic showed dimmer staining. Since color augmentation and normalization were applied globally and not separately per domain these differences may have biased the model toward the brighter scanners. This explains why predictions for Leica and Motic often looked smoother but occasionally missed some nuclei boundaries.

Validation-Tuned Post-Processing

Post-processing parameters, such as binarization threshold and morphological filters, were tuned on the validation subset, which shared the same imbalance as the training data. As a result, the global threshold may have been slightly better suited to the dominant scanners, since their probability distributions influenced the calibration process more strongly.

Experimental Constraints

As we mentioned earlier, the experiments were constrained by computational and data limitations. The UNI2-H encoder combined with its different decoder variants required between 11 and 19 GB of GPU memory, which limited the depth of architectures and the batch sizes that could be tested. In addition, because the dataset was relatively small (180 patches of 1500×1500 px), random effects like batch sampling or learning-rate scheduling may have had a bigger impact on training stability and generalization.

7.3 Cross-Scanner performance and architectural comparison

Across all models, the scanners that consistently perform best are those with bright, warm tones and higher tumor coverage, namely 3d-250 and 3d-1000. Even though 3d-250 was unseen during training, it's visually similar to 3d-1000, which helps the model generalize well. For instance, in the UNI2-h + DeepLabV3 (no-aux) setup, the final test scores were 0.8801 and 0.8355 for 3d-250 and 3d-1000 respectively, and this pattern holds across the other variants: DeepLabV3+Aux (0.8746 / 0.8345), PixelFPN+ASPP (0.8792 / 0.8355), and Lite Query (0.8730 / 0.8341). These results fit the dataset trends: brighter staining and richer tumor content make boundaries easier to detect and global thresholding more stable.

KFBIO-400 stays somewhere in the middle but remains remarkably consistent across models (0.8436 no-aux; 0.8514 aux; 0.8447 FPN+ASPP; 0.8521 Lite Query). As a scanner seen during training and one with stable illumination and color tone, it benefits from good calibration. Here, architectural changes mainly affect how logits are scaled rather than how well tumors and background are separated.

The harder scanners are Leica-450 and Motic, both unseen during training and characterized by paler or darker tones and lower tumor ratios. Every model struggles here: DeepLabV3 (no-aux) gives 0.7991 / 0.7288, DeepLabV3+Aux is similar (0.7939 / 0.7274), PixelFPN+ASPP slightly improves (0.7981 / 0.7275), Lite Query is close (0.7872 / 0.7211), and PixelFPN+1×1 drops the most (0.7634 / 0.7023). This follows what was observed in dataset exploration, as pale or dim staining leads to weaker edges and more diffuse boundaries. Architectures that rely on pixel-level aggregation, like ASPP or 1×1 heads, tend to over-smooth or under-segment in these conditions. The query decoder helps somewhat by grouping spatially related regions, but its calibration still struggles with unseen, low-contrast domains.

Teksqray-600p stands apart. It's a seen scanner but darker and higher in contrast. Most models land around 0.81 (no-aux 0.8110, aux 0.8099, FPN+ASPP 0.8106, Lite Query 0.8092), yet PixelFPN+1×1 jumps to 0.8654. That jump makes sense: its simple fusion and clean calibration preserve the sharp edges typical of Teksqray images, making its probability maps easier to threshold and refine. Essentially, this scanner rewards sharp, well-defined logits over

more complex multi-scale reasoning.

From an architectural perspective:

- Dual-path HR/LR Fusion + DeepLabV3 (0.8217) delivers the strongest overall score by aligning global and local information, which helps the most in scanners like Leica and Motic where structure, not just context size, drives correct segmentation
- PixelFPN+ASPP (0.8206) comes very close with a simpler, single-pass setup-FPN captures scale hierarchy, and ASPP recovers recall in low-contrast regions, making it the best efficiency–accuracy trade-off
- DeepLabV3 (no-aux) (0.8209) remains a strong and stable baseline: solid on 3d-250 and 3d-1000, steady on KFBIO, weaker on Leica/Motic, and unaffected by the auxiliary branch
- PixelFPN+1×1 (0.8130) is the lightweight option: less capable on pale, low-tumor scanners but excelling on high-contrast domains like Teksqray, where clean calibration dominates over context depth
- PixelFPN+Lite Query (0.8176) adds flexible region-level grouping and handles dense clusters well, though its calibration across stain domains still trails behind ASPP and Fusion

In short, the ranking across scanners reflects the dataset's inherent biases. Scanners with more tumor coverage and brighter tones (3d-250, 3d-1000) produce higher scores; unseen domains with pale or dim staining (Leica, Motic) lag unless the model can explicitly organize spatial and scale information (as in Fusion or ASPP); and seen, dark, high-contrast scanners (Teksqray) favor simpler, sharply calibrated decoders. The results reinforce that segmentation performance depends not only on architecture but also on how well it aligns with the visual and structural traits of each scanner.

Chapter 8

Conclusion and Perspectives

8.1 Assessment of Objectives and Achievements

Motivated by the Cross-Organ Cross-Scanner Adenocarcinoma Segmentation (COSAS 2024) challenge, this thesis investigates how foundation vision transformers can be adapted for scanner-agnostic adenocarcinoma segmentation in H&E-stained histopathology images. The goal was to design efficient, generalizable segmentation models that perform consistently across scanners with different color tones, resolutions, and tissue characteristics.

The proposed framework builds on the UNI2-H foundation model and examines several decoder strategies to understand how architectural choices affect segmentation under domain shift. Specifically, the work compares convolution-based, pyramid-based, and transformer-based decoders within a unified training and evaluation setup. All models were trained and tested on the COSAS dataset, which includes patches from six scanners (three seen during training and three unseen at test time). Despite this cross-domain complexity, the final models achieved Dice scores above 0.85 and competition-weighted scores between 0.81 and 0.82, placing them among the strongest submissions in the challenge.

Beyond performance, the experiments provide a detailed view of how multi-scale feature design and content-adaptive fusion influence generalization across domains. Each architecture contributed distinct insights into what drives robustness in histopathological segmentation.

The main contributions of this thesis are summarised as follows:

1. **Comprehensive architectural study** Evaluated multiple decoder designs for the UNI2-H encoder, comparing convolutional, pyramid-based, and transformer-based variants to isolate the effects of multi-scale fusion, dilation, and attention mechanisms on segmentation quality
2. **Cross-scanner generalization** Demonstrated stable performance on unseen scanners (Leica-450, Motic, 3D-250) despite training on only three domains, confirming the robustness of UNI2-H representations to color and stain variation
3. **Domain-specific performance analysis** Identified how scanner characteristics influence

results: bright, high-tumor domains (3D-1000, 3D-250) yield the highest scores, while pale or dark scanners (Leica, Motic) benefit from explicit multi-scale or attention-based decoders

4. **Lightweight, efficient design.** Achieved a balance between accuracy and resource use, with models requiring 11–19 GB GPU memory and completing inference in 20–25 seconds per domain.
5. **Reproducible training and post-processing framework** Developed a unified pipeline using AdamW optimization, cosine scheduling, and BCE + Dice loss, combined with threshold sweeping and morphological refinement for stable boundary predictions.
6. **Practical foundation for future work** Established a flexible, multi-domain segmentation framework that can be extended to domain calibration, stain adaptation, and whole-slide image inference

8.2 Future Work

Future work should aim to scale the dataset to the whole-slide image (WSI) level, since the current patch-based setup limits how well the model can understand full tissue structure and context. While public datasets like PanNuke and CoNSeP are useful for benchmarking, they are still too small and patch-focused to support real WSI-level generalization. Building a larger, multi-institutional dataset that includes different scanners, staining styles, and magnifications would make training and evaluation much more realistic.

Bibliography

[1] Mohamed Khalifa and Mona Albadawy. "AI in diagnostic imaging: Revolutionising accuracy and efficiency". In: *Computer Methods and Programs in Biomedicine Update* 5 (2024), p. 100146. ISSN: 2666-9900. DOI: https://doi.org/10.1016/j.cmpbup.2024.100146. URL: https://www.sciencedirect.com/science/article/pii/S2666990024000132.

[2] Clare McGenity et al. "Artificial intelligence in digital pathology: a systematic review and meta-analysis of diagnostic test accuracy". In: *npj Digital Medicine* 7 (2024). DOI: 10.1038/s41746-024-01106-8. URL: https://doi.org/10.1038/s41746-024-01106-8.

[3] *Breast cancer: Fact sheet*. World Health Organization. 2022. URL: https://www.who.int/news-room/fact-sheets/detail/breast-cancer (visited on 09/22/2025).

[4] Bo Jiang, Li Bao, Shuai He, et al. "Deep learning applications in breast cancer histopathological imaging: diagnosis, treatment, and prognosis". In: *Breast Cancer Research* 26 (2024). DOI: 10.1186/s13058-024-01895-6. URL: https://doi.org/10.1186/s13058-024-01895-6.

[5] Kaiyang Zhou et al. "Domain Generalization: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 1–20. ISSN: 1939-3539. DOI: 10.1109/tpami.2022.3195549. URL: http://dx.doi.org/10.1109/TPAMI.2022.3195549.

[6] S. R. Duenweg et al. "Whole slide imaging (WSI) scanner differences influence optical and computed properties of digitized prostate cancer histology". In: *Journal of Pathology Informatics* 14 (2023), p. 100321. DOI: 10.1016/j.jpi.2023.100321. URL: https://doi.org/10.1016/j.jpi.2023.100321.

[7] Karin Stacke et al. "Measuring domain shift for deep learning in histopathology". In: *IEEE Journal of Biomedical and Health Informatics* 25.2 (2020), pp. 325–336. DOI: 10.1109/JBHI.2020.2993952. URL: https://doi.org/10.1109/JBHI.2020.2993952.

[8] Daisuke Komura, Mitsuru Ochi, and Satoru Ishikawa. “Machine learning methods for histopathological image analysis: Updates in 2024”. In: *Computational and Structural Biotechnology Journal* 22 (2024), pp. 1256–1274. DOI: 10.1016/j.csbj.2024.12.033. URL: https://doi.org/10.1016/j.csbj.2024.12.033.

[9] Ruochen Liu et al. “Exploring Domain Generalization in Semantic Segmentation for Digital Histopathology: A Comparative Evaluation of Deep Learning Models”. In: *Proceedings of the 2024 9th International Conference on Biomedical Signal and Image Processing*. ICBIP ’24. Suzhou, China: Association for Computing Machinery, 2024, pp. 110–116. ISBN: 9798400717970. DOI: 10.1145/3691521.3691537. URL: https://doi.org/10.1145/3691521.3691537.

[10] Yann LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541. URL: https://doi.org/10.1162/neco.1989.1.4.541.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 25. NIPS 2012. 2012, pp. 1097–1105. URL: https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

[13] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical Image Analysis* 42 (2017), pp. 60–88. DOI: 10.1016/j.media.2017.07.005. URL: https://doi.org/10.1016/j.media.2017.07.005.

[14] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations (ICLR)*. arXiv:2010.11929. 2021. DOI: 10.48550/arXiv.2010.11929. URL: https://openreview.net/forum?id=YicbFdNTTy.

[15] Ashish Vaswani et al. “Attention is All You Need”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017. URL: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[16] Mathilde Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”. In: *arXiv preprint arXiv:2104.14294* (2021). URL: https://arxiv.org/abs/2104.14294.

[17] Kaiming He et al. “Masked Autoencoders Are Scalable Vision Learners”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 16000–16009. DOI: 10.1109/CVPR52688.2022.01551.

[18] Maxime Oquab et al. “DINOv2: Learning Robust Visual Features without Supervision”. In: *arXiv preprint arXiv:2304.07193* (2023). URL: https://arxiv.org/abs/2304.07193.

[19] Marc Aubreville et al. *Quantifying the Scanner-Induced Domain Gap in Mitosis Detection*. 2021. arXiv: 2103.16515 [cs.CV]. URL: https://arxiv.org/abs/2103.16515.

[20] David Tellez et al. “Quantifying the Effects of Data Augmentation and Stain Color Normalization in Convolutional Neural Networks for Computational Pathology”. In: *Medical Image Analysis* 58 (2019), p. 101544. DOI: 10.1016/j.media.2019.101544. URL: https://doi.org/10.1016/j.media.2019.101544.

[21] Katarzyna Faryna, Jeroen van der Laak, and Geert Litjens. “Tailoring Automated Data Augmentation to H&E-Stained Histopathology”. In: *Medical Imaging with Deep Learning (MIDL)*. 2021. URL: https://openreview.net/forum?id=p8vY5W9XG2.

[22] Chuanyun Xu et al. “Stain Normalization of Histopathological Images Based on Deep Learning: A Review”. In: *Diagnostics* 13.21 (2023), p. 3405. DOI: 10.3390/diagnostics13213405. URL: https://www.mdpi.com/2075-4418/13/21/3405.

[23] Richard J Chen et al. “Towards a General-Purpose Foundation Model for Computational Pathology”. In: *Nature Medicine* (2024). DOI: 10.1038/s41591-024-02857-3. URL: https://www.nature.com/articles/s41591-024-02857-3.

[24] Eric Zimmermann et al. “Virchow2: Scaling Self-Supervised Mixed Magnification Models in Pathology”. In: *arXiv preprint arXiv:2408.00738* (2024).

[25] Mikhail Karasikov et al. “Training state-of-the-art pathology foundation models with orders of magnitude less data”. In: *arXiv preprint arXiv:2504.05186* (2025). URL: https://arxiv.org/abs/2504.05186.

[26] Liang-Chieh Chen et al. *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017. arXiv: 1706.05587 [cs.CV]. URL: https://arxiv.org/abs/1706.05587.

[27] Jing Wang and Xiuping Liu. “Medical image recognition and segmentation of pathological slices of gastric cancer based on Deeplab v3+ neural network”. In: *Computer Methods and Programs in Biomedicine* 207 (2021), p. 106210. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2021.106210. URL: https://www.sciencedirect.com/science/article/pii/S0169260721002844.

[28] Hangbo Bao et al. *BEiT: BERT Pre-Training of Image Transformers*. 2022. arXiv: 2106.08254 [cs.CV]. URL: https://arxiv.org/abs/2106.08254.

[29] Archimedes AI Research Unit, Athena Research Center. *Archimedes AI – Artificial Intelligence Research Unit*. https://archimedesai.gr/en/. Accessed: October 17, 2025. 2025.

[30] MICS Laboratory, CentraleSupélec, University Paris-Saclay. *Mathematics and Computer Science Laboratory (MICS)*. URL: https://www.centralesupelec.fr/en/mics-laboratory (visited on 10/17/2025).

[31] Daphne Tsolissou et al. *Multimodal Carotid Risk Stratification with Large Vision-Language Models: Benchmarking, Fine-Tuning, and Clinical Insights*. 2025. arXiv: 2510.02922 [cs.CV]. URL: https://arxiv.org/abs/2510.02922.

[32] Andreas Lolos et al. *SGPMIL: Sparse Gaussian Process Multiple Instance Learning*. 2025. arXiv: 2507.08711 [cs.CV]. URL: https://arxiv.org/abs/2507.08711.

[33] Ishaan Gulrajani and David Lopez-Paz. *In Search of Lost Domain Generalization*. 2020. arXiv: 2007.01434 [cs.LG]. URL: https://arxiv.org/abs/2007.01434.

[34] Noam Shazeer. *GLU Variants Improve Transformer*. 2020. arXiv: 2002.05202 [cs.LG]. URL: https://arxiv.org/abs/2002.05202.

[35] Reza Azad et al. *TransDeepLab: Convolution-Free Transformer-based DeepLab v3+ for Medical Image Segmentation*. 2022. arXiv: 2208.00713 [eess.IV]. URL: https://arxiv.org/abs/2208.00713.

[36] Yutong Xie et al. *CoTr: Efficiently Bridging CNN and Transformer for 3D Medical Image Segmentation*. 2021. arXiv: 2103.03024 [cs.CV]. URL: https://arxiv.org/abs/2103.03024.

[37] Qiran Jia and Hai Shu. "BiTr-Unet: A CNN-Transformer Combined Network for MRI Brain Tumor Segmentation". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer International Publishing, 2022, pp. 3–14. ISBN: 9783031090028. DOI: 10.1007/978-3-031-09002-8_1. URL: http://dx.doi.org/10.1007/978-3-031-09002-8_1.

[38] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. 2017. arXiv: 1612.03144 [cs.CV]. URL: https://arxiv.org/abs/1612.03144.

[39] Yuxin Wu and Kaiming He. *Group Normalization*. 2018. arXiv: 1803.08494 [cs.CV]. URL: https://arxiv.org/abs/1803.08494.

[40] Bowen Cheng et al. *Masked-attention Mask Transformer for Universal Image Segmentation*. 2022. arXiv: 2112.01527 [cs.CV]. URL: https://arxiv.org/abs/2112.01527.

[41] Jun Ma et al. "Segment anything in medical images". In: *Nature Communications* 15.1 (Jan. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-44824-z. URL: http://dx.doi.org/10.1038/s41467-024-44824-z.

[42] Usha Ruby and Vamsidhar Yendapalli. "Binary cross entropy with deep learning technique for Image classification". In: *International Journal of Advanced Trends in Computer Science and Engineering* 9 (Oct. 2020). DOI: 10.30534/ijatcse/2020/175942020.

[43] Frauke Wilm et al. *Domain and Content Adaptive Convolutions for Cross-Domain Adenocarcinoma Segmentation*. 2024. arXiv: 2409.09797 [eess.IV]. URL: https://arxiv.org/abs/2409.09797.

[44] Huang Jiayan et al. *Domain-stratified Training for Cross-organ and Cross-scanner Adenocarcinoma Segmentation in the COSAS 2024 Challenge*. 2024. arXiv: 2409.12418 [cs.CV]. URL: https://arxiv.org/abs/2409.12418.

[45] Pengzhou Cai et al. *Cross-Organ and Cross-Scanner Adenocarcinoma Segmentation using Rein to Fine-tune Vision Foundation Models*. 2024. arXiv: 2409.11752 [eess.IV]. URL: https://arxiv.org/abs/2409.11752.

[46] Pierre Marza et al. *THUNDER: Tile-level Histopathology image UNDERstanding benchmark*. 2025. arXiv: 2507.07860 [cs.CV]. URL: https://arxiv.org/abs/2507.07860.

Appendix A

Source Code and Extra Material

This appendix includes qualitative visualizations and supplementary material referenced in the thesis. The full source code is available at: `https://github.com/mkontaratou/Cross-Scanner-Breast-Adenocarcinoma-Image-Segmentation-With-Deep-Learning-Algorithms`

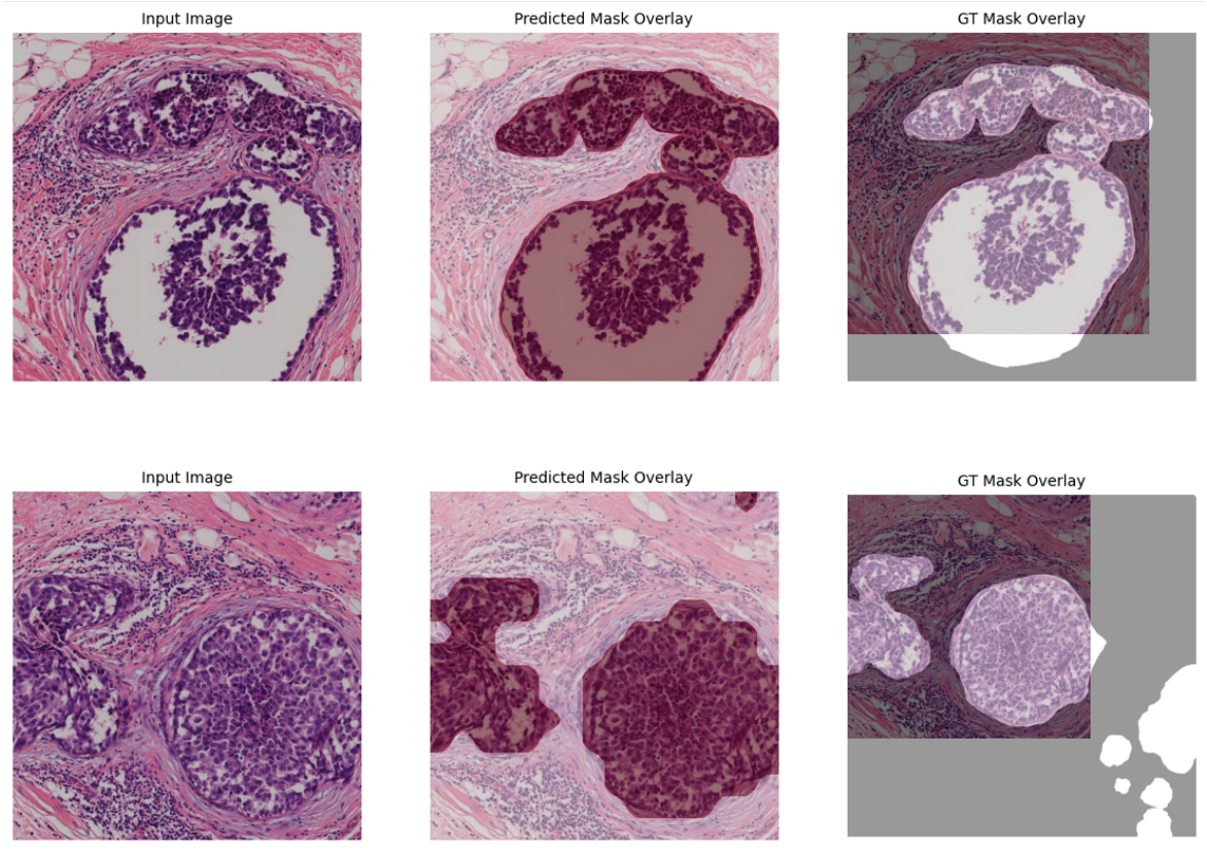

Figure A.1: Example predictions from UNI-2H. Transformer-based architectures demonstrated smooth and consistent segmentation across scanners. It showed minor over-smoothing and occasional merging of nearby glands, suggesting room for refinement in boundary precision

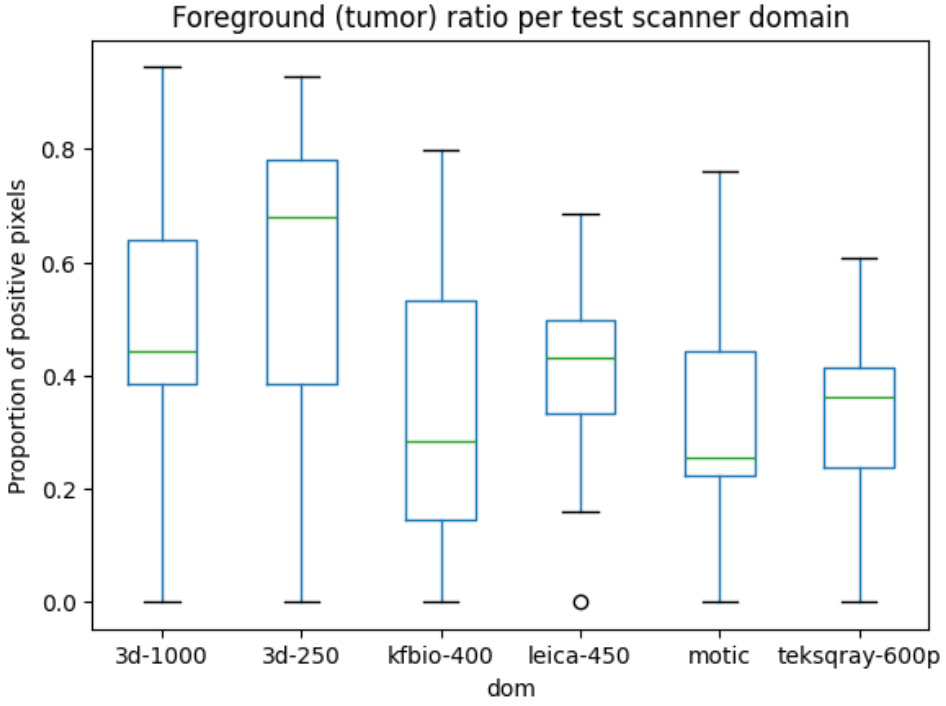

Figure A.2: Foreground (tumor) ratio per test scanner domain. Box plot showing the proportion of positive (tumor) pixels across different scanners: 3D-1000, 3D-250, KFBIO-400, Leica-450, Motic, and Teksqray-600p. This analysis highlights the variation in tumor coverage and domain imbalance among scanners

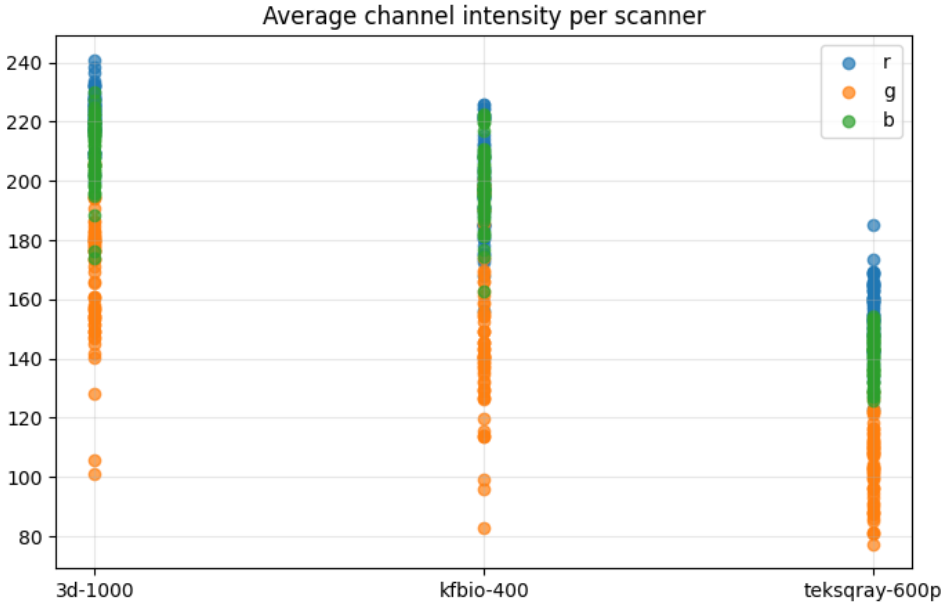

Figure A.3: Average channel intensity per scanner (training subset). Scatter plot showing the mean red, green, and blue channel values for images from the 3D-1000, KFBIO-400, and Teksqray-600p scanners. This visualization highlights color consistency and scanner-level differences in the training set

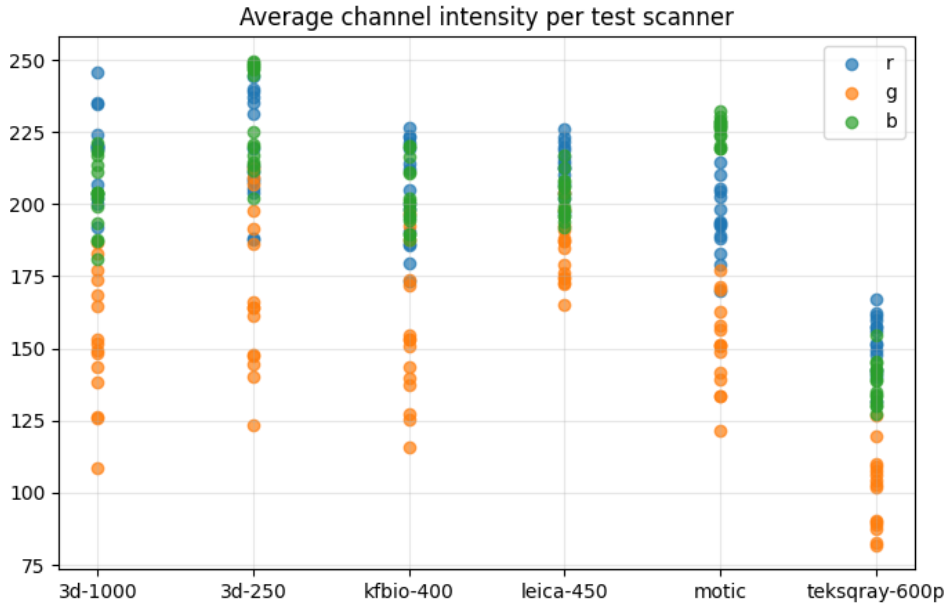

Figure A.4: Average channel intensity per test scanner (test subset). Scatter plot showing mean red, green, and blue channel values for all six domains: 3D-1000, 3D-250, KFBIO-400, Leica-450, Motic, and Teksqray-600p. This visualization illustrates cross-domain differences in stain appearance and illumination, highlighting the variability present in the test set