DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

# Comparing a New Implementation of SegFormer to UNet for Pathological Scan Segmentation

An Implementation and Evaluation Study within an Applied Digital Pathology AI Platform

Carlotta Sophia Hölzle

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

# Comparing a New Implementation of SegFormer to UNet for Pathological Scan Segmentation

An Implementation and Evaluation Study within an Applied Digital Pathology AI Platform

# Vergleich einer neuen Implementierung von SegFormer mit UNet für die Segmentierung pathologischer Scans

Eine Implementierungs- und Evaluierungsstudie innerhalb einer angewandten digitalen pathologischen KI-Plattform

| | |
|---|---|
| Author: | Carlotta Sophia Hölzle |
| Examiner: | Prof. Dr. Nassir Navab |
| Supervisor: | Vanessa Gonzalez, Kai Standvoss |
| Submission Date: | 15.08.2023 |

I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.08.2023 Carlotta Sophia Hölzle

# Abstract

Cancer diagnosis and research heavily rely on the accurate segmentation of histopathological images, with Hematoxylin and Eosin (H&E) staining being a prevalent technique that enhances tissue contrast, revealing vital cellular and structural information for cancer detection and treatment. While convolutional neural networks have been pivotal in automating H&E image segmentation, their limitations in capturing long-range relationships have motivated exploring of novel approaches for this task. In recent years, the introduction of Transformer-based models leveraging self-attention mechanisms has shown promising results across various machine learning domains. By embracing attention mechanisms that connect all elements irrespective of their positions, Transformers excel in capturing long-range dependencies and global context. By segmenting images into patches, the Transformer architecture has proven its effectiveness in image segmentation. Despite several hybrid approaches for H&E segmentation, a computationally efficient architecture devoid of complex pre- and post-processing is not yet available. This thesis investigates the potential of a specific Transformer-based model, characterized by improved computational efficiency and devoid of pre- or post-processing stages, to cater to the task of H&E segmentation. The research focuses on quantitatively and qualitatively evaluating the performance of this architecture, assessing its robustness and susceptibility to inter-annotator variability, and contrasting it with a state-of-the-art UNet implementation. A comprehensive evaluation is conducted using five different datasets obtained from a pathology company's database. The findings reveal that SegFormer, although not surpassing UNet in terms of quantitative performance, exhibits greater resilience to input perturbations. In qualitative assessment, SegFormer displays increased uncertainty compared to UNet in complex regions with intricate cases that warrant further expert scrutiny. This behavior could offer real-world advantages by pinpointing areas necessitating focused expert attention. Ultimately, empirical evidence suggests that SegFormer might not be the primary choice for general histopathological H&E segmentation due to its limited quantitative and qualitative performance. However, the architecture emerges as a potential alternative in scenarios demanding robustness and generalization. This study underscores the significance of automating histopathological segmentation and the need for continued research into computationally efficient Transformer-based methods to surmount the challenges of existing solutions.

***Keywords: Transformer, SegFormer, UNet, H&E segmentation, inter-annotator variability, robustness***

# Contents

# List of Tables

# List of Figures

# 1 Comparing a New Implementation of SegFormer to UNet for Pathological Scan Segmentation: An Implementation and Evaluation Study within an Applied Digital Pathology AI Platform

Segmentation in computer vision and imaging is the process of generically dividing the image into specific regions based on shared features [1], like color or texture similarities. The goal is to simplify the image for more straightforward analysis and comprehension. Semantic segmentation, or pixel-level classification [2], is a sub-category of image segmentation that assigns specific labels to each pixel based on their semantic meaning. It offers a higher level of detail, enabling the recognition of objects present and providing a comprehensive understanding of its content. Medical image segmentation extracts regions of interest from 3D or 2D medical images [3]. Non-semantic segmentation techniques identify general structures or regions of interest without labeling the whole image. For example, cell detection is a non-semantic segmentation technique essential to medicine for blood cell counting or cell culturing [4]. In contrast, semantic medical segmentation, like organ, rumor or vessel segmentation, is needed for precise anatomy [5] [6] [7] [8]. Semantic segmentation provides fine-grained segmentation and detailed labeling of anatomical and pathological structures needed for accurate diagnosis, treatment planning, and monitoring of various medical conditions [5–12]. In this thesis, only semantic segmentation is considered and therefore referred to as segmentation.

Pathologists diagnose severe diseases like cancer, identify abnormalities, and base treatment decisions on the visual analysis of tissue samples. As tissues are usually transparent and lack contrast under standard light microscopes, histopathological scans are stained with specific dyes to highlight cellular details, visualize tissue morphology, and make it possible to identify various cell types. Hematoxylin and eosin (H&E) staining is the most frequently used method because of its simplicity, cost-effectiveness, and versatility [13]. H&E staining involves applying two dyes, hematoxylin and eosin, to tissue sections to enhance the contrast and visualize cellular structures, as hematoxylin binds to nucleic acids (DNA) and stains cell nuclei blue-purple. In contrast, eosin binds to proteins and stains the cytoplasm and extracellular matrix pink. There are multiple challenges associated with the segmentation of H&E images:

1. **Local and Global Context:** H&E-stained slides reveal significant microscopic characteristics alongside extensive spatial relationships. These intricate spatial patterns are paramount for precisely identifying and segmenting diverse regions of interest, encompassing distinct cell types, tissue compositions, and pathological anomalies. Integrating both local and global contextual information is essential to achieve a comprehensive understanding of the entire image.

2. **Digital Slide Imperfections:** Variations in staining duration result in color and contrast disparities. Uneven staining, speckles, dust, and scratches introduce noise and artifacts in digitized H&E samples.

3. **Diverse Pathological Samples:** Tissues from different individuals exhibit distinct traits, often with overlapping structures compounding complexities.

4. **High Dimensionality:** The high-resolution and large image size of whole slide H&E scans demands splitting it into smaller, more manageable segments for training machine learning networks.

The process of manual H&E segmentation comes with significant labor and time requirements, alongside its inherent subjectivity stemming from variations in annotator style [14–16]. In response, machine learn-

ing, particularly deep learning methods utilizing neural networks, has emerged as a predominant strategy in medical image segmentation, aimed at mitigating these challenges through automated segmentation to enhance consistency and reproducibility [14, 17].

Automating precise H&E segmentation enables quantitative insights at scale and increased segmentation speed. This is important for streamlining pathology processes, enhancing diagnostic accuracy, aiding treatment planning, and advancing cancer research.

Convolutional neural networks (CNNs) have been providing state-of-the-art segmentation performance for H&E segmentation, with the UNet architecture being particularly prominent in this domain [14, 18–20]. The suitability of the UNet architecture for H&E segmentation has been extensively explored in both academic literature [21] [22] and practical applications within digital pathology [23] [24]. UNets have proven effective in addressing the inherent challenge of working with relatively small H&E patch dimensions, thereby avoiding susceptibility to challenge #4. These models excel in learning pixel correlations, adeptly handling the intricacies and variabilities of pathology scans at a fine scale [25].

CNNs employ downsampling to extract hierarchical features across scales, broaden the receptive field, and optimize image segmentation efficiency. However, this process leads to a loss of fine-grained details due to spatial resolution reduction in the feature maps. This limits the network's understanding of distant spatial relationships, which is critical for perceiving remote connections [26] [27].

Despite the integration of skip connections in the UNet architecture, certain limitations persist. Fixed downsampling operations and a confined receptive field in the final stages can still constrain a comprehensive understanding of complex long-range structures. Such structures bear considerable significance in pathology, as elucidated in challenge #1.

Furthermore, models derived from the UNet architecture frequently encounter difficulties in generalizing effectively across divers scanners and show declined robustness to input perturbation. Even minor deviations in image attributes can induce a decrement in performance, a phenomenon underscored by research studies [25,28]. This challenge is pronounced when dealing with H&E stained image segmentation; a topic discussed more extensively within challenge #2.

An additional architectural facet to contemplate is the pronounced inductive bias inherent in Convolutional Neural Network (CNN)-based architectures. This bias facilitates training on small datasets, a common scenario for H&E slides; however, it also introduces prior assumptions that might diverge from the intricacies of challenge #3.

Furthermore, the trajectory of performance enhancement has plateaued, leading to a discernible stagnation in the progression of models. Consequently, the imperative for pioneering approaches has surfaced as a response to surmounting this limitation.

These challenges and the need for H&E segmentation algorithm's comprehensive grasp of long-range dependencies among pathological structures and high robustness prerequisites underscores a potential for enhancing automated segmentation through incorporating extended contextual relationships and bolstering algorithm resilience.

Inspired by the success of Transformers in other domains, like natural language processing, sentiment analysis and text classification [29] [30] [31], researchers are exploring their potential in computer vision, including medical image segmentation. Transformer-based architectures are known for their attention mechanism, which enables them to selectively focus on relevant regions of the input. This capability allows the models to capture both local and long-range dependencies across the whole sequence and to learn contextual interdependencies among different elements of the input sequence [32] [33].

Using Transformer models to incorporate self-attention mechanisms into H&E segmentation presents an opportunity to add the ability to discern overarching patterns in tissue samples, which manifest when individual cells are closely clustered or widely dispersed. s

Potential challenges within utilizing Transformer-based architectures for H&E image segmentation comprise diverse aspects. These involve a notable computational complexity of $O(N^2)$ intrinsic to the self-attention calculation, a tendency towards an excessive focus on global context, reliance on stationary positional encoding to infuse spatial information into the self-attention mechanism, and a restricted inductive

bias intrinsic to the architectural structure, culminating in the demand for substantial training datasets and extended training durations. A comprehensive exploration of these challenges is introduced in chapter 3.

Additionally, to the sparse amount of training data for H&E segmentation task, annotator variability hinders the training process of deep learning methods. Agreement on annotations in the ground truth is essential for the models to make correct and accurate predictions. Variability within the annotations by one expert, when presented with the same data at different times or conditions, is in medicine referred to as intra-annotator variability. Automated segmentation can produce consistent and reproducible results, eliminating inter- and intra-variability associated with manual segmentation. However, for this, the susceptibility of different architectures towards inter-annotator variability plays a significant role, as high susceptibility translates to incorrect and inconsistent predictions, prevention from learning patterns, and ineffective generalization to unseen data.

Even though research on applying hybrid and pure Transformer-based models in medical image segmentation is growing, the recent approaches targeting H&E segmentation are all based on hybrid approaches, including high computational complexities and complex preliminary and subsequent data manipulations. When applied to digital pathology, the later limitations are not feasible. To contribute to advancing research in the field, a novel strategy was pursued for H&E segmentation, involving utilizing an efficient and minimally complex Transformer-based architecture known as SegFormer, as proposed by Xie [34]. This architecture addresses the computational complexity by reducing the self-attention calculations and retaining from a complex decoder and additionally solves the problem of fixed positional encoding. This study aims to assess the performance of SegFormer, compare it to a well-tested UNet model, and provide insights into its suitability for integrating into a continuous deployment pipeline. The quantitative and quantitative performance is compared using multiple different histopathological datasets. An additional dataset is used to evaluate the differences in robustness between the models. The susceptibility of the SegFormer and the UNet model towards inter-annotator variability is demonstrated on a final dataset. Additionally potential biases of the segmentation when training the model on a single annotator style are investigated based on a subset of the last dataset. The research concludes by considering whether the architecture is suitable for practical application in a pathological setting. Figure 1.1 visualizes the framework of this study.

To the best of the author's knowledge, no consensus in literature proves that Transformer models without extensive pre-training and data post-processing can outperform UNet-based approaches in H&E segmentation. To date, no instance exists of employing a computationally efficient and straightforward Transformer model within the context of pathological applications that eliminates the need for pre- or post-processing steps. The results obtained from this study will provide valuable insights into the potential of SegFormer as a practical solution for accurate and efficient semantic segmentation in pathology. In summary, the main contributions are as follows:

1. Integration of the SegFormer architecture to an applied pathological pipeline, enabling easy and continuous deployment for new H&E segmentation tasks.

2. Determining the optimal architecture and machine learning parameter initialization for best segmentation performance.

3. Comparison of quantitative and qualitative performance between SegFormer and a reference UNet implementation.

4. A thorough investigation into the impact of various input augmentation and perturbation techniques on segmentation performance.

5. Assessment of inter-annotator variability towards the segmentation performance and generalizability.

6. Consideration of the applicability of SegFormer for H&E segmentation in applied pathology, including economic implications and overall segmentation quality.

The thesis is organized as follows: Chapter 2 discusses the current research in H&E segmentation and discloses the resulting scientific questions. Chapter 3 details the methodology used and the necessary

adaptions to deploy the architecture for H&E segmentation. Chapter 4 overviews the different datasets used for training and evaluation. Chapter 5 describes the research design and experiments to answer the research questions. The last section, Chapter 6, concludes the findings of this work, discussing the impact on research and industry while giving insight into possible limitations. Chapter 7 provides a summary of the thesis together with possible future work.

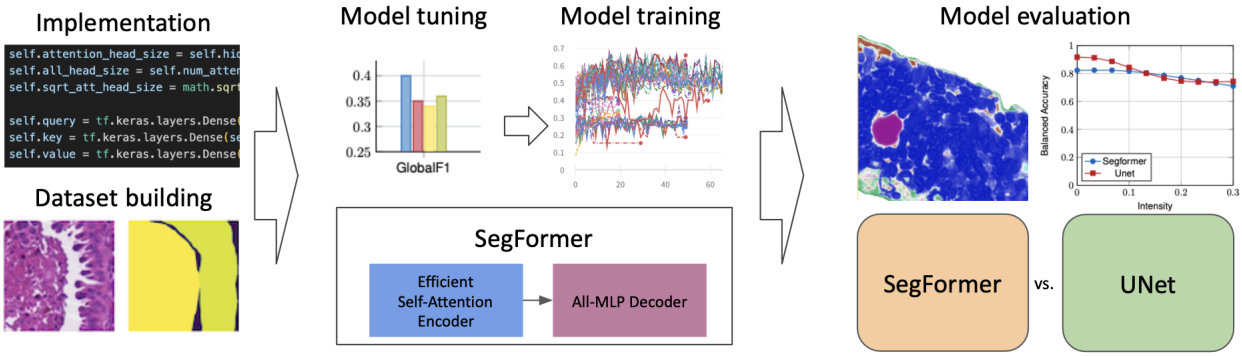



**Figure 1.1** Illustration of the Thesis' Framework

# 2 Related Work

## 2.1 Transformer models

Transformer models have gained much popularity in recent years and are one of the most powerful neural networks to date [35]. Originally designed to address natural language processing tasks [29], they utilize self-attention modules to establish meaningful relationships among them [36]. Architectures similar to this paradigm are categorized as pure Transformer models, while those fusing attention modules with convolutions or graphs are denoted as hybrid implementations, as defined by Andrade et al. [37].

## 2.2 Differences between Transformers and CNNs

The main differences between CNNs and Transformer models lie in their architectural structures and intended data types [38] [39]. Transformers perform well in multiple domains, like natural language processing, sentiment analysis, and object recognition [35], in contrast to CNNs, which are predominately deployed for computer vision tasks [40].

The architectures of the two techniques differ significantly: Convolutional Neural Networks use a hierarchical structure of convolutional layers for processing grid-like data [41], while Transformers rely on self-attention mechanisms for sequential data analysis [38]. The convolutional layers in CNNs employ fixed filters applied uniformly across the input grid, whereas the attention mechanisms in Transformers dynamically adjust weights for each input element based on sequence context [39] [38]. This contextual weight adaptation enables these models to handle variable-length sequential data, capture long-range relationships, and assign varying levels of importance to input elements during predictions [42] [40]. This flexibility combined with the ability to minimize reliance on predefined patterns and focus on long-range relationships, makes Transformers particularly effective in tasks where rigid patterns may not be present or applicable. However, this flexibility results in the necessity for long training times on large amount of data to learn patterns and relationships for the task at hand. Additionally the low inductive bias can result in difficulties in effectively capturing local spatial patterns and hierarchical structures. The pronounced inductive bias inherent in CNN-based models plays a crucial role in effectively capturing and utilizing local spatial patterns and hierarchical structures in the input data [43]. This bias is introduced through the design of local connectivity and shared weights in the convolutional layers [41]. Local connectivity is the principle of each neuron being connected to a small, localized region of the input data. This allows them to capture specific local patterns and extract relevant information from neighboring pixels, often containing correlated and contextually meaningful features. In CNNs, multiple neurons across different spatial locations use the same set of learnable parameters (weights). This enables efficient learning and recognition of common features across distinct image parts. Both assumptions enhanced data efficiency and the network's ability to generalize effectively to new data, even when the training dataset is limited [44]. However, the restricted receptive field at the top stage of the CNN-Encoder, attributed to spatial resolution reduction through pooling layers, limits the contextual information available in that layer and potentially hinders the inclusion of long-range dependencies [45]. Another difference lies in the size and memory utilization. While CNNs are known for their compact size and efficient memory utilization, Transformer models tend to have considerable model sizes and high memory requirements [46].

## 2.3 (Vision) Transformer for Medical Image Segmentation

Vision Transformers (ViTs) adapt the Transformer idea to image analysis tasks. They first divide the input image into non-overlapping patches and then subject them to parallel processing through the self-attention mechanism [47] [33]. In literature, ViTs are predominately trained with Cross Entropy loss for classification tasks, while Transformer models designed for medical segmentation tasks often combine different loss functions [48] [49].

Lately, hybrid ViT models have gained importance in the context of medical image segmentation tasks. Considerable research has emerged, with promising results on applying Transformer models to medical image segmentation in radiology, including X-ray, computed tomography, magnetic resonance imaging, and ultrasound [50] modalities. In 2021, J. Chen et al. introduced the TransUNet [51]. This hybrid methodology combines Transformer-encoded tokenized image patches and high-resolution CNN feature maps to extract global contexts and achieve precise localization, achieving results outperforming the state-of-the-art implementations for medical multi-organ segmentation in CT scans. For X-ray images, the Metal Segmentation Transformer by Fan et al. shows high robustness and generalization with a three times higher dice score than the compared UNet [52]. The hierarchical Swin Transformer Encoder used in Swin UNETR achieved top-performing results for 3D brain tumor semantic segmentation in MRI scans [53]. For breast ultrasound lesion segmentation, the CSwin-PNet Transformer outperforms state-of-the-art image segmentation methods [54].

However, transitioning this research to H&E segmentation is not straightforward. Disparities exist between radiology scans and pathological H&E slides. MRI and CT scans encompass extensive anatomical regions, while H&E slides furnish intricate cellular and tissue insights at a microscopic scale. Radiologic scans exhibit a similarly modest spatial resolution, capturing macroscopic organ contours [55]. Transformers manifest efficacy in segmenting structures, even amid limited spatial resolution, rendering them fitting for tasks emphasizing broader contextual attributes over meticulous spatial details. The global attention introduced in ViT models can be pivotal to accommodate variances in shape and proportions arising from varying viewpoints and patient orientations in MRI and CT scans [56]. In contrast, H&E slides possess a more uniform and localized spatial context, which is not naturally provided in Transformer-base models. Additionally, Transformers for image segmentation require a considerable amount of labeled training data [57], which is overall limited in medical fields, but more abundant in radiology than pathology, posing challenges for training Transformer models for H&E image analysis. Furthermore, H&E scans are of high spatial resolution, capturing small details at a microscopic level [58]. The substantial spatial resolution translates into a high number of input tokens to the self-attention steps, making it difficult to effectively incorporate all tokens during computation and introducing a high computational complexity [59]. This raises the question of whether ViT models are applicable to the specific field of H&E image analysis.

## 2.4 Deep Learning Models for H&E Analysis

### 2.4.1 CNN Models for H&E Analysis

Convolutional Neural Networks (CNNs) based approaches, particularly utilizing UNets as the foundational architecture, have demonstrated remarkable success in H&E segmentation [60]. These methods have attained state-of-the-art segmentation performance, even surpassing human-level segmentation accuracy [61].

### 2.4.2 ViTs for H&E Analysis

In H&E image analysis, there are several tasks where Transformer-based models are already applied. For instance, Ikromjanov et al. used a ViT model for accurately classifying the grading of prostate cancer [62]. Another study utilized an adapted ViT architecture to classify brain tumors and biomarker prediction in

histopathological scans [63]. The Swin-T transformer by Guo et al. is optimized for predicting key biomarkers in colorectal cancer from H&E-stained images [64]. For gastric histopathological image detection, the GasHis-Transformer by Chen et al. has proven advantages over traditional CNN-based methods [65].

However, the transition from H&E classification or detection tasks to H&E segmentation is more complex. Unlike classification and detection tasks, which primarily require recognizing global relationships and prominent features, segmentation necessitates a finer level of precision, aiming to recognize and label specific regions or entities within the image at the pixel level. The sparse research done in the field reflects the higher intricacy of deploying Transformer models to H&E segmentation. Most hybrid models trying to address the problem between modeling local features and global context awareness, essential for H&E segmentation, exhibit either an extension in the complexity dimension or a decrease in performance when confronted with down-scaled data [48] [49].

Research on using transformer models for segmentation tasks in pathology is scarce and requires further development, as mentioned in "Transformers in Medical Image Analysis: A Review" [50].

The recently introduced CellVit model [49] integrates an attention-based Encoder, skip connections, and three convolutional Decoders, followed by a intricate post-processing phase. This post-processing involves merging the feature maps from the three Decoders, calculating gradients of horizontal and vertical feature maps, employing an edge detection filter to detect salient regions, followed by applying a marker-controlled watershed algorithm for boundary delineation, concluding in a final segmentation achieved through majority class voting. The architecture showed promising results for detection and classification but lacked segmentation performance when confronted with down-scaled data, unable to outperform state-of-the-art implementations. The study by Zidan et al. deployed a cascaded Swin Transformer to the segmentation of histopathological microscopy images showing promising results [48]. The architecture comprises a Transformer Encoder, skip connection and a cascade upsampler. However, the model heavily relies on data pre-processing with cluster algorithms, contrastive self-supervised pre-training and initialization of the model with pre-trained weights to archive a performance that challenges state-of-the-art architectures. The hybrid MESTrans [66] integrates various components, including a UNet Encoder, multi-scale embedding block, multi-layer spatial attention Transformer, and multiple feature fusion modules, exhibiting exceptional generalization capabilities and superiority over other state-of-the-art methods at the cost of high computational demands and a complex structure.

## 2.5 Research Question

The existing research landscape reveals a noticeable deficiency in computationally efficient Transformer models with low complexity for histopathological segmentation.

The scientific question arising from this analysis can be formulated as follows: "How does a computationally efficient Transformer-based model, requiring no post- and pre-processing, perform within the context of H&E segmentation?". This inquiry gives rise to multiple sub-questions, formulated as follows:

1. Can a high-performing image segmentation transformer model be adapted for H&E slide segmentation in a histopathological setting?

2. Can this implementation quantitatively outperform the existing UNet reference model, specifically regarding the Global Macro F1 score and accuracy?

3. How does the robustness of the two models compare when subjected to different input perturbations?

4. What is the impact of the share of different losses in the Combined loss function on the Transformer's performance, and does it significantly influence the robustness of the SegFormer model?

5. How do the network sizes, particularly the number of parameters, and the computational complexities of SegFormer and UNet compare? What are the implications for a digital histopathological setting?

6. How does the segmentation compare qualitatively? Are there any significant differences visible during inference?
7. How is the SegFormer model affected by inter-annotator variability compared to the UNet?
8. Can the SegFormer generalize when trained on a singular annotator style, or does it exhibit more bias towards this style than UNet?

# 3 Methodology

## 3.1 UNet Architecture

The UNet architecture, a widely recognized CNN architecture, comprises an Encoder and Decoder path interconnected by skip connections [20]. These connections link corresponding Encoder and Decoder blocks on each level, preserving spatial information and gradient flow during back-propagation.

The Encoder path consists of down-sampling blocks, traditionally consisting of a 2D convolutional layer followed by max-pooling operations, reducing spatial resolution by half. This process is iterated to capture larger receptive fields and extract high-level features. The Decoder block focuses on restoring spatial resolution and combining information from different levels using transposed convolutions and skip connections to merge low-level and high-level features.

In the context of UNet, the term "backbone" pertains to adaptations in the Encoder architecture. The ResNet18 backbone, utilized in this study, closely resembles the traditional Encoder path. However, the Encoder blocks within the backbone employ no max pooling and two opposed to one convolutional layer linked by skip connections, enhancing gradient flow during training. Figure 3.1 illustrates the architecture of the UNet with a Resnet18 backbone employed in this study. Within the pathological company UNets with a ResNet backbone are the default convolutional neural network chosen for the segmentation task of histopathological scans. The UNet implementation utilized is openly available at [67].

## 3.2 Vision Transformer

The original Transformer architecture is designed for sequence-to-sequence tasks. In order to use the Transformer architecture for image segmentation, multiple adaptions have to be made. Image segmentation is a complex spatial task that requires the model to understand pixel-level spatial relationships and local details within an image. Each pixel is treated as a separate entity when representing an image in a digital format, resulting in a high-dimensional data structure. Vision Transformers (ViTs) are built on the Transformer idea but are specifically designed for image analysis tasks. They split the input image into a fixed-size of non-overlapping patches, also called tokens, which are then sequentially fed into the

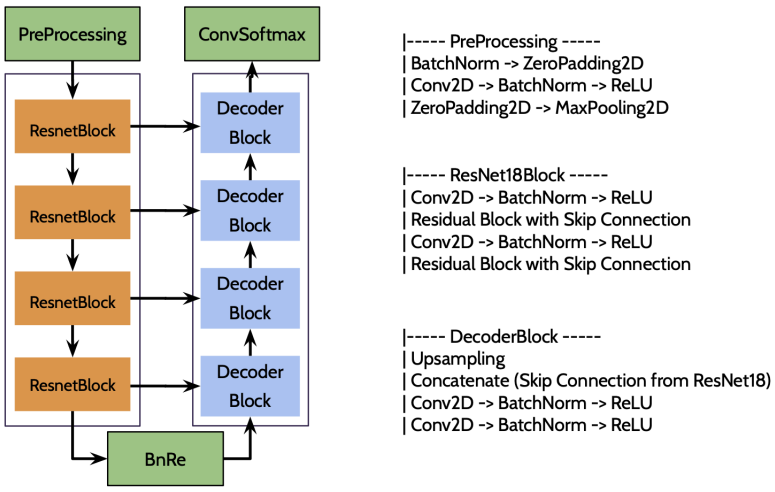



**Figure 3.1** UNet architecture

self-attention mechanism. This sequential operation on the input tokens prohibits any understanding of the relative or absolute position of elements within the original input image. However, understanding the spatial arrangement of image elements is crucial for grasping the overall context of the image. ViTs introduce positional encoding (PE) to inject spatial layout information before processing image patches as sequences to address this issue. PE adds a fixed set of vectors containing information about each token's absolute position to the self-attention input. Multi-head attention calculations within one self-attention step allow each head to attend to different patterns or features in the image at different resolutions and scales. ViTs can capture local contextual information through this adaption, empowering them to capture fine-grained details within the input. Employing a patch-based multi-head self-attention strategy, coupled with positional encoding, empowers ViTs to adeptly encompass contextual information across various scales within an image, spanning both local and global aspects. This methodology holds the potential to excel in segmentation tasks that necessitate meticulous object boundary demarcation and precise structure localization while also engendering the need for a comprehensive grasp of long-range relationships.

### 3.2.1 Attention Mechanism

During the attention process, each input token is transformed into three vectors: Query, Key, and Value. The Query vector represents the relevance of a token to the other units. The Key vector holds features from every token in the input sequence to which the current token will be compared. The Value vector carries the inherent representation and features of the current unit. The attention probability is computed using equation 3.1. First, the attention scores are obtained through the dot product between the Query and Key vector of each token, signifying the importance of each token concerning all the others in the input sequence. To ensure balanced contributions of each token, the attention scores are normalized by dividing them by the square root of the embedding dimension, preventing any single token from dominating the mechanism. The embedding dimension refers to the size or dimensionality of the vectors and determines the number of features or dimensions used to encode information about each token. A softmax layer is then applied to convert the raw scores into a probability distribution, determining the relative importance of each token concerning to others. In the second step, the context representation is derived through equation 3.2. This representation encompasses relevant information from all tokens, and emphasizes the most relevant aspects and relationships in the input sequence, facilitating context-aware processing. The attention probabilities are then multiplied with the Value vector to scale the importance of each token's associated information according to its attention weight and obtain a context representation. This calculation is performed in parallel across all attention heads; all contextual representations are then concatenated before being passed through a linear transformation to produce the final output of the self-attention layer.

The attention process is formulated as follows:

$$attention\_probability = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \tag{3.1}$$

$$context = attention\_probability \cdot V \tag{3.2}$$

### 3.2.2 Potential of ViTs for H&E sSegmentation

For H&E segmentation, the prospect of including long-range relationship understanding through a self-attention mechanism is essential for a variety of reasons:

1. Tumor Segmentation: Tumors can vary significantly in size and shape. Segmentation of tumors requires understanding the full extent of the tumor region, including the central mass and any satellite lesions. Long-range relationships allow the model to capture the entire tumor region and differentiate it from surrounding healthy tissues.

2. Cell Nuclei Segmentation: In histopathology, analyzing cell nuclei is crucial for diagnosing cancer and other diseases. Nuclei segmentation involves distinguishing individual nuclei, which can be

densely packed. Long-range relationships help correctly identify nuclei clusters and maintain separation between adjacent nuclei.

3. Glomeruli Detection: Distinguishing specific cell types is crucial for assessing the health of organs, like detecting glomeruli to assess kidney health in renal pathology. To identify glomeruli in biopsy images, understanding the overall glomerular structure, which may span large areas, is necessary. Long-range relationships aid in capturing the full extent of glomeruli and their connections.

4. Tissue Anomaly Detection: In anomaly detection, finding rare and abnormal tissue regions is essential for identifying diseases or abnormalities. Long-range relationships enable models to capture global context and differentiate between normal and abnormal tissue patterns.

Additionally, the improved robustness and generalisability related to the self-attention mechanism promise high potential for H&E segmentation tasks which are often paired with blurry and noise input data.

### 3.2.3 Challenges with ViTs

While the ViT architecture promises an innovative approach to advance H&E segmentation, the following challenges have to be considered:

1. **Quadratic Time Complexity:** The self-attention calculation exhibits $O(N^2)$ complexity due to the dot product interactions among all input tokens. The high dimensionality of H&E slides would introduce resource-intensive computation and inefficient inference speed.

2. **High Data requirements due to low inductive bias:** Transformer networks do not include significant prior assumptions within their architecture, necessitating substantial amounts of training data and time to effectively grasp complex relationships. Conversely, architectures like CNN-based UNet exhibit high data efficiency despite limited training data due to the inductive bias built into the architecture. In the context of digital pathology, the considerable demand for training data presents a substantial challenge. Labeled H&E training data often remains scarce due to data privacy regulations and the intricate nature of annotating accurate ground truths. This issue is accentuated by the time-consuming process of annotating tissue samples.

3. **Global Dependency Overemphasis:** Despite the multi-head attention mechanism, ViTs tend to rely too much on global dependencies and struggle to capture thin boarders and subtle structures, which is crucial for precise segmentation for H&E slides.

4. **Absence of Local Feature Buffering:** Capturing and preserving local features poses a challenge without skip connections. This may further limit the model's ability to capture intricate details in the image.

5. **Fixed Positional Encoding:** The immutability of positional encoding in ViTs limits adaptability to diverse input shapes without compromising segmentation performance.

### 3.2.4 SegFormer

The Transformer-based semantic segmentation model SegFormer by Xie et al. [34] proved to circumvent the identified limitations of traditional ViT models. The following reasons motivated the implementation of the architecture for H&E segmentation:

1. **High Performance on multiple Image Segmentation Datasets:** SegFormer set in 2021 state-of-the-art performance in terms of efficiency, accuracy, and robustness on ADE20K, Cityscapes, and COCO-Stuff [34].

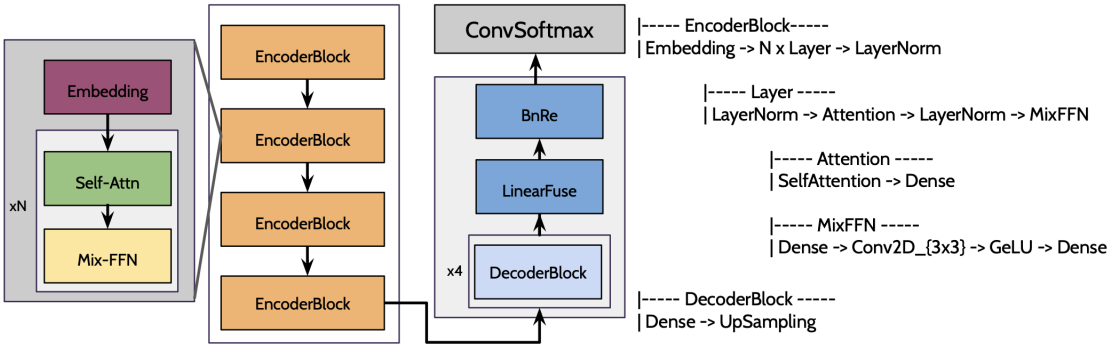



**Figure 3.2** Architecture SegFormer

2. **No Positional Encoding:** The authors recognized the problem with fixed positional encoding in traditional ViTs and designed a positional-encoding-free hierarchical Transformer Encoder, making it possible to adapt to arbitrary test resolutions without impacting the performance.

3. **Introduction of Multiscale Features:** The hierarchical part of the Encoder addresses the limitation of single-resolution feature maps in traditional ViTs. The hierarchical structure within the Encoder block allows for capturing contextual information at multiple scales.

4. **Convolutional Reduction of Self-Attention Mechanism:** By applying convolutional reduction to the Value and Key vectors, the computational complexity associated with the self-attention calculation is mitigated, concurrently introducing the capability to capture local patterns and relationships within a defined receptive field.

5. **Reduced Number of Parameters:** The All-MLP decoder is implemented without complex modules, thus significantly reducing the computational effort and presenting the possibility to deploy the architecture into an applied digital pathological setting.

6. **Heightened Robustness Due To Attention Module:** The inherent potential of enhanced robustness arising from the utilization of the attention module suggests that the model can effectively generalize in real-world contexts, even when facing significant input perturbations in the context of H&E segmentation tasks.

The SegFormer demonstrated promising results in semantic segmentation on the CityScape dataset. However, its applicability to H&E slide segmentation requires investigation. First, H&E patch sizes are commonly smaller due to their high resolution. The reduction is essential to put the computational complexity of the high number of pixels within acceptable limits for an applied pathological setting. However, the SegFormer demonstrated its best segmentation performance on image patches of sizes exceeding 512x512, compared to the patch sizes used for training of 224x224 to reduce computational complexity. Second, the high significance of fine-grained and local features in the H&E slides could still pose a challenge as the SegFormer does not incorporate any skip connections to preserve local features for a specific resolution.

### Difference to UNet

Figure 3.2 illustrates the framework of the proposed SegFormer methodology. The architecture adheres to an Encoder-Decoder structure similar to UNet's design. The Encoder path design follows the principles of ResNet; the channel dimensions increase while the spatial resolution shrinks with the layer goes deeper. However, notable distinctions emerge in the configuration of the Encoder blocks. While the Encoder path of the chosen UNet model employs a ResNet18 implementation consisting of paired convolutional layers, batch normalization, and rectified linear units for each Encoder block, SegFormer's Encoder block follows

a divergent approach. It initiates with an embedding layer that transforms the input into a sequence of overlaying patches. Subsequently, multiple iterations of self-attention mechanisms and feed-forward neural network layers follow, incorporating the multi-head approach introduced by Vaswani et al. [68].

In UNet, the preservation of spatial details between Encoder and Decoder blocks is assisted through skip connections at each layer. This practice is absent in SegFormer's architecture. In the Decoder path, UNet incorporates an equivalent number of Decoder blocks as Encoder blocks. Each Decoder block fuses feature maps from the lower layer with those obtained through the skip connection. Conversely, the SegFormer Decoder concatenates feature maps from all Encoder blocks simultaneously.

Both architectures end with a convolution softmax layer to translate acquired features into pixel-wise predictions, yielding the final segmentation mask. In the following the proposed architecture is described in detail. The modules described

#### Hierarchical Transformer Encoder

The Encoder is based on the ViT Encoder but tailored to optimize image semantic segmentation by reducing the computational complexity and removing the positional encoding prevalent in traditional ViT. The initial paper introducing the SegFormer architecture designed a series of Encoders with identical architecture but different sizes. The Encoder consists of four Encoder blocks, each block consisting of one overlapping patch embedding layer and multiple so-called SegFormer-layers. The number of layers depends on the SegFormer size and introduces a hierarchical feature representation, unlike the traditional ViT Encoder, which only generates a single-resolution feature map. Unlike the UNet architecture, the Encoder does not incorporate convolutional layers but combines multi-head self-attention layers and feed-forward networks in the SegFormer-layer. The following section precisely explains the individual parts of the Encoder.

**Overlap Patch Embedding Layer** Traditional ViTs divide the input image into non-overlapping patches and compute attention weights between them to capture their relationships. The underlying assumption is that local continuity can be preserved through PE without overlap between the patches. However, in practice, this approach may not effectively capture local relationships around the patch boundaries. Furthermore, this process restricts ViTs from generating feature maps with a single resolution, limiting their capacity to comprehend information at different scales and hierarchies, potentially impacting their performance on tasks that require understanding detailed and complex image structures. The overlapping patch merging technique in the SegFormer architecture splits the input into overlapping patches to form hierarchical feature maps. It processes them individually to enable the model to capture information at different scales.

The process of overlapped patch embedding can be understood as such: Given an input image of resolution H x W x 3, patch embedding is done to get a hierarchical feature map $F_i$ with a resolution of $\frac{H}{2^{(i+1)}} \times \frac{W}{2^{(i+1)}} \times C_i$ where $i$ is the index of the encoder block and $C_{i+1}$ is larger than $C_i$, indicating that the model progressively learns higher-level features at each stage, similarly to UNet. To enable overlapping patch merging, the produced features must maintain the same size as those obtained in the non-overlapping process. his is done by introducing three additional variables $K$, $S$, and $P$. Here, $K$ represents the patch size, $S$ represents the stride between adjacent patches, and $P$ represents the padding size, equal to half the patch size($P = K \div 2$). The default values for $K$, $S$, and $P$ are typically set as $K = [7, 3, 3, 3]$, $S = [4, 2, 2, 2]$, and $P = [\frac{P_i}{2}, \frac{P_{i+1}}{2}, \frac{P_{i+2}}{2}, \frac{P_{i+3}}{2}]$. Keeping the same feature size ensures that the overlapping patches align correctly, facilitating a smooth and accurate merging process during the hierarchical feature representation generation. This hierarchical approach allows the model to capture both fine-grained details and coarse contextual information, generating UNet-like multi-level feature maps.

**SegFormer Layer** The number of layers within one Encoder block corresponds to the $Depth$ specified in the initialization, defaulting to two layers per Encoder block. Fundamentally, each layer comprises a multi-head self-attention layer and a mixed Feed Forward Neuralnet (MixFFN). Layer Normalization is applied before the attention block and again before the MixFFN to standardize and stabilize the activation within

the network during training. This normalization process is performed independently for each batch sample, ensuring numerical stability and avoiding division by zero. The epsilon, added to avoid the zero-division problem, defaults to 1e-05 for every normalization layer within the architecture but can be changed in the configuration.

**Efficient Self-Attention** A dense layer with num_attention_heads * attention_head_size neurons is used for the Query, Key, and Value vectors to introduce multi-head attention. This allows the model to simultaneously attend to different aspects of the input and capture complex relationships. Each head within the attention can focus on different parts of the input, enhancing the model's ability to understand diverse patterns and dependencies in the data.

The self-attention mechanism in SegFormer employs a sequence reduction process through a convolution step to avoid the high computational complexity faced by traditional multi-head self-attention processes of $O(N^2)$.

Similar to UNet SegFormer uses a convolutional layer to effectively reduce the spatial resolution of the input tensor before the self-attention operation. Using convolutional layers for sequence reduction is a practical choice to handle the large spatial dimensions of the H&E images efficiently, as they have proven effective in capturing local patterns and spatial hierarchies. This down-sampling operation results in an output tensor with fewer spatial dimensions and an increased depth (number of channels), effectively representing a lower-dimensional representation of the input data. The convolutional operation is executed with kernel and stride size based on a factor $R_i$ specific for each Encoder layer. In the attention process, the Key and Value vector operate on the input's down-sampled version, while the Query vector is computed on the original input.

$$K = Dense(filter = hidden\_size)(Conv_{RxR}(input))$$

Given an input tensor of shape $N \cdot C$ with $N = H \cdot W$, the attention probability is estimated as:

$$Attention(Q, K, V) = Softmax(\frac{QK^T_{reduced}}{\sqrt{d_{head}}})V_{reduced}$$

The attention mechanism calculates attention scores between all pairs of elements in the sequence. Due to the reduction in $K$ and $V$, the attention operation within each attention head $i$ has a reduced complexity of $O(\frac{N^2}{R[i]})$. As a consequence, the computational complexity of the full self-attention mechanism is reduced to $O(\frac{N^2}{R})$, where $R$ is the average reduction ratio across all attention heads. The final attention output is fed through another Dense Layer to adjust the dimensions for the next Encoder block. During training a drop-out layer is deployed to prevent overfitting.

The observed reduction in computational complexity is achieved by sacrificing spatial resolution. This enables more efficient computation while preserving relevant context information for the attention mechanism. However, it is crucial to acknowledge that this trade-off could pose challenges in H&E segmentation tasks, as maintaining spatial resolution on lower layers is critical for capturing fine-grained information commonly present in histopathological scans.

**Drop Path Module** During the training process the attention outputs are subject to a SegFormer-specific drop-path operation if the $drop_path$ parameter is set to a value greater than zero. In contrast to the traditional TensorFlow DropOutLayer, where individual elements are dropped, this module stochastically drops entire layers during training. By generating a random tensor based on a given probability and then element-wise scaling the input tensor with this random tensor, certain activations are randomly dropped out, mimicking the effect of skipping specific layers during training. This technique aids in improving the model's generalization capabilities and contributes to regularization during the training of the model [69].

$$keep\_prob = 1 - drop\_path$$

$$random\_tensor = keep\_prob + tf.random.uniform(tf.shape(x), 0, 1)$$

$$random\_tensor = \lfloor tf.cast(random\_tensor, x.dtype \rfloor$$
$$output = \frac{x}{keep\_prob} \cdot random\_tensor$$

**MixFFN** Vision Transformers (ViTs) extensively employ Positional Encoding (PE) to infuse spatial location information into input tokens. However, the inherent limitation of fixed PE resolution necessitates interpolation when the test resolution differs from the training resolution. This interpolation poses disadvantages due to the potential loss of fine-grained spatial details, impact on the attention mechanism, and diminished robustness. In contrast, the authors of the SegFormer architecture argue that conventional positional encoding is dispensable for image semantic segmentation. Instead, they introduce a "Mix Feed-Forward Neural Network" (MixFFN) that exploits the impact of zero padding to encode location information. This is realized by integrating a 3x3 convolution layer with a multi-layer perceptron (MLP) to yield location-aware features. The MixFFN can be represented with the subsequent equation:

$$x_{out} = MLP(GELU(Conv_{3\times3}(MLP(x_{in})))) + x_{in}$$

This step captures local and non-local attention information by combining MLP and a convolution layer. The depthwise convolution operation specializes in extracting local patterns and features, ensuring the retention of fine-grained spatial information. Meanwhile, the MLP layers enable the extraction of higher-level abstract representations encompassing non-local relationships and dependencies across the input. Herefore, the MixFFN's design enables it to simultaneously attend to both local and non-local aspects of the data, contributing to a comprehensive understanding of the input's contextual information. The inclusion of drop-out layers throughout the operation aids regularization and prevents overfitting. The activation function is applied post-depthwise convolution, introducing non-linearity into the MixFFN, and enhancing its ability to discern complex relationships within the data. This orchestration of different deep learning modules endows the MixFFN with the capacity to understand complex spatial relationships and local semantic dependencies, making it more effective for semantic segmentation tasks than traditional positional encoding.

### Lightweight All-MLP Decoder

The effective receptive field relates to the portion of the input data that influences the output of a specific neuron within the network. In the case of Transformers, the self-attention mechanism allows each token to attend to all other tokens in the sequence. This process enables the model to carefully evaluate and assess information from distant regions of the input data, resulting in a large effective receptive field. Consequently, regions in the input data that significantly impact the final predictions are more effectively integrated into the network's activations. In the original literature, the proponents of the SegFormer architecture assert that the Decoder can be simplified by solely comprising Multi-Layer Perceptrons (MLPs), capitalizing on the comprehensive information already extracted by the Encoder. The Decoder step of the architecture is constructed as follows: Initially, each feature map from the Encoder path is subjected to processing via an MLP layer. This procedure harmonizes the channel dimensions across these feature maps. Subsequently, the feature maps undergo an upsampling operation, enlarging their dimensions to a quarter of their original input sequence's extent. The combined representation is then directed through a batch normalization and an activation layer, enhancing the network's capacity to learn intricate patterns and relationships. During the training phase, an additional drop-out layer is introduced. The outcome is then directed into a linear MLP-classifier, wherein the number of output channels aligns with the count of labels required for segmentation purposes. While a traditional MLP typically comprises multiple hidden layers with non-linear activation functions, this classifier transforms a linear transformation without additional non-linearities. This design choice aims to reduce the complexity of the model while still providing the necessary mapping from features to label predictions.

The resultant logits, representing the network's output scores, are transformed into probabilities by utilizing of a Softmax layer to assign probabilities to each label category for effective segmentation.

$$F_i = UpSample(MLP(F_i))\forall i$$
$$F = MLP(Concat(F_i))\forall i$$
$$Output = Softmax(MLP(RELU(BatchNorm(F))))$$

The Lightweight MLP Decoder has a reduced parameter count compared to traditional Transformer or CNN-based decoders, mitigating the computational overhead even further.

## 3.3 Implementation

The SegFormer architecture was chosen based on its novelty, its outstanding performance on CityScape Image segmentation, and the available huggingface TensorFlow implementation [70]. A problem with the open-source huggingface library is its granularity and the dependency of modules on huggingface internal classes and functions. These dependencies and the complexity of the whole library prohibit downloading and training the models in an air-gaped environment, like most medical training pipelines. For this thesis, it was not possible to introduce these open dependencies or an open-source library into the training pipeline due to security concerns. Therefore all inheritance from huggingface modules and references to huggingface-internal functions had to be removed, and necessary substitutions had to be rewritten to load the model into the company's training pipeline.

The SegFormer model in the huggingface implementation inherits the base class of the library. This class handles storing configuration, loading, downloading, and saving of the model. However, this dependency had to be removed to avoid loading the entire library into the internal pipeline. This replacement of the huggingface base class with the fundamental $tf.keras.Model$ class [71] resulted in the loss of some supporting functions. However, this modification rendered the model more universally applicable and self-contained, aligning with the guidelines of the internal code standards. The parameters for the original implementation are provided through a specific configuration class named SegFormerConfig, which inherits from the huggingface internal configuration framework. This dependency had to be removed for two reasons. To integrate SegFormer into the company's pipeline, the ability to transfer parameters from a Hydra configuration file [72] to the model is essential. Therefore, relying on a huggingface configuration file that combines all parameters into a single class was not feasible. Secondly, the configuration file by huggingface inherits from $PushToHubMixin$ class which introduces an interdependence to the open-source library. This reliance presents a notable security concern due to the possibility of exposing the internal system structure to the web, which is deemed infeasible when considering sensitive medical data. All modules in the SegFormer implementation rely on the SegFormerConfig class for initialization. Hence all models needed to be adapted. The solution was to replace the class with a Python dictionary structure and build this dictionary in the model's constructor. Additionally the SegFormer Layer and Encoder module in their original implementation provided the possibility to return either a standard tuple or a huggingface output class. This class consolidates all states collected during the forward pass of the input through the model. For the H&E segmentation task, it suffices to provide the essential states and parameters for the subsequent layers as tuples, eliminating the need for utilizing this particular class. Removing this dependency entailed modifying all relevant return functions and internal dependencies associated with this class.

The internal pipeline expects a build model upon invocation. Therefore, a supplementary method was formulated to initialize the SegFormer model using a TensorFlow Input and subsequently returns a model ready for execution. With this workaround various methods within the internal pipeline, such as data compatibility checks, device compatibility checks, and wrapping the keras model as a network, would avoid failure. Initially implemented in the eliminated base class, the train step and test step functions have become redundant within the internal process and thus require no reconfiguration. Within the company's framework, training procedures are managed by a TensorFlow module called "Trainer" which offers comprehensive functionalities for neural network optimization and evaluation. This module configures the loss function and optimizer, affording the flexibility to select from various TensorFlow options for these components. During the train step the Trainer computes gradients relative to the trainable variables in the given

network and employs the provided optimizer to apply these gradients to the network. For both train and validation step the necessary information, such as model predictions, loss values, and gradients, are organized in a data dictionary and returned by the Trainer. This in turn is used to emit the loss and calculate the evaluation metrics. The original code uses the train and eval step of the base class but computes the loss function and does the back-propagation within the SegFormer implementation. Because the internal trainer module is tasked with these responsibilities, both functionalities were removed from the code. In the company's framework, training processes are overseen by a TensorFlow module, referred to as "Trainer", which provides a comprehensive range of functionalities for optimizing and evaluating neural networks. This module configures both the loss function and optimizer, offering the flexibility to choose from various TensorFlow options for these components. In the training step, the Trainer computes gradients relative to the trainable variables in the given network and employs the provided optimizer to apply these gradients. The Trainer organizes and returns essential information, such as model predictions, loss values, and gradients, as a data dictionary for the training and validation steps. This is used to emit the loss and calculate evaluation metrics. The original code initially used the train and evaluation step function of the base class, with the computation of the loss function and back-propagation being integrated into the SegFormer implementation. As these tasks are now entrusted to the internal Trainer module, both functionalities had to be eliminated from the code. Initially, an attempt was made to incorporate a method for importing pre-trained weights into the SegFormer architecture. However, this approach was eventually abandoned because the existing pre-trained weights were exclusively compatible with the huggingface model and could not be seamlessly transferred to the now generic implementation.

Following the elimination of all huggingface dependencies, the internal structure of the layers had to be adapted. While input data is internally loaded in the channel last format (batch_size, in_height, in_width, in_channels) (NHWC), the huggingface implementation expects the tensors in channel first format (batch_size, in_channels, in_height, in_width) (NCHW). This necessitated the modification of each call function within every module, requiring the adaptation of all computations performed on the input tensors to accommodate a permutation of the channels. After evaluating the customized implementation using simulated tensors and a simplified training process, it became necessary to modify the internal wrapper function. This function is responsible for loading the appropriate network as specified in the Hydra configuration file [72]. The wrapper function takes the parameters the machine learning engineer specified and passes them to the model. For this purpose, the dictionary in the constructor of the SegFormer was leveraged. By composing an extra helper function to the SegFormer codebase, it becomes feasible to assign parameters manually, select one of the six predefined SegFormer types based on the original paper, or revert to default values when no specific ones are given. A function that automatically chooses the correct parameters was implemented to access the values related to the predefined network types. Table 3.1 shows the different network types with their corresponding parameters. For the task of H&E segmentation only the smallest three network types were used. From now on the network types are referred to through the variant name. The hugging face loss function works on the model's output logits, not the normalized output probabilities like standard TensorFlow loss functions applied within the training applied pipeline. An additional Softmax layer had to be added as the last layer to the SegFormer architecture to return a probability segmentation mask as model output. This Softmax layer was introduced as a distinct layer, separate from the Decoder, to allow the wrapper function to modify the data type of the final Softmax layer to float32, regardless of the data type set for the input tensor. This is required for numerical stability throughout the entire training and evaluation process. The final adaption introduced the possibility to relax the input shape during the model training process. This ensures that the input shape is not considered during training which allows input shapes to differ from the initial training shape in inference or fine-tuning tasks.

## 3.4 Evaluation Metric

To test whether this adapted implementation of the SegFormer architecture can provide better segmentation performance than UNet, a comprehensive assessment comprising both quantitative and qualitative

| Variant | Layers per Encoder Block | Encoder Block Dimension | Decoder Dimension | Nr. Parameters |
|---|---|---|---|---|
| b0 | [2, 2, 2, 2] | [32, 64, 160, 256] | 256 | 3.7 million |
| b1 | [2, 2, 2, 2] | [64, 128, 320, 512] | 256 | 14.0 million |
| b2 | [3, 4, 6, 3] | [64, 128, 320, 512] | 768 | 25.4 million |
| b3 | [3, 4, 18, 3] | [64, 128, 320, 512] | 768 | 45.2 million |
| b4 | [3, 8, 27, 3] | [64, 128, 320, 512] | 768 | 62.6 million |
| b5 | [3, 6, 40, 3] | [64, 128, 320, 512] | 768 | 82.0 million |

**Table 3.1** Parameters for SegFormer Network Variants

analyses was conducted. For quantitative assessment primarily the Global Macro F1 score was used. The Global Macro F1 score is calculated as the unweighted average of all class F1 scores indicating the model's precision and recall performance across all classes. "Global" refers to the F1 score being computed on all state updates instead of a per-sample or per-batch computation. This approach allows for a fair comparison of the models' performance if there is a heavy imbalance across batches. "Macro" denotes that the final score is computed across all classes, resulting in equal consideration of all class contributions toward the overall performance. This helps when dealing with high class imbalances, traditionally present in H&E datasets, by avoiding bias towards the dominant class. The average "Micro" was not considered as an evaluation metric because of the high class imbalances and uneven distribution of the labels in the ground truth. The F1 score for individual classes is computed using the formula:

$$F1 = \frac{TP}{TP + 0.5 * (FP + FN)}$$

Where TP represents the true positives, FP denotes false positives, and FN corresponds to false negatives. The F1 score is a harmonic mean of precision and recall, reflecting the balance between false positives and false negatives for each class. TP, FP, and FN are collected in the evaluation step on a validation share of the entire dataset, which is not seen during the training step. The inclusion of sample weights per class is a feature integrated into the calculation of the Macro F1 score. This functionality enables assigning higher importance to specific classes of interest, which addresses the limitation of the Macro F1 score's inability to reflect the significance of particular classes. Additionally to the Macro F1 score, the balanced accuracy was measured to assess the robustness of the two architectures when confronted with input augmentation techniques and H&E specific perturbations. The balanced accuracy averages sensitivity and specificity across all classes, providing a single scalar value representing the model's ability to discriminate between all classes. This metric measures the model's overall performance in terms of true positive rates (recall) and true negative rates (specificity), ensuring that the dominating class does not skew the model. The balanced accuracy metric is introduced in the robustness analysis to tackle the problem of class imbalance caused by introduced perturbations. The additional metric allows for a more complete and unbiased understanding of the overall performance of both models and a more informed evaluation of the results. Hydra [72] was employed as a scheduling tool to ensure the reliability and validity of the evaluation metric, enabling consistent and systematic evaluation of both models. The model's predictions were visualized for qualitative assessment on full H&E slides, as illustrated in Figure 3.3.

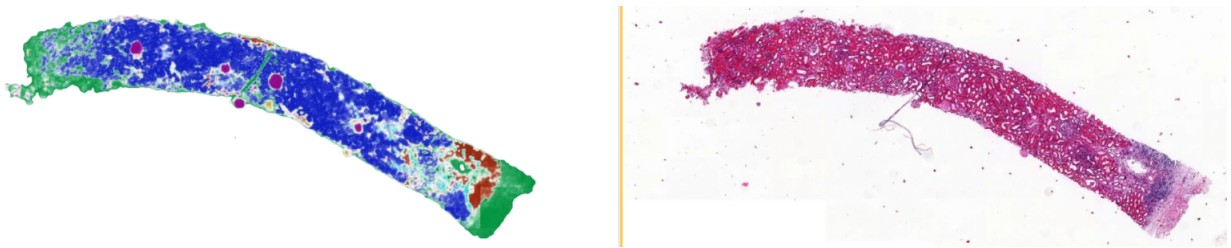


**Figure 3.3** Full H&E Slide Segmentation

For real-world applications the segmentation of complete whole slide images is essential. To accomplish this, the whole H&E slide is divided into smaller patches, exceeding the size of the training patches to address inference speed. Predictions are made on these individual patches and subsequently aggregated and merged to form a segmentation for the entire image. An alpha value is used to scale the transparency of the model on a prediction with its certainty. This allows the visualization of how sure the model is of its own segmentation. Personal and expert judgments are then consulted to evaluate the graphic representation of the predictions. Furthermore, the visualization tool ImFusion [73] is employed to facilitate an in-depth analysis of the similarities and differences between the two segmentation masks. The combination of quantitative and qualitative evaluations provides a comprehensive investigation into the research question, allowing for a thorough understanding of whether SegFormer is a competitive alternative to UNet for H&E segmentation.

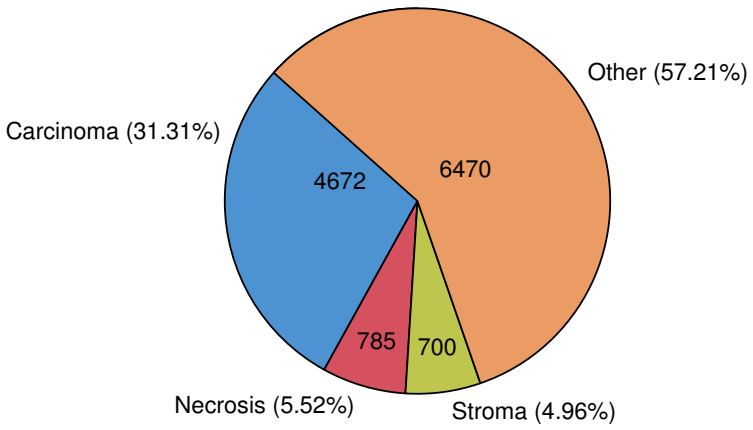
# 4 Material

## 4.1 Dataset Description

Dataset building in H&E pathology involves the process of curating and assembling datasets of annotated histopathological images. Annotator projects play a crucial role in dataset building as they provide the annotations needed to create ground truth labels for the images. Annotator projects refer to efforts where one or multiple annotators, pathologists, or medical experts annotate and label specific regions or features of interest in a fixed set of histopathological images stained with Hematoxylin and Eosin (H&E). The datasets used in this thesis are based on four different projects. Each project includes different H&E whole slide images, or the annotations on these slides are done by different annotators. It was essential to utilize as many different annotator projects, including different organs and annotator styles, as possible to acquire a comprehensive understanding of how the SegFormer performs on H&E slide segmentation. The following section describes the dataset building from the different annotator projects, including displayed organs, size of the final dataset, annotation criteria, and eventual dataset limitations.

### 4.1.1 Dataset 1

This dataset is used to evaluate which machine learning and architectural parameters yield the best segmentation performance for the SegFormer architecture. The annotation project includes 11135 annotations for 19 categories by one pathologist. The dataset was built with a patch overlap of 25% resulting in 85182 224x224 patches. For training the 19 categories were fused into 4 classes, Carcinoma, Necrosis, Stroma, and Other. The data does show a high percentage of annotation for the "Other" class. However, because



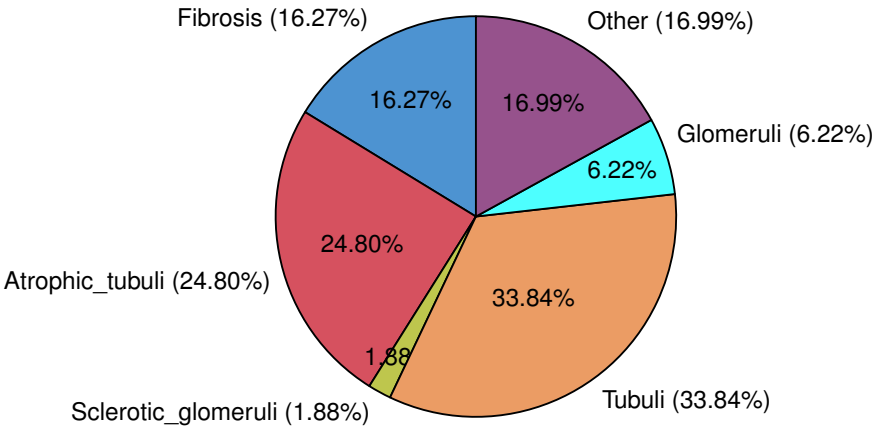

annotations from 14 different categories are included within this category, a class imbalance towards this class can be disregarded.

### 4.1.2 Dataset 2

This dataset was built of the same whole slide images as dataset 1, therefore including the same base annotations. However, this dataset was built without any patch overlap, and the patches extracted were cropped into 512x512 dimensions rather than 224x224. Consequently, the dataset is comprised of a total of 5540 patches.

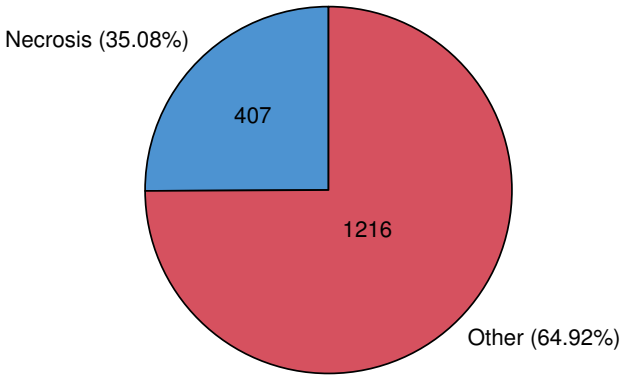
### 4.1.3 Dataset 3

A different dataset from another annotation project was used to test the robustness of the SegFormer architecture compared to UNet. The annotation project includes 45631 annotations for 22 categories by two annotators on kidney H&E slides. The dataset was built with a 50% overlap, resulting in 127300 224x224 patches. For training the 22 categories were fused into 6 classes.




### 4.1.4 Dataset 4

A dataset that aims for binary model segmentation was used to test whether the SegFormer model performs differently on a segmentation task for fewer classes. This dataset was in an active annotation process during the experimental process. This necessitated communication with annotators, developing an annotation strategy, and multiple rebuilding of the dataset to include new annotations. In the beginning annotations were gathered for 3 categories but later extended to 4, resulting in a more comprehensive dataset and more annotations the model could learn from. A central challenge were incorrect "Microns Per Pixel" (MPP) values in the main database. The MPP value represents the physical size of each pixel in an image. Assuming wrong MPP values when building H&E datasets for deep model training can result in the models learning spatial inaccuracies and unrealistic measurements of structures. AAdditionally, inconsistent comparisons and misinterpretations may arise, leading to erroneous conclusions and potentially impacting the models' segmentation performance's effectiveness and reliability. The final dataset was built with no overlap, resulting in 268.274 patches of 1.623 annotations,




### 4.1.5 Dataset 5

To understand which impact inter-annotator variability has on the segmentation performance of both SegFormer and UNet six additional datasets were built of one annotation project. These datasets were used

to evaluate the potential biases of the algorithms towards one annotator style and the ability of the models to handle inter-annotator variability. For this, the annotations needed to be made on the same focus areas of a slide to ensure the model has to cope with variability of annotations within the exact same area of a tissue slide. In previous datasets different pathologist annotations were included; however these were not on the same areas of the slides. The annotation project includes 14 slides, each containing only one focus area, fully annotated by the two pathologists. Three training/validation datasets were built, two containing the annotations by only one annotator respectively and one containing the annotations from both. Additionally, three hold-out datasets to answer the above-named questions were constructed. Table 4.1 shows the number of patches and annotations, additionally providing a reverence number later used to refer to the specific sub-datasets.

| Annotator | Dataset Nr. | Nr. Patch |
|---|---|---|
| Annotator A | 5.1 | 4617.0 |
| Annotator B | 5.2 | 4676.0 |
| Both | 5.3 | 4962.0 |
| Test set A | 5.4 | 1512.0 |
| Test set B | 5.5 | 1100.0 |
| Test set Both | 5.6 | 1673.0 |

**Table 4.1** Annotator Variability Datasets

It is essential to note the following dataset characteristics in advance:

- The annotations of annotator B are much more precise and separated from each other. The focus areas annotated by B include blank areas, while A annotated the full focus space. The finer segments in B could introduce a high level of complexity for the models and a lack of complete context due to the white spaces.
- The annotations frequently disagree on the specific classes. An example for the class "fibrosis" is visualized in Figure 4.1.
- The test slides exhibit more fine-grained features and complex cases compared to the training data.

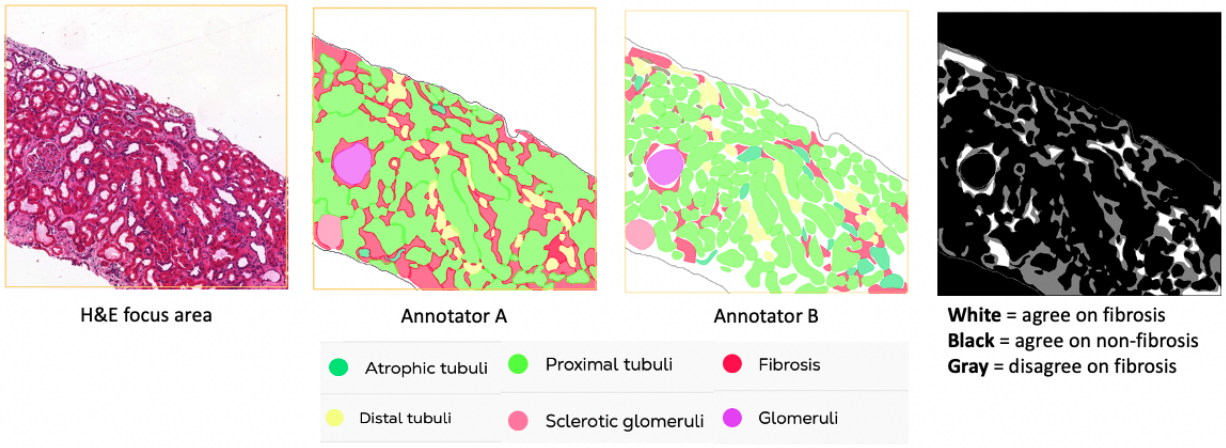



**Figure 4.1** Comparing Annotations on Fibrosis

## 4.2 Experimental Setup

Both UNet and SegFormer are implemented using the deep learning library TensorFlow. As a configuration framework, Hydra was used to streamline and standardize the setup of training, validation, and test steps,

ensuring reproducibility and facilitating easy experimentation with different parameters and configurations. Additionally, Hydra enabled easy debugging through efficient logging during the implementation phase of the SegFormer, within the modularity of sweep, training and evaluation configuration files, and simplifies the management of experiments. All experiments were run on a V100 or a NVIDIA A100 or an NVIDIA V100 gpu through Google Cloud [74]. This setup provided the necessary computational power to train the deep learning models effectively. Prefect [75], a workflow management system, was utilized for scheduling and managing the runs and sweeps of the experiments. Ray [76], a framework for distributed computing, was used for dataset building and inference tasks. The reference UNet model is initialized with a ResNet18 backbone, exhibiting excellent segmentation results on multiple H&E segmentation tasks. The parameters used for the initialization of the network were set according to best practices which have shown ideal segmentation accuracy. The SegFromer model was adapted from the GitHub huggingface implementation [77]. Both models were not initialized with pre-trained weights, allowing for a fair comparison and evaluation of the experiments.

# 5 Experiments

This chapter addresses the scientific questions that arose from the research gap. First, the experimental setup is justified and described. Then, the individual experiments are evaluated before a result is drawn that answers the scientific question. The algorithms, datasets, and environment utilized are described in chapter Chapter 4.

## 5.1 Research Design

The research questions derived from the research gap disclosed in Chapter 2 are addressed through several phases, ensuring a comprehensive and systematic comparison. Addressing the sub-research questions necessitates the execution of five phases, each comprising various practical experiments. Implementing the SegFormer architecture into real-world data is imperative, as theoretical evaluations fail to provide reliable insights. Given the lack of theoretical foundation for applying streamlined, computationally effective Transformer-based models to H&E segmentation, practical evidence remains the only viable means to effectively address these questions and gather meaningful insights.

**1st Phase: Implementation** The initial phase involves the implementation of the novel SegFormer architecture into the applied pathological AI pipeline, as described in Chapter 3.

**2nd Phase: Hyperparameter Testing & Quantitative Performance Evaluation** Subsequent to the implementation phase, the next step involved a series of experiments aimed at determining optimal machine learning and architectural parameters that result in the quantitative highest segmentation performance for the SegFormer model. Upon identifying functional parameter configurations, various practical experiments were undertaken to compare the quantitative segmentation performance with a reference UNet model. This phase encompasses the utilization of four distinct datasets, facilitating a comprehensive assessment of the H&E segmentation performance.

**3rd Phase: Robustness Analysis** The next stage provides insights into how the robustness of both models and therefore the generalizability in real-world scenario compare. This practical analysis evaluates the models' abilities to handle variations in the input data, ensuring that the comparison extends beyond ideal or controlled conditions.

**4th Phase: Qualitative Analysis** Following the quantitative assessment of parameter performance and robustness analysis, the subsequent phase delves into a qualitative analysis and comparison between the two architectures. The inference explores the architectures' behavior and efficacy in real-world histopathological segmentation tasks.

**5th Phase: Annotator Variability** This phase aims to assess SegFormer's response to inter-annotator variability compared to UNet. The model's capacity to handle biases stemming from single annotator datasets, with a comparative analysis against UNet, is also conducted.

**6th Phase: Analysis of the Applicability to an Industrial Setting** Assessing the memory requirements and the training times of both models enables the comparison of the architectures from an economic perspective, hich is essential to decide whether the implementation of SegFormer is feasible in a digital pathological setting.

## 5.2 Reference model

The UNet reference model utilizes a ResNet18 backbone comprising 14.321 million parameters. The selection of a ResNet18 backbone aimed to achieve a comparable number of parameters when compared to the SegFormer model. The reference model was trained without employing any pre-trained weight initialization. The decision to refrain from using pre-trained weights is justified by the objective of establishing a fair and equitable basis for the comparison between the two models. The performance evaluation of the UNet model is conducted based on the Macro Global F1 score, which consequently motivated the consistent utilization of this metric throughout all experiments.

## 5.3 Implementation

The effective incorporation detailed in Chapter 3 demonstrates the adaptability of the SegFormer Transformer model for application in the field of histopathology. Subsequent testing using dummy input data verifies the successful integration of the model into the AI pipeline.

## 5.4 Parameter Testing and Qualitative Evaluation

The primary objective of this phase was to formulate an initial configuration of network and training parameters for the SegFormer. Additionally, this phase addresses research question 1 by exploring the feasibility of effectively tuning the model within the histopathology context.

### 5.4.1 Fundamental Parameter Testing

Dataset 1 was used to find an initial configuration for SegFormer's parameters. This dataset strikes a balance between having enough data to perform meaningful experiments and being manageable enough to conduct multiple runs efficiently. The smallest SegFormer variant, b0, with 3.71 million parameters, was chosen to minimize the computational time during the initial runs. Each run was limited to 20 epochs to provide a primary overview of the parameters without excessive computational resource consumption.

In H&E segmentation, the Dice loss, in combination with a Cross-Entropy loss, is commonly used in a combined loss of $\lambda * CrossEntropy + (1-\lambda) * Dice$. The Dice loss component helps capture detailed structures, while the Cross-Entropy component assists in handling class imbalances and providing additional regularization during training. To ensure a fair comparison between known well-performing parameters and original SegFormer training standards, which utilize a pure Cross-Entropy loss, the two combinations, one using a higher shared Cross-Entropy, $\lambda = 0.9$, and one using a higher Dice loss, $\lambda = 0.1$, were tested. It is pertinent to mention that additional experiments were conducted to assess the performance of pure Cross-Entropy, Dice, Focal, and Jaccard Loss. However, owing to the constraints of this thesis and the lack of notably significant outcomes, these results have been omitted at this stage. Four different learning rates, 1e-04, 1e-05, 1e-06 and 1e-07, were tested to identify the range within which the SegFormer model learns most effectively. Additionally, three weight decay values for the AdamW optimizer were included in the sweep. The following parameters were maintained across all runs: Batch size: 64, input shape: 224x224, seed: 42, and simple color augmentation was applied during training. Each of the 24 sweep runs ran for approximately 41 minutes. The evaluation of which parameter combination yields the best segmentation results was based on the segmentation Macro Global F1 score computed on the hold-out

set. This score is pivotal as it measures the models' performance on unseen data, thereby directly assessing the segmentation performance. Table 5.1 gives an easy comparison of the Global F1 scores for the four best performing runs and their parameter combination. Among the tested learning rates, a value of 1e-04 demonstrated superior performance, indicating that a higher learning rate facilitated faster convergence and more effective optimization of the network's parameters. Table 5.1 presents the four best performing runs from the hyperparameter sweep based on the criterion of Macro F1 score. Notably, only one run employing a higher Dice loss, associated with a low $\lambda$, is encompassed. This leaves to assume that the SegFormer favors a higher Cross-Entropy over Dice loss for better segmentation performance. In contrast, the impact of weight decay (wd) on the model's performance was minor. Among the tested wd values, a decay of 1e-06 slightly outperformed the other values. Although wd had a limited influence on the model's performance, it can be derived that a low decay plays a role in preventing the model to overfit and preserve the network's ability to generalize well. The superior performance of a higher learning rate of 1e-04 motivated two additional tests with learning rates in its vicinity, specifically 1e-03 and 5e-04. The latter yielded an even better Macro F1 score, visualized as run number 4 in Table 5.1 and black columns in Figure 5.1. Additionally to the Global F1 score, Figure 5.1 illustrates the test set's recall and precision. The Segformer-b0 exhibits comparable high recall compared to UNet but lacks precision. This phenomenon could be attributed to Segformer's Transformer-based architecture. The ability to model long-range dependencies effectively can lead to a broader context understanding, contributing to higher sensitivity, correctly identifying most positive classes, but introducing some false positives impacting its precision. The broader context also introduces more ambiguity and uncertainty when making precise pixel-wise predictions, reducing precision. These initial findings serve as a foundation for subsequent experiments to refine the model's performance and achieve even better results for pathological scan segmentation.

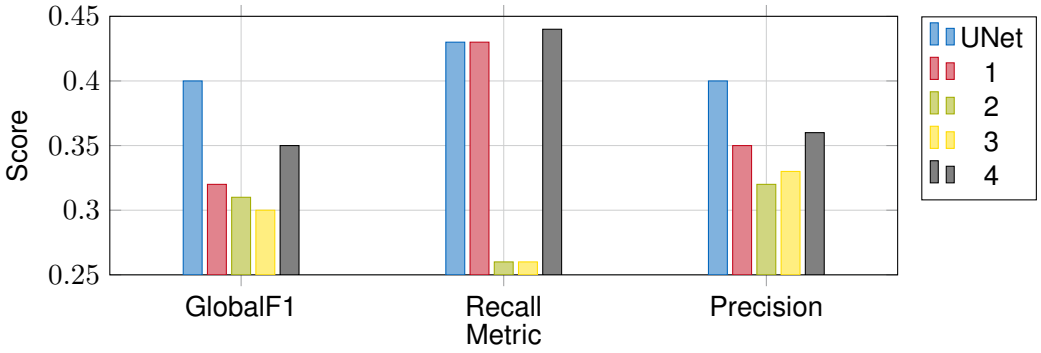



**Figure 5.1** Compared Performance of Tested SegFormer-b0 Parameters with Dataset 1

| Run | lr | wd | $\lambda$ | Global Macro F1 |
|---|---|---|---|---|
| 1 | 1e-04 | 1e-06 | 0.9 | 0.32 |
| 2 | 1e-06 | 1e-04 | 0.1 | 0.31 |
| 3 | 1e-04 | 1e-06 | 0.9 | 0.30 |
| 4 | 5e-04 | 1e-06 | 0.9 | 0.35 |

**Table 5.1** Impact of Learning Rate, Weight Decay, $\lambda$ of Combined Loss on Global Macro F1 of SegFormer-b0

### 5.4.2 Influence of Architecture Network Size

In the subsequent step, the impact of network type was assessed, focusing on settling on the most suitable network configuration for H&E segmentation. The high performance on general image segmentation and the extra complexity that would come with custom testing of architectural parameter configurations were reasons for the selection of predefined parameters. It is important to note that the b0 differs from

the b1 network in the size of the attention heads during the self-attention mechanism. The b2 network, in contrast to the b1, includes more SegFormer layers per Encoder block and a larger Decoder size. These facts suggest that the deeper networks yield higher performance as they capture more information during training. However, the training of the b2 network posed the challenge of higher memory requirements, and thus the need to resort to a NVIDIA A100 instead of the preferred cheaper NVIDIA V100 gpu. Figure 5.2 shows the performance of the three SegFormer network types compared to the UNet with ResNet18 backbone, proving that increased network size leads to better segmentation performance in terms of Global F1. The higher precision to recall for the b2, as opposed to recall exceeding precision for the ResNet18 and SegFormer-b0 and b1, can be attributed to the overall more profound architecture, which allows the network to capture the increased complexity of H&E slides better. The increased local attention in the b2, introduced through more Encoder layers within one Encoder block, allows it to delineate complex regions, leading to higher precision. However, the increased complexity might also cause the model to be more selective in making predictions, resulting in the lower recall seen in 5.2. However, the difference in quantitative performance between SegFormer-b1 and b2 was to marginal to justify the higher cost related to training a b2 network for H&E image segmentation. It should be noted that parameters were not explicitly tested for the b2 network, leaving room for potential performance improvements with different parameter configurations, especially considering the deep architecture, which can be sensitive to parameter tuning. Nevertheless, pursuing such experimentation at this stage would be too costly and not directly address the research question. The b1 network showed significantly better segmentation performance than SegFormer-b0, which motivated the prioritization of this network for further experiments.

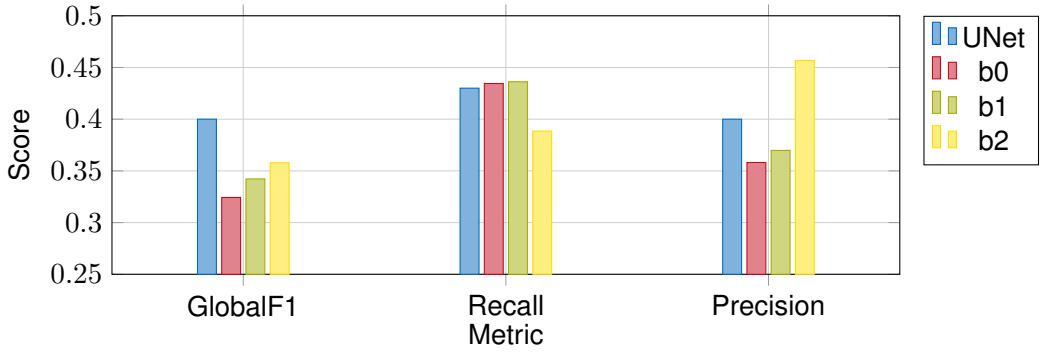



**Figure 5.2** Impact of SegFormer Network Size on Performance with Dataset 1

### 5.4.3 Influence of Learning Rate on SegFormer-b1

The initial parameter sweep was done on the b0 network, making it inevitable to retest the parameters for the b1 network. As the network size of deep learning models commonly affects the optimal learning rate in a first iteration a retesting of the learning rates 1e-04, 1e-05, 1e-06, 1e-07 was conducted. With all runs using a $\lambda = 0.9$ for the Combined loss function of $\lambda * CrossEntropy + (1 - \lambda) * Dice$, a weight decay of 1e-06 and seed of 42. The results are shown in Figure 5.3.

This sweep showed that the learning rate of 5e-04 for the deeper network yields the best segmentation results in terms of Macro F1.

### 5.4.4 Impact of Lambda in Loss Function on SegFormer-b1

The Combine loss function can be formulated as $\lambda * CrossEntropy + (1 - \lambda) * Dice$. Four different lambda magnitudes were tested to investigate the impact of different lambda values on the model performance regarding Global Macro F1 score, four different lambda magnitudes, 0.9, 0.8, 0.1, 0.2, were tested. Figure 5.4 illustrates the results combined with a learning rate 1e-04, revealing that a lambda favoring Cross-Entropy outperforms a higher Dice loss share for the SegFormer architecture. Despite efforts to optimize the weight combinations, the Macro F1 score could not be surpassed. Consequently, the initial weighting

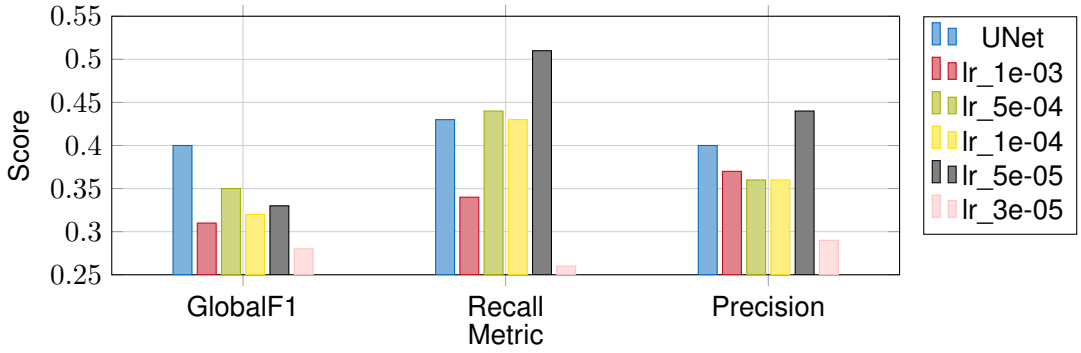



**Figure 5.3** Influence of Learning Rate on Performance of SegFormer-b1 with Dataset 1

with $\lambda = 0.9$ was retained. It is essential to highlight that additional lambda values were subjected to testing; however, their presentation was omitted due to the extensive scope of the experiments and the limitations of this thesis. Nonetheless, it can be concluded that all optimizations failed to yield superior outcomes compared to the results illustrated in Figure 5.4.

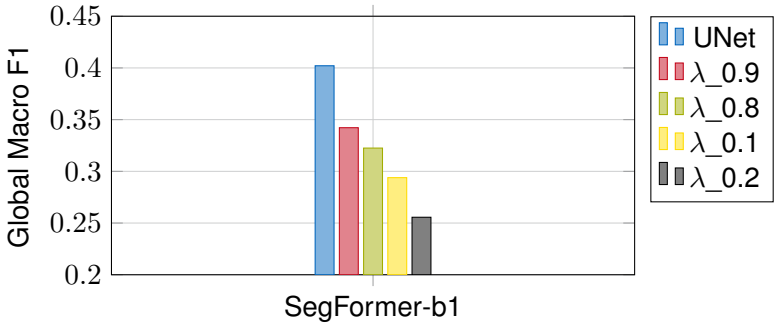



**Figure 5.4** Influence of Shares in Combined Loss Function on Global Macro F1 with Dataset 1

### 5.4.5 Impact of Stain Argumentation during Training

To investigate the impact of additional data augmentation, specifically including stain normalization to the augmentation techniques during the training process, on the model's performance, a third iteration on the b1 network was conducted. During initial parameter testing, color augmentation introduces controlled image appearance variations, enhancing model resilience to diverse conditions. Subsequent stain augmentation addresses histopathological image variability, simulating staining procedure differences and thereby enhancing the model's adaptation and generalization capacities. The internal implementation allowed for configuring a transformer module that automatically applies additional stain variations to the training data based on the parameter values of do_stain (either true or false). It was decided to keep the color augmentation constant to evaluate if H&E-specific data perturbations yield better Macro F1 on the hold-out set. Figure 5.5 presents qualitative outcomes with consistent color augmentation and varied stain augmentation and learning rates. The results show that augmentation does not notably affect the Macro F1 score; however, non-stain augmentation slightly outperforms it. This might be due to misalignment between synthetic and actual staining diversity, hindering the model's effective adaptation and learning of relevant patterns, potentially leading to lower segmentation performance and reduced generalization.

### 5.4.6 Input Size

The effect of the H&E input size on the model's performance was investigated to assess if the SegFormer needs larger input images for increased performance. The complex full H&E slide is commonly split into

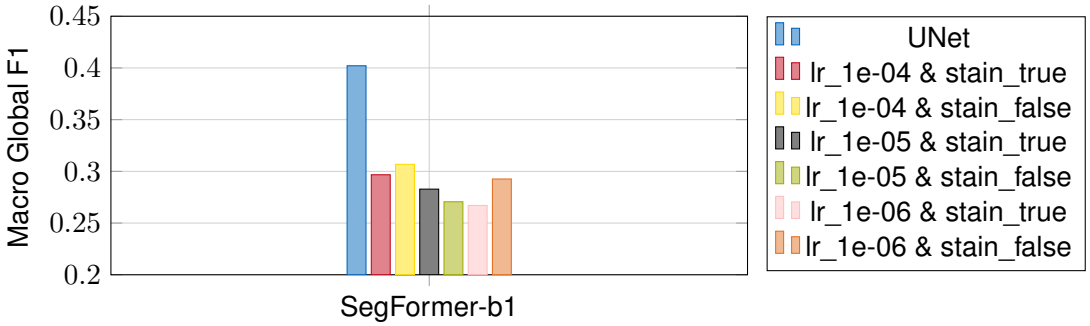



**Figure 5.5** Influence of Stain Argumentation and Learning Rate on Global Macro F1 with Dataset 1

small input patches to allow for computational efficiency. CNN-based architectures, like UNet, are exemplary at extracting local features from these smaller size inputs. In contrast, ViT models are designed to be trained on the whole input to fully capitalize on the global attention they provide. Typically training images for ViTs exceed sizes of 512x512. To investigate whether a larger input shape can aid quantitative performance, dataset 2, which contains 512x512 H&E patches, was used to train a SegFormer model. The larger input image size directly relates to a higher number of tokens in the input sequence to the self-attention mechanism, resulting in increased computational complexity when calculating the dot product between Query, Key, and Value vectors. This requires higher memory consumption and necessitates the use of the NVIDIA A100 gpu. To manage resources, this experiment was conducted with the smaller SegFormer-b0. The b0 network was tested on three different learning rates and compared with the reference UNet trained on the same dataset. As shown in 5.6, the results indicate that the reference UNet still outperforms the SegFormer model, regardless of larger input size. An additional experiment tested the influence of the lambda in the combined loss function on segmentation performance for larger input sizes. The $\lambda$ values examined for the combined loss, adhering to the same equation as previously mentioned, encompassed 0.9, 0.5, and 0.1, coupled with the documented best performing learning rates of 1e-04, 3e-05, and 5e-05. The findings depicted in Figure 5.6 illustrate that allocating a more significant proportion to Dice loss, achieved by reducing $\lambda$, does not approach the segmentation performance attained through $\lambda = 0.9$. This affirms the significance of a higher weighted Cross-Entropy loss for the segmentation performance of SegFormer, irrespective of the input image size.

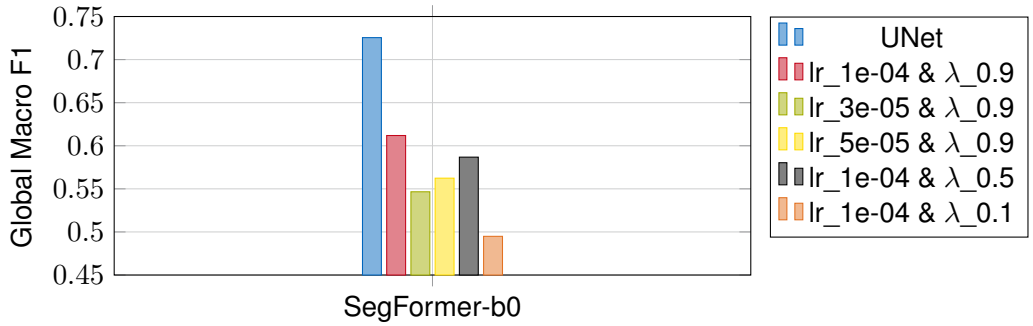



**Figure 5.6** Impact of Learning Rate, $\lambda$ value in Combined Loss on Performance with Dataset 2

### 5.4.7 Impact of Reduced Dataset Size

To test the influence of a smaller dataset on the segmentation performance, the best performing parameters from previous experiments were utilized to conFigure b0 and b1 SegFormer networks, which were then trained on dataset 2. Figure 5.7 presents the segmentation performance of the two SegFormer types compared to the reference UNet model, measured using the Macro Global F1 score. Surprisingly, the performance discrepancy between UNet and SegFormer on this smaller dataset was even more extensive

than on the training set with more data. This can be explained by the fact that Transformers generally require more training data to learn relations and contexts, lacking the inductive bias, explained in Chapter 3, that CNNs inherently possess. These results provide valuable insights into the viability of utilizing Transformer models for H&E segmentation tasks, especially when the training data is sparse, and ground truth annotations are challenging to acquire. The experiment also proves the known observation that not-so-deep networks tend to perform and generalize better on smaller datasets for the SegFormer architecture. The deeper b1 network might excel in capturing fine-grained details and intricate structures, contributing to higher recall. However, such fine details may not consistently appear for this small dataset, leading to an overemphasis on specific patterns and reduced overall segmentation performance, as reflected in the macro global F1 score. The segmentation task for this dataset is binary, additionally providing an insight into how the SegFormer model compares to UNet when faced with a different number of classes to segment. As the difference between the UNet and SegFormer performance exceeded compared to the 4-class segmentation task with the previous experiments, it can also be hypothesized that the SegFormer has difficulties with a binary segmentation task. Transformer models are known for performing better with more classes in a segmentation task due to the self-attention mechanism, which benefits from more positional relationships and context introduced through a larger number of classes. This assumption could be made for the SegFormer but needs to be confirmed by testing the implementation on a dataset introducing an increased number of classes. Such a multi-class dataset will be used in the robustness analysis described in Section 5.5.1. Larger datasets typically provide sufficient samples for the model to generalize and capture intricate patterns. However, in smaller datasets like the one at hand, capturing finer details and local features may be limited due to the scarcity of samples. A higher weighting of the Dice loss in the combined loss function is expected to prioritize accurate boundary delineation and preservation of local structures since it measures the pixel-wise overlap between predicted and ground truth segmentation. The SegFormer-b1, with its best performing parameters, was utilized to attain a fair insight into how the dataset size influences the segmentation performance when $\lambda$-values 0.9 and 0.1 are used with the combined loss function during training. Figure 5.8 proves the assumption that the SegFormer-b1 trained with a lower lambda, related to a higher Dice loss, outperforms the model trained with a higher weighted Cross-Entropy loss on a smaller dataset. The high dominance of recall over precision in this experiment can be attributed to the model's limited diversity and potential overfitting to positive instances. In contrast, for the previous experiments, a dominance of recall over precision was still present, but the trade-off was more balanced, likely due to the increased variability and improved generalization when confronted with more training data.

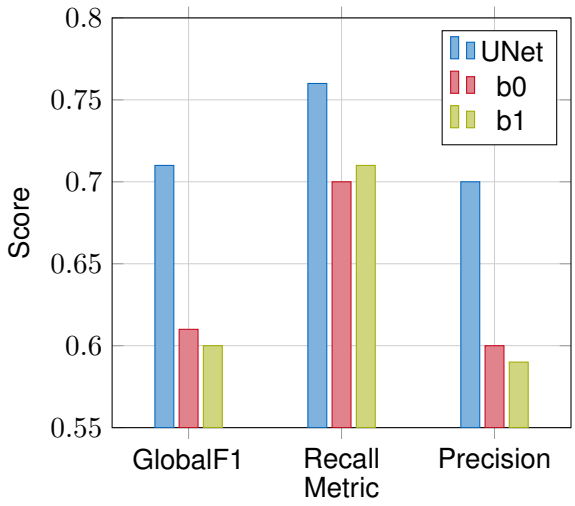



**Figure 5.7** Influence of Network Size on Performance with Dataset 4

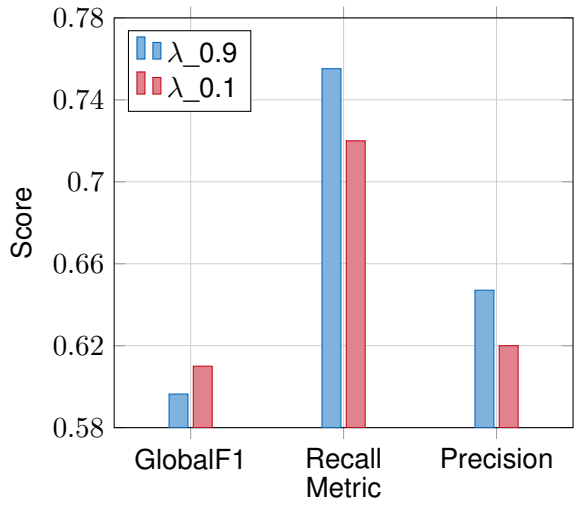



**Figure 5.8** Influence of $\lambda$ in the Combined Loss on Performance of SegFormer-b1 with Dataset 4

## 5.5 Robustness Evaluation

The assessment of the third research question, pertaining to the comparative robustness of the SegFormer architecture, was conducted utilizing a distinct dataset from that employed in the second phase of experimentation. Dataset 3 is the largest one available during this thesis. It was selected to compare the robustness of the two models because of the diverse range of complex examples it includes, enabling both models to learn from challenging scenarios and reducing potential bias or overfitting during training. Ensuring such robustness is vital to enable the models to effectively manage the perturbations introduced during the robustness analysis, thereby facilitating a fair comparison. Dataset 3 presents a six-class segmentation problem, allowing an assessment of how the SegFormer performs on a larger class segmentation problem. In the search for appropriate parameters to initialize the SegFormer, the best performing parameters from previous experiments were re-evaluated. Tested were the model size, b1 and b0, together with additional stain augmentation, do_stain parameter set to true or false, and the best performing learning rates, 1e-04, 3e-05, 5e-05. Table 5.2 show the Global Macro F1 on the hold-out set for the four best performing runs. Comparing this score, recall, and precision to UNet's performance on Dataset 3 shows that the performance difference is less pronounced than in the previous two-class segmentation problem. This difference can be attributed to the SegFormer's potential superiority in handling a more extensive multi-class problems or the overall increase in training samples, allowing the models to learn from more diverse data. Further experimentation is required to pinpoint the exact reason behind this observation. Based on the results from Figure 5.2, the SegFormer-b1 network with a learning rate of 1e-04 and no additional stain augmentation was trained for the subsequent robustness experiment.

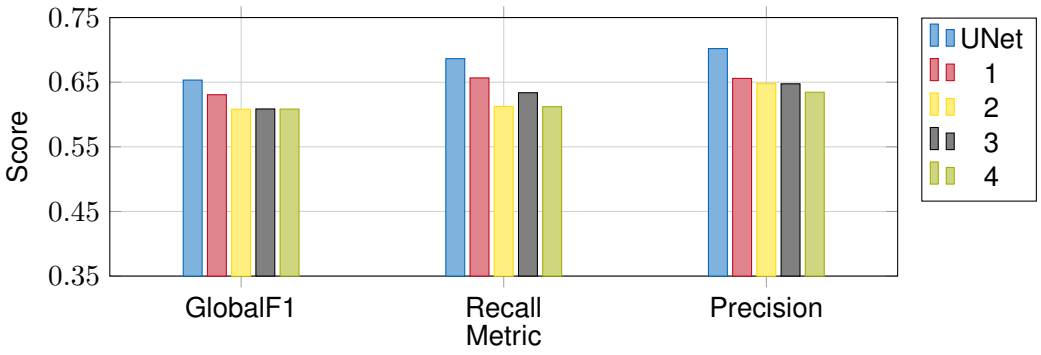



**Figure 5.9** Compared Performance of Tested SegFormer Parameters on Dataset 3

| Run | lr | Model Size | do_stain | Global Macro F1 |
|---|---|---|---|---|
| 1 | 1e-04 | SegFormer-b1 | false | 0.63 |
| 2 | 5e-05 | SegFormer-b1 | true | 0.61 |
| 3 | 1e-04 | SegFormer-b1 | true | 0.61 |
| 4 | 1e-04 | SegFormer-b0 | false | 0.61 |

**Table 5.2** Impact of Learning Rate, Model Size, Stain Augmentation with Dataset 3 on Performance of SegFormer

### 5.5.1 Robustness SegFormer vs. UNet

In the context of machine learning and deep learning, robustness refers to the ability of a model to maintain stable and accurate performance when exposed to various types of perturbations or changes in the input data. These perturbations can include noisy data, adversarial examples, variations in lighting or viewpoint, occlusions, and other distortions.

Robustness tests are crucial in assessing a model's ability to maintain performance and adapt to various conditions, encompassing diverse noise and intensity variations. Evaluating the robustness of the two architectures provides insights into their reliability and generalizability in real-world scenarios. Robustness is desirable because it indicates that the model can handle noisy or imperfect data, making it more reliable and applicable in practical settings. In some contexts' it might be required to compromise performance for robustness. This phase of the experimental part of the study targets research question 3 and can be divided into two subsections. The first subsection conducts a series of general data augmentation experiments to compare the robustness of SegFormer and UNet models on common input perturbations. The second subsection focuses on color deconvolution experiments, which are specific to the channels eosin and hematoxylin. These experiments are essential for comparing the two model architectures regarding H&E segmentation, necessitating H&E-specific augmentation tests. It is important to note that the best performing UNet model, with the ResNet18 backbone, outperforms the SegFormer regarding the Macro F1 score. Consequently, the difference in performance between the two models is not further emphasized in the evaluations. The main focus lies on the robustness behavior of the models, not the performance differences.

### 5.5.2 General Data Augmentation Experiments

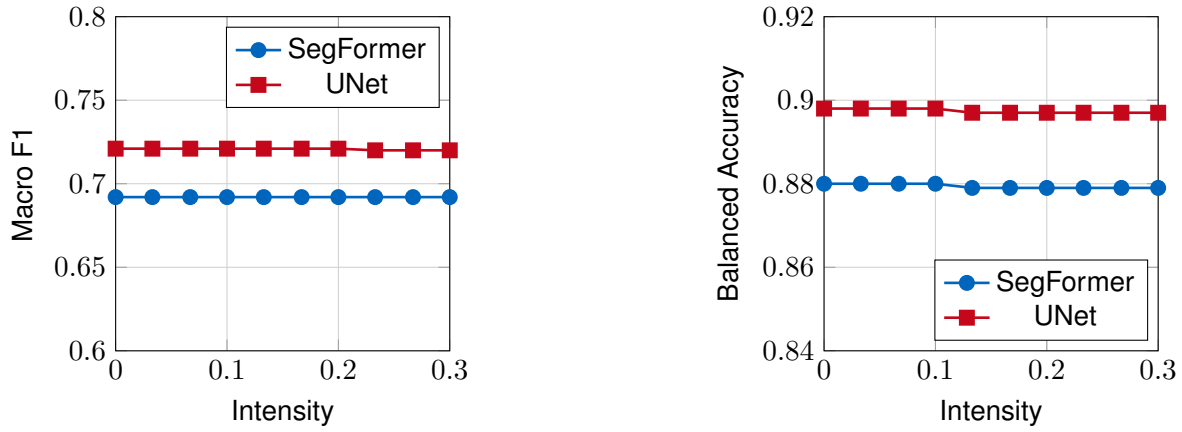



**Figure 5.10** Impact of Increased Brightness on Performance of SegFormer and UNet

Both networks demonstrated minimal changes in Macro F1 and accuracy metrics as brightness in the input data increased. This observation implies that the SegFormer and UNet architecture are equally robust to brightness augmentation, indicating their ability to maintain stable performance even under variations in image brightness.

Through various experiments, the robustness of the SegFormer and UNet architectures was assessed and compared by subjecting them to different data augmentation scenarios, including both common input perturbations and H&E-specific augmentations. The results showed that both architectures experienced declines in performance when subjected to perturbations, with the UNet generally exhibiting more significant declines in both Macro F1 and accuracy.

Remarkably, the SegFormer outperformed the UNet at the highest level of augmentation in almost all experiments. The experiments involving general augmentation highlighted that the performance decrease patterns are overall similar, with the UNet's declines being more intense. This implies that the architectures

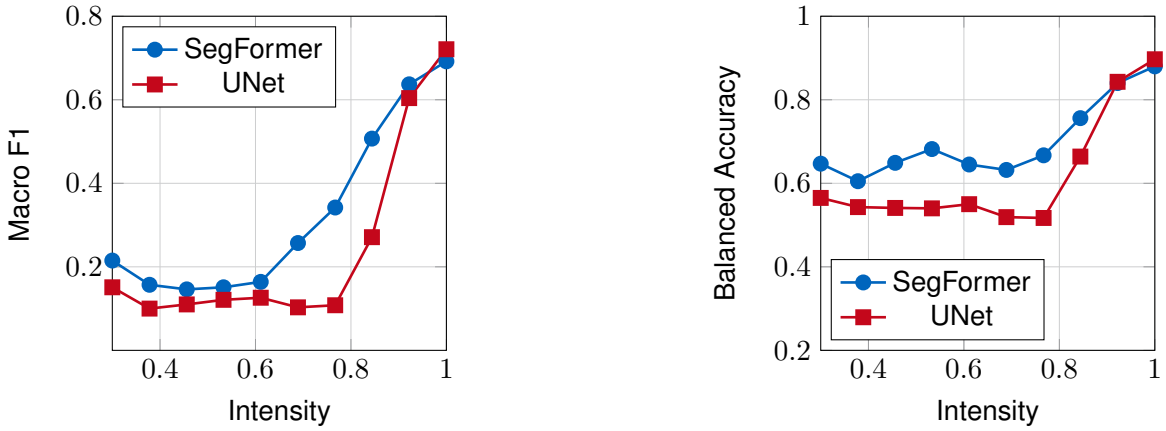



**Figure 5.11** Impact of Decreased Hue on Performance of SegFormer and UNet

The SegFormer and UNet networks demonstrated a decline in performance as the hue intensity decreased. Both networks exhibited similar performance changes during the transition from level 1 to level 10 of hue intensity. The decrease in performance, particularly in the Macro F1 score, highlighted the sensitivity of both networks to changes in hue. Interestingly, the SegFormer network showed a slightly lower overall performance drop, particularly in accuracy. However, its response to hue changes was more inhomogeneous, suggesting its susceptibility to variations in color information at different hue intensity levels. In contrast, the UNet displayed a more stable accuracy performance after the initial drop from level 1 to 4. However, its overall accuracy was lower than that of SegFormer at lower hue intensities, indicating that SegFormer is more robust in handling hue alterations. Similarly, the SegFormer outperformed the UNet in Macro F1 at the lowest hue intensity, emphasizing its superior robustness in handling hue variations. This implies that the UNet might be more sensitive to hue variations, impacting the model's ability to generalize and make accurate predictions in practical scenarios where shifts in color information occur.

exhibit similar susceptibilities and behavior under the same variations but to varying degrees. Additionally, the SegFormer demonstrated resilience to most H&E-specific input perturbations, maintaining relatively stable or only minimally decreasing performance across multiple intensity levels. In contrast, the UNet's performance exhibited predominantly steep and linear declines in these scenarios, indicating its heightened sensitivity to these augmentations. These findings suggest that SegFormer's transformer architecture and attention mechanism contribute to its ability to capture global context and spatial relationships, enabling it to preserve essential features better and perform well under perturbations.

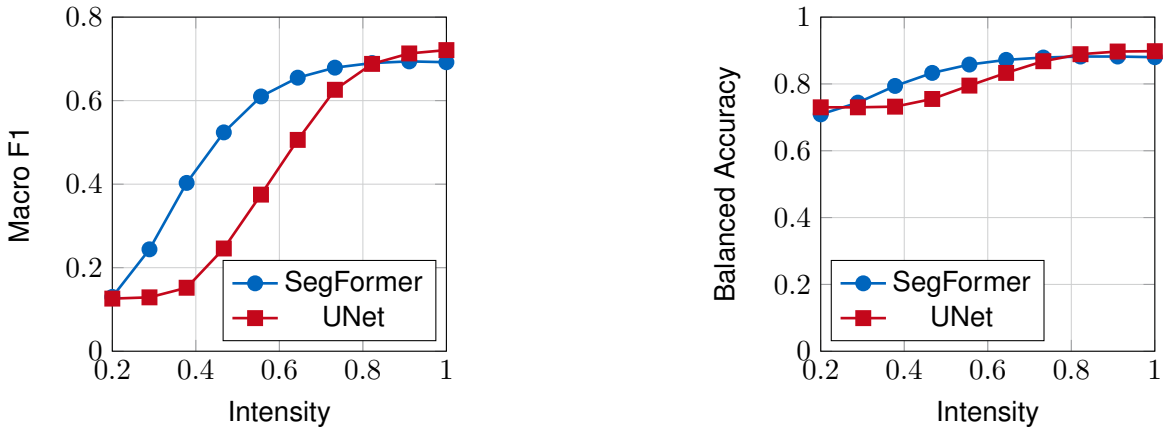



**Figure 5.12** Impact of Decreased Contrast on Performance of SegFormer and UNet

The nearly identical score decline from level 10 to level 1 suggests a consistent impact of contrast intensity on model performance, independent of the architecture. Different from other augmentation methods, the SegFormer's comparable performance to UNet in balanced accuracy indicates its effectiveness in handling contrast variations SegFormer shows a slightly steeper decline in performance at the lower end of the intensity spectrum, suggesting its sensitivity to extreme intensity variations. On the other hand, the UNet demonstrates a more intense decline in Macro F1 and accuracy at the middle contrast levels, implying that initial and final contrast decline may not significantly impact UNet's performance. However, the middle levels have a notable effect, suggesting that the UNet might be particularly sensitive to moderate contrast variations.

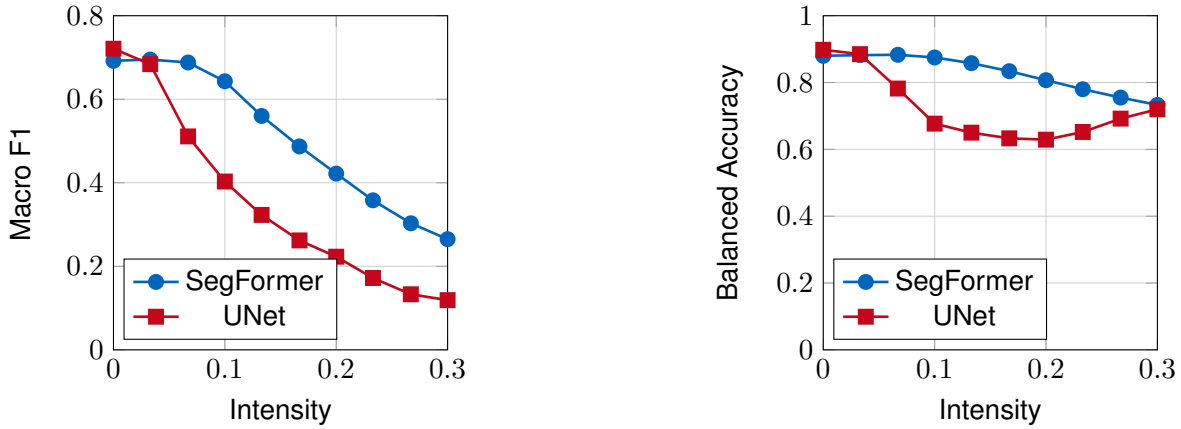



**Figure 5.13** Impact of Gaussian Noise on Performance of SegFormer and UNet

Both networks experience a significant decline in performance under the influence of Gaussian noise, indicating the adverse impact of this augmentation on their overall performance. However, the SegFormer exhibits a smaller decline in Macro F1 score compared to the UNet and even outperforms the UNet at higher augmentation intensity, implying that the SegFormer may be more resilient to Gaussian noise in terms of Macro F1 evaluation. Furthermore, the SegFormer displays lower susceptibility to changes in balanced accuracy than the UNet, demonstrating its ability to maintain a balanced performance. While both architectures achieve similar final accuracy performance, the UNet exhibits a more pronounced decline in the early stages of augmentation, followed by a less impactful decline in later stages, leading to a slight accuracy increase. This observation suggests that the UNet's initial decline might be more affected by noise-induced perturbations, while its later stages could benefit from the regularization effects of the noise, potentially aiding its accuracy recovery. These findings highlight the SegFormer's stability and resilience to Gaussian noise, particularly in Macro F1 and balanced accuracy evaluation.

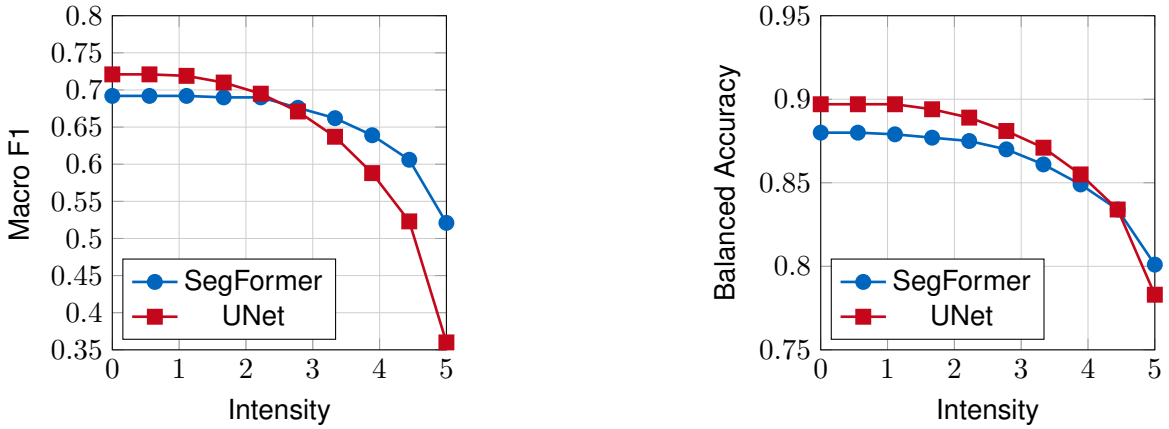



**Figure 5.14** Impact of Gaussian Blur on Performance of SegFormer and UNet

Both architectures display a similar performance decline pattern over the different intensity levels of Gaussian blur. However, the SegFormer exhibits a smaller and less steep overall decline in contrast to the UNet, indicating its greater resilience to the smoothening effect induced by Gaussian blur. The more pronounced decline in performance, especially Macro F1, for the UNet suggests its higher sensitivity to Gaussian blur augmentation. From level 1 to 7, the SegFormer demonstrates superior robustness under Gaussian blur, maintaining its performance, while the UNet starts experiencing a performance drop from level 3 onwards. Additionally, at level 5 for Macro F1 and level 9 for accuracy, the SegFormer outperforms the UNet, implying its advantage in accurately predicting positive instances despite Gaussian blur.

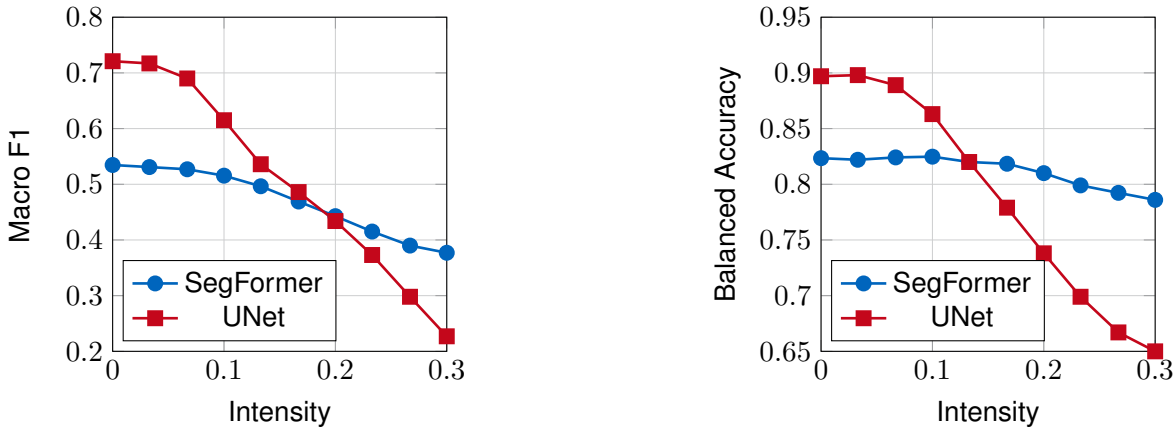



**Figure 5.15** Impact of Hematoxylin Noise on Performance of SegFormer and UNet

This experiment shows that SegFormer exhibited a higher level of resilience to hematoxylin noise in the input than UNet. Its Macro F1 score only slightly declined with increasing h-noise intensity, suggesting its reliability in tasks with such variations. Conversely, UNet experienced a noticeable decline in Macro F1 score by around 0.5 units, while the decline for SegFormer was less than 0.2 units. The accuracy of SegFormer remained relatively stable, with only a slight decrease observed after reaching level 5 out of 10. In contrast, UNet showed a significant decline in performance between intensity levels 3 and 8 for both Macro F1 and accuracy. This suggests that UNet is highly susceptible to hematoxylin noise intensity variations within this specific range, potentially compromising its effectiveness. Interestingly the SegFormer architecture outperforms the UNet after level 5 or an intensity higher than 0.133 in both measured scores.

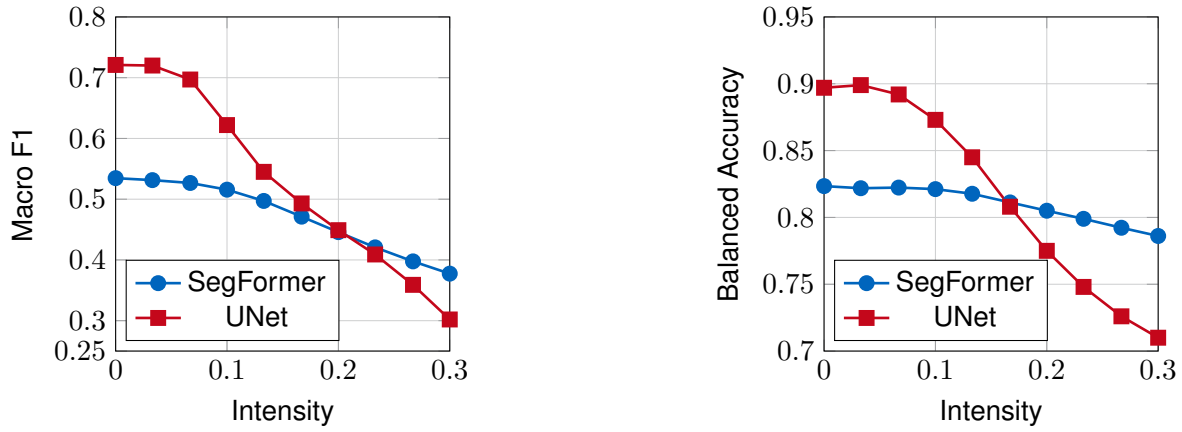



**Figure 5.16** Impact of Eosin Noise on Performance of SegFormer and UNet

In this experiment additional eosin noise was intorduced to the input image. Similar to the hematoxylin noise experiment, the UNet architecture exhibited a more substantial decline in performance compared to the SegFormer. The UNet's Macro F1 score decreased by approximately three times, and its accuracy experienced a decline about four times stronger than that of the Transformer model. This model demonstrated resilience in Macro F1 to e-noise variations until level 4, while the UNet's segmentation score showed a substantial decrease starting from level 1. In terms of accuracy, the SegFormer maintained its performance until level 7, whereas the UNet experienced a decline from the beginning. These findings highlight the SegFormer's robustness in the early stages of e-noise perturbation, while the UNet is more susceptible to performance degradation across all stages.

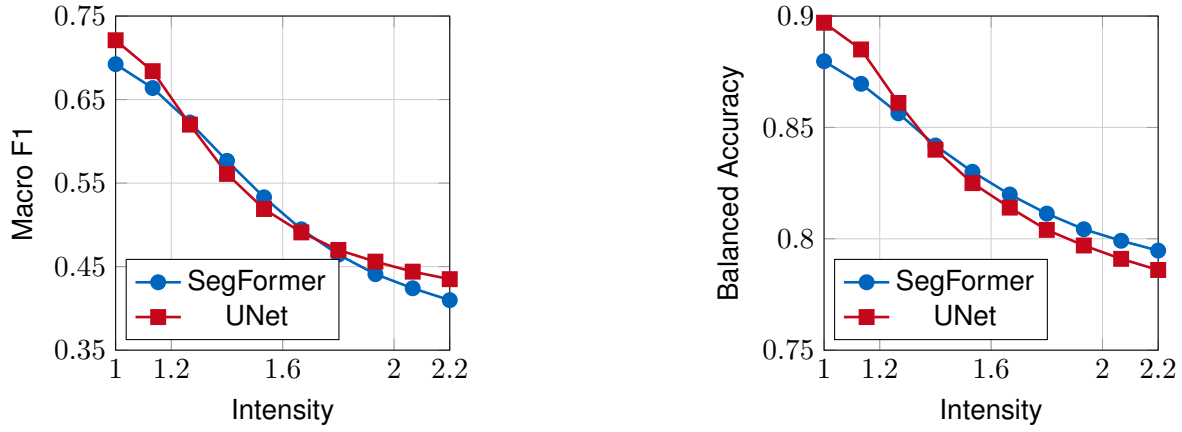



**Figure 5.17** Impact of an Increased Eosin Channel on Performance of SegFormer and UNet

In this experiment, the intensity of the eosin channel was varied from 1 to a max intensity of 2.2 over ten levels. Both architectures showed a similar performance decline for both metrics, with a decline of around 0.1 units for Macro F1 and 0.28 units for accuracy. This suggests that the increase of the eosin channel for the input constantly impacts overall performance independent of the architecture.

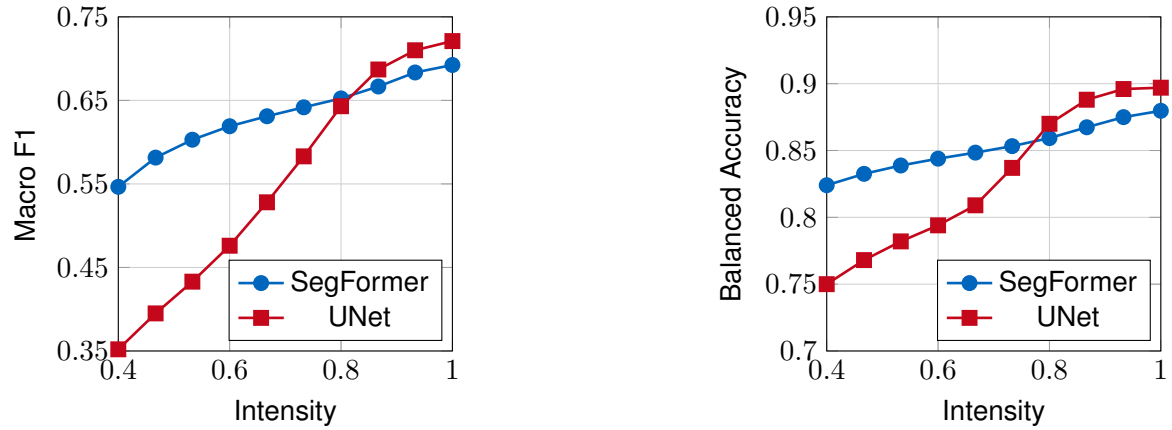



**Figure 5.18** Impact of a Decreased Hematoxylin Channel on Performance of SegFormer and UNet

In the subsequent experiment, the intensity of the eosin channel was reduced across ten levels. The overall decline in performance for both architectures was more intense than in the previous experiment, where the eosin channel intensity was increased. The SegFormer demonstrated a notably smaller decline in both Macro F1 and accuracy metrics, while the UNet experienced nearly double the decline in both aspects. The similar decline patterns, indicate that both networks' performance is influenced similarly, though not to the same degree, by the intensity of the eosin channel.

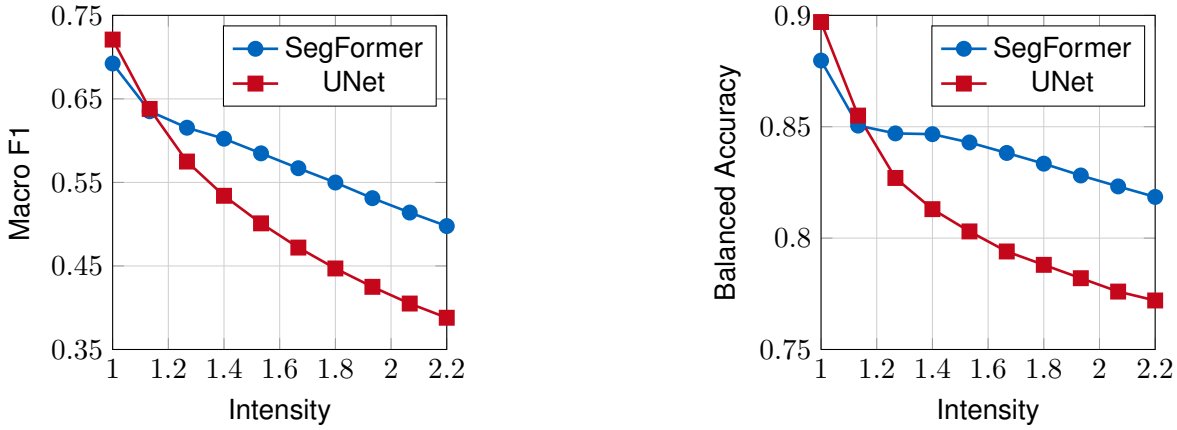



**Figure 5.19** Impact of an Increased Hematoxylin Channel on Performance of SegFormer and UNet

In this experiment, the intensity of the hematoxylin channel varied from the baseline value of 1 up to 2.2 across ten levels. Both models exhibited a significant drop in performance for the first two increases. However, after this initial drop at level 2, the decline in performance became more gradual and slower for the SegFormer compared to the UNet, with the SegFormer surpassing the UNet in performance beyond this level. The overall difference in Macro F1 and accuracy scores was more pronounced for the UNet. These findings indicate that the SegFormer architecture is less affected by the intensity variation and demonstrates greater robustness under hematoxylin intensity augmentation.

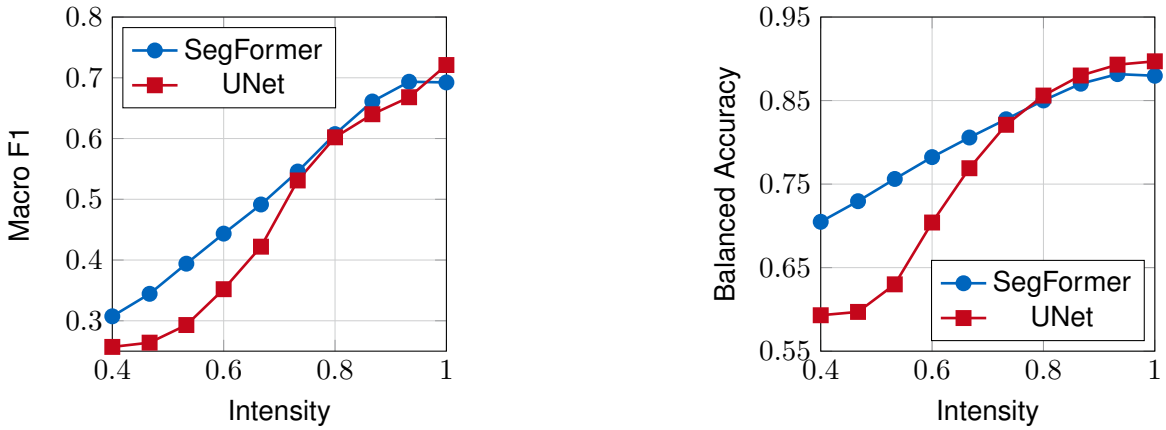



**Figure 5.20** Impact of a Decreased Hematoxylin Channel on Performance of SegFormer and UNet

In the final experiment, the intensity of the hematoxylin channel was systematically decreased from the baseline value of 1 to 0.4. Both models demonstrated a decline in performance. However, the CNN-based model exhibited a more pronounced decline in Macro F1 and balanced accuracy. The decline in performance for the UNet was steep and linear starting from level one. At the same time, the SegFormer maintained a relatively constant performance for the first two levels but experienced a linear decrease in both metrics thereafter. Remarkably, the decrease in accuracy performance for the UNet was twofold more intense and represented one of the most substantial declines observed among all experiments compared to the SegFormer. These results imply the UNet's heightened sensitivity to this specific augmentation.

### 5.5.3 Effect of Varying Loss Weighting on the Robustness of SegFormer-b1

The prior experiments were inconclusive regarding a general $\lambda$ value within the combined loss function $\lambda * CrossEntropy + (1 - \lambda) * Dice$ for optimizing segmentation performance with the SegFormer architecture. The optimal distribution of loss function components within the combined loss, yielding the highest macro F1 performance on the hold-out dataset, varied across different datasets. These variations motivated another experiment evaluating how the $\lambda$ value in the combined loss impacts the robustness of the SegFormer model, thus testing research question 4. To evaluate this, a binary model was trained on dataset 3 with a class imbalance of more non-necrosis to necrosis cases. Given the performance superiority of the Dice coefficient in class-imbalanced problems, an investigation into the potential impact of a low $\lambda$ on the performance-robustness trade-off was conducted. While Dice loss is common in image segmentation tasks and Cross-Entropy loss in classification tasks, no established evidence or consensus suggests that a transformer model trained on Dice loss is inherently more robust to input perturbations than one trained on Cross-Entropy. Two identical models were trained to examine this specifically for the SegFormer architecture, differing solely in the $\lambda$ value employed: 0.9 and 0.1. The following experiments evaluate the impact of different input perturbations on the quantitative performance of the networks.

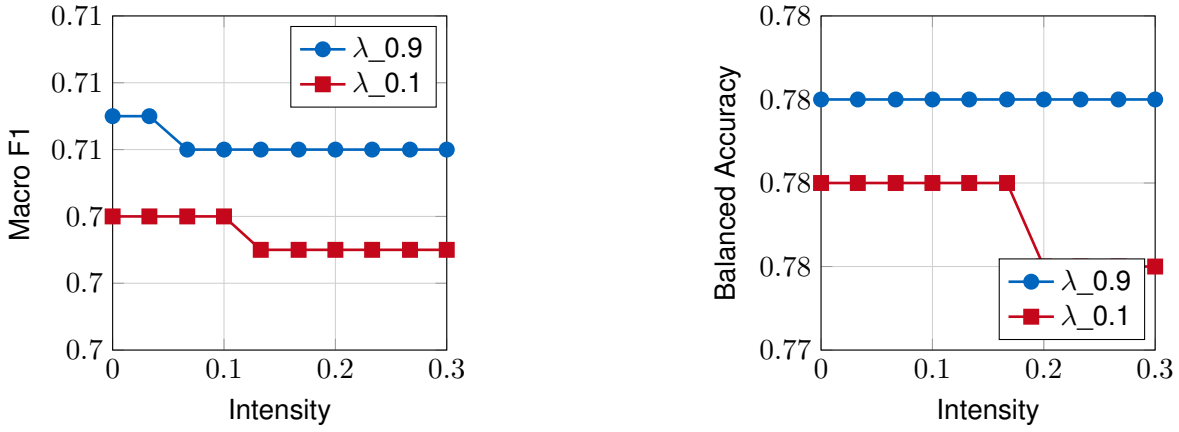



**Figure 5.21** Impact of Increased Brightness on SegFormer Performance

The two models were tested under varying brightness conditions, showing minimal performance declines with increased brightness overall. However, the model trained on a $\lambda$ of 0.1 exhibited a slight accuracy drop in these conditions, highlighting the influence of loss functions on sensitivity to illumination changes. Notably, the drop in Macro F1 score was so minimal across both models that it can be disregarded, indicating the robustness of the architecture to brightness change.

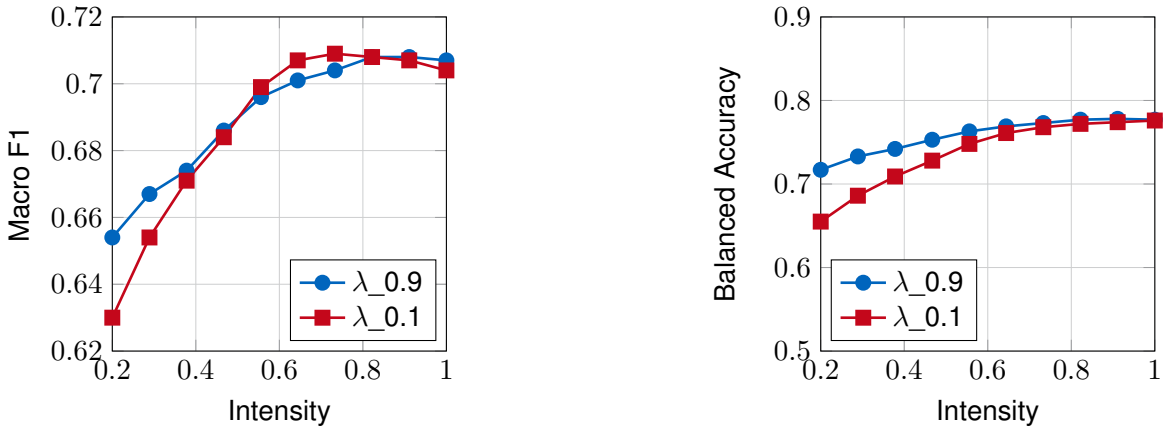



**Figure 5.22** Impact of Decreased Contrast on SegFormer Performance

The experiment demonstrated that under conditions of reduced contrast, both models experienced a decline in macro F1 score. The model trained on 10% Cross-Entropy and 90% Dice loss showed a slightly more pronounced drop than the one trained on a $\lambda$ of 0.9. However, the accuracy remained constant for both models with declined contrast. The overall decline in performance was minimal, emphasizing again the robustness under input transformation.

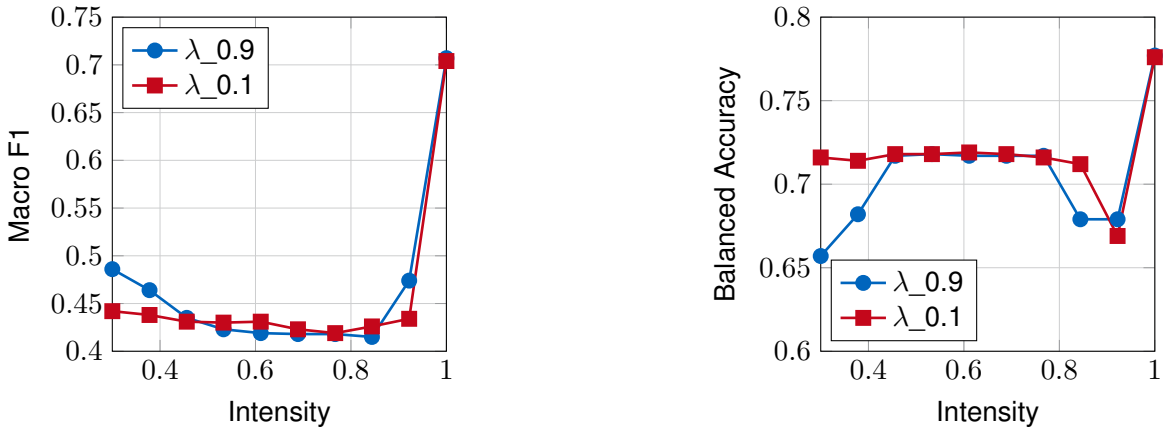



**Figure 5.23** Impact of Decreased Hue on SegFormer Performance

The experiment demonstrated a reduction in Macro F1 scores for both models when exposed to decreased hue conditions. Notably, the model trained with 90% Cross-Entropy exhibited a slight advantage in Macro F1 performance while experiencing a minor decrease in accuracy under the minimum hue level. The models showed negligible performance differences across hue levels 3 to 10, indicating their robustness in handling moderate hue variations. However, significant disparities emerged at hue levels one and two, suggesting the criticality of the initial hue levels for the models' performance, irrespective of the utilized loss function. This implies that higher hue accuracy leads to improved performance, and once the hue is only slightly reduced, further performance decline is limited to the initial drop.

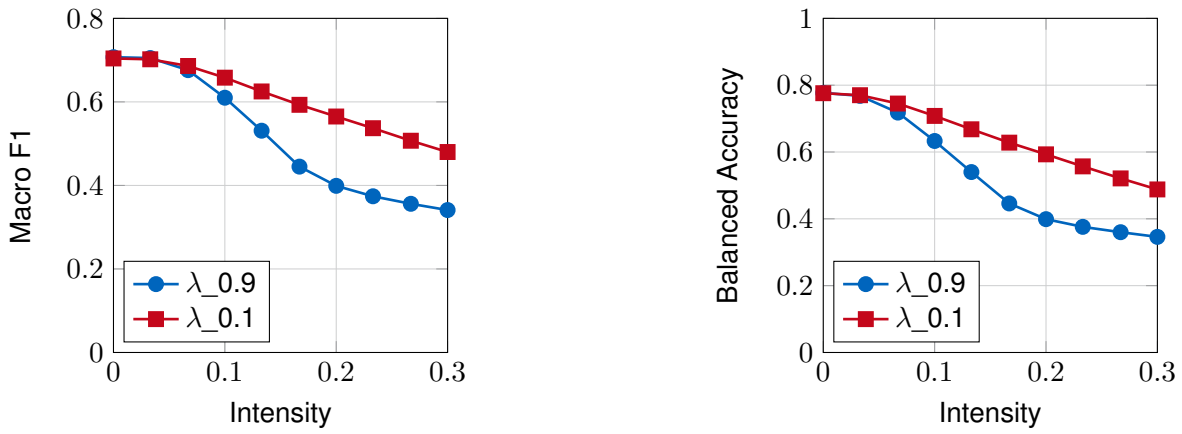



**Figure 5.24** Impact of Gaussian Noise on SegFormer Performance

The experiment revealed a nearly linear decline in both performance metrics under Gaussian noise, with the $\lambda = 0.9$ model being more sensitive to higher Gaussian noise, introducing random fluctuations or uncertainties. This suggests that the Dice loss encourages the model to focus on capturing overlapping regions and structural similarities, which is beneficial for handling blurry input data.

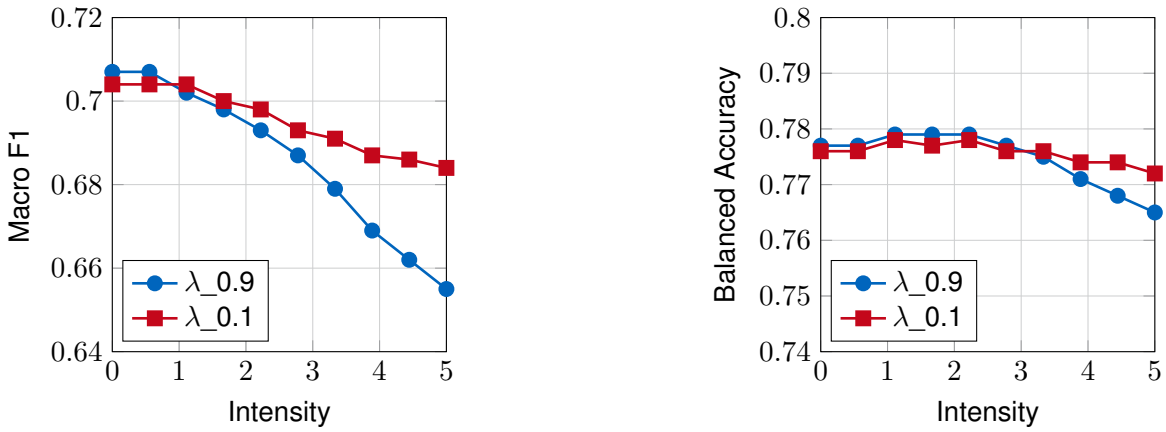



**Figure 5.25** Impact of Gaussian Blur on SegFormer Performance

The robustness of the model on smoothened input is evaluated by testing Gaussian blur. The reduction of high-frequency details mimics the effect of blurriness or defocus in real-world situations. The decline in macro F1 due to Gaussian blur was observed to be linear but less intense than that caused by Gaussian noise, indicating that noise perturbations present a greater challenge to the model's performance. The $\lambda = 0.9$ model exhibited a higher decline in Macro F1, but both models showed similar stability in accuracy, with only a slightly higher decline for the $\lambda = 0.9$ model during the last three intensity levels. This highlights the model's architecture robustness in maintaining precision-recall balance under varying blur and intensity conditions.

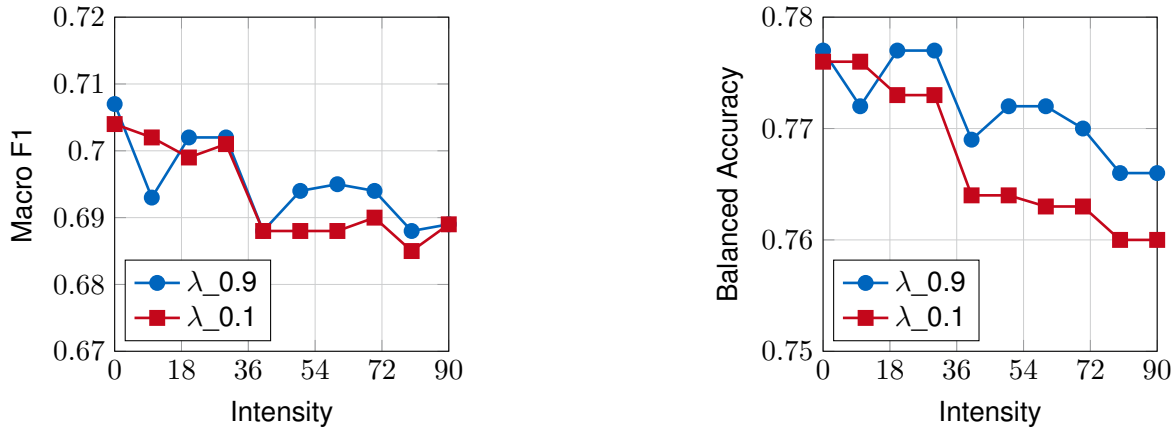



**Figure 5.26** Impact of Rotation of the Input on SegFormer Performance

The influence of input rotation on the macro F1 score and balanced accuracy on the two models was investigated through 10 levels, each introducing a 10-degree rotation of the input, ranging from 0 to 90 degrees. Observations revealed that both models were only marginally affected in performance. The changes in macro F1 and balanced accuracy score were only around 0.01 and 0.16 points respectively, which is negligible compared to preceding experiments. This minimal impact could be attributed to chance or minimal dependence of the model's learning on the orientation of the input H&E patch. Each model showed varying degrees of improvement or deterioration for different rotation angles, making it challenging to determine whether one model performed significantly better. The observed bumpy performance changes and lack of clear convergence could be attributed to several factors. Firstly, the SegFormer's attention mechanism enables it to focus on relevant regions within the input patch, irrespective of its orientation. Secondly, the patch-based processing approach allows the model to capture local patterns and structures without relying on the global orientation of the entire image. Lastly, the multi-head attention approach enables both models to identify significant patterns and structures in the image regardless of their orientation. The slightly improved balanced accuracy of the SegFormer trained with $\lambda = 0.9$ for the combined loss function suggests that the Cross-Entropy loss contributes to the better spatial alignment of the segmented output with the ground truth when dealing with rotated input images. Overall, this experiment has shown that the architecture is less sensitive to input rotation.

Comparing the robustness of the SegFormer architecture trained with different loss functions provided insights into the model's ability to handle input variations relevant to real-world settings. The study revealed that the choice of loss function significantly shaped the model's behavior under random deviations across the entire image (e.g., Gaussian blur and Gaussian noise), implying distinct abilities to handle noise-induced variations in the data. The model trained on a higher weighted Dice loss turned out to be minimally more robust to input transformation, suggesting that the focus on spatial overlap in the Dice loss helps the model produce more consistent and stable segmentation results in the presence of small perturbations. However, the differences were so minor that the choice of the $\lambda$ value for the combined loss function could not be directly tied to the model's strong generalization capabilities to different lighting conditions and color variations.

## 5.6 Qualitative Analysis

After the quantitative experiments, the qualitative evaluation involved visualizing the predictions. Through inference on both models qualitative differences and similarities were evaluated. The graphical representation revealed previously unnoticed problems of the SegFormer architecture in H&E segmentation. Figure 5.27 shows the prediction by UNet on the left and on the right of SegFormer. Multiple things can be derived from this:

- The SegFormer model exhibits increased uncertainty in its predictions, evident by the notable lighter transparency observed in the segmentation mask, as depicted in a cutout from an inference slide in Figure 5.28. Elevated uncertainty in a medical segmentation model should be recognized as an aspect requiring attention rather than seen as inherently negative. This awareness of uncertainty prompts cautious interpretation of predictions, promoting informed decision-making in medical contexts.
- Figure 5.29 shows significant patching characteristics in the predictions. This behavior could be explained with the internal inference pipeline, which is designed and optimized to combine the prediction of UNet.
- Inconsistent segmentation prediction across patch boundaries can be noticed through the visualizations, as seen in Figure 5.30. The absence of positional encoding and the introduction of overlapping patch merging in the SegFormer architecture could contribute to these differing predictions. The model might struggle to recognize the continuity and relationships between adjacent patches, especially when dealing with complex structures like cell boundaries.
- Compared with the CNN-based predictions, the Transformer ones exhibit a substantial lack of fine-grained structure, falling back on coarser predictions for larger areas.

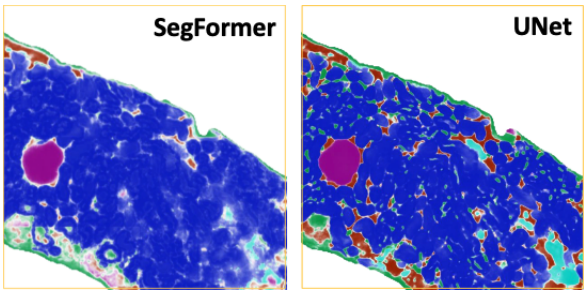



**Figure 5.27** Qualitative Comparison of SegFormer and UNet

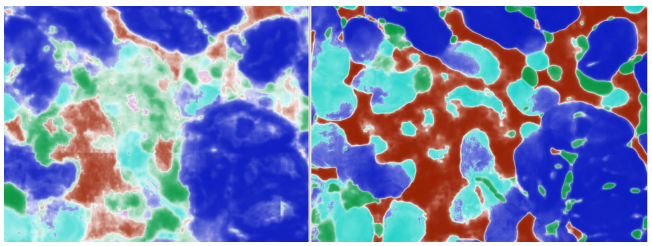


**Figure 5.28** Transparency Illustrating Uncertainties in SegFormer Prediction

By comparing the segmentation of SegFormer and UNet for the class "fibrosis" in the visualization tool ImFusion refined understanding of accuracy on significant classes could be gained. Figure 5.31 displays the segmentation mask for fibrosis for SegFormer and UNet in the left column. The white areas in the segmentation mask stand for the part of the slide classified as "fibrosis" while black visualizes "non-fibrosis". Evidently, the UNet predicts more fibrosis while the SegFormer predicts only one area with high confidence

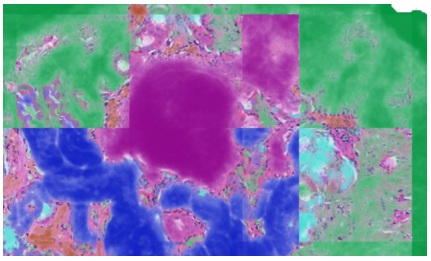


**Figure 5.29** Patching Characteristics in Prediction

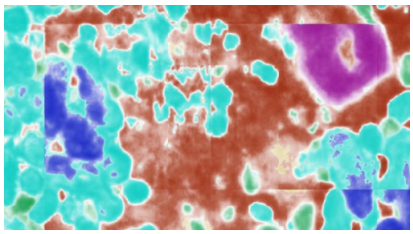


**Figure 5.30** Inconsistent Segmentation Prediction Across Patch Boundaries

and a few instances with low confidence, which are not seen in the segmentation mask due to their low transparency in the inference. Both models tend to capture the fibrosis on the border better than within the slide. The second column combines the two segmentation masks displaying the areas the two models predicted as fibrosis in white and in black, the areas both models agree on non-fibrosis. The gray areas are where the UNet predicts fibrosis, but the Transformer model does not. Consulting the annotations by two experts, seen in the third column of Image 5.31, and comparing them to the model's predictions, UNet comes closer to the annotations, while the predictions by SegFormer struggle to capture the annotations where two experts agree. The quantitative evaluation enables a good understanding of the shortcomings of SegFormer compared to UNet's segmentations and annotator opinion.

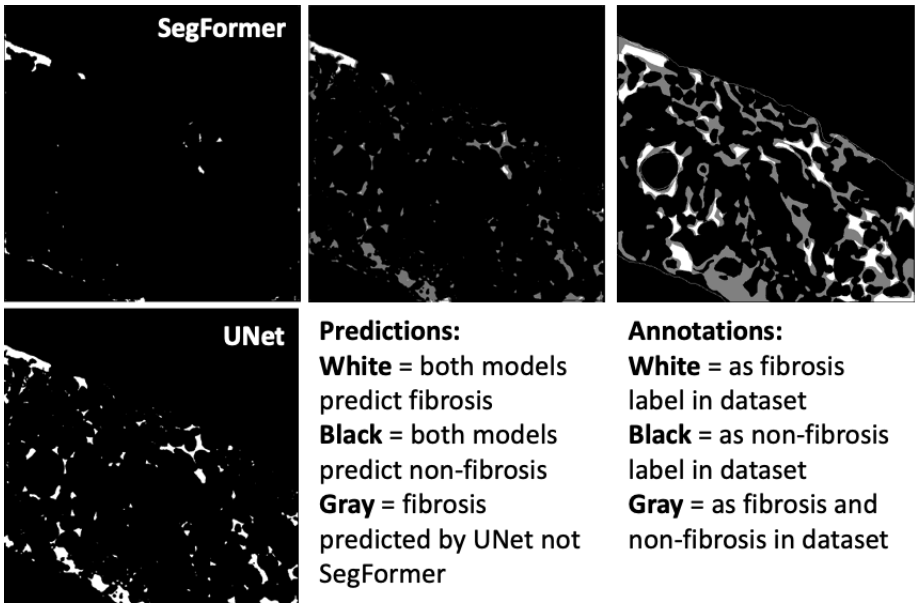



**Figure 5.31** Comparing Prediction of SegFormer and UNet

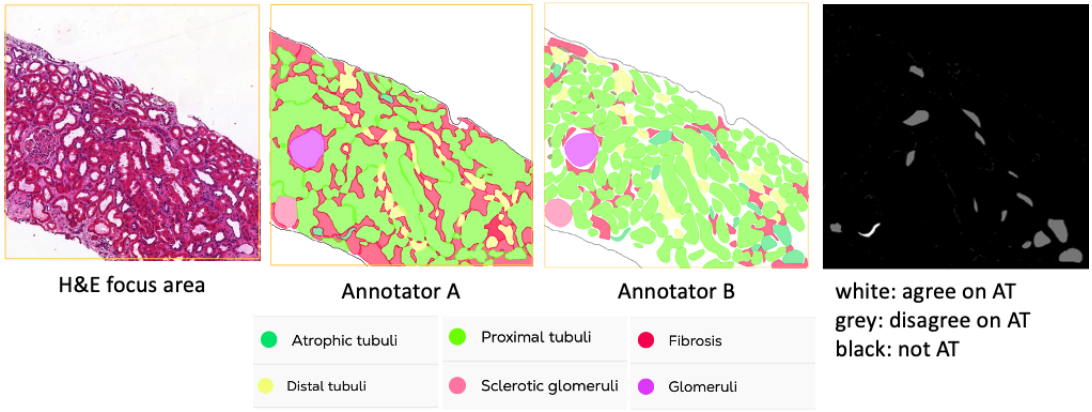



**Figure 5.32** Visualization of Inter-Annotator Variability

## 5.7 Annotator Variability

The quantity and quality of training data significantly impact the H&E segmentation performance of deep learning model. Annotated by pathologists following guidelines, training data introduce variability due to annotator knowledge, experience, and subjectivity, known as annotator variability [78]. Annotation discrepancies among experts affect precise segmentation borders and cell granularity. This variability's effect on deep learning models remains elusive due to task-specific context [78]. Annotator variability is divided into inter- and intra-annotator variability. The latter arises from human factors, while trained algorithms eliminate it. However, training on one expert's annotations may introduce bias. Inter-annotator variability measures the disagreement among annotators and requires low values to ensure training data consistency. The annotation process's complexity makes collecting diverse annotations challenging, necessitating bias assessment in deep learning models.

Addressing Transformers' potential in H&E segmentation compared to CNNs requires understanding bias and convergence in models trained on annotations by distinct experts. Transformer-specific issues stem from annotator variability, affecting training data and, consequently, model behavior:

- **Attention Mechanism Sensitivity:** Varied annotators may alter attention patterns, potentially constraining long-range relationship capture due to differing annotation patterns.
- **Long-Range Dependencies:** Transformer strengths lie in long-range dependency capture. Annotator variability might introduce inconsistencies, affecting the model's ability to recognize critical relationships between distant tokens.
- **Contextual Bias:** Transformers are sensitive to token sequence order. One annotation style may introduce contextual bias, while varied styles may hinder pattern recognition, affecting the model's comprehension of token context and dependencies [79].

Therefore, it is vital to evaluate how the introduced SegFormer architecture response to these sources of variability contrasted to the UNet model. Consequently, it is essential to comprehensively assess how the novel architecture's performance is impacted when faced with diverse annotation styles. Figure 5.32 provides a visual representation of a specific region of interest extracted from one slide of the hold-out dataset. Figure 5.32 visualizes the problem of inter-annotator variability. Here the focus area of one slide from the hold-out set is displayed. Within this visualization, the annotations offered by the two distinct pathologists and the associated similarity mask concerning atrophic tubuli (AT) are displayed. Utilizing this similarity mask reveals a notable difference between the two pathologists in AT annotations. Only one tubule is annotated by both experts, while the combined annotations result in a total of 15 ATs within this specific area of interest. To assess the model's reliability in real-world clinical contexts, examining its susceptibility to bias when trained on datasets annotated by a single annotator and its capacity to generalize across different annotator styles during testing is crucial. Subsequent experiments utilize three

separate test sets comprising unseen slides to quantitatively evaluate segmentation performance and gain insights into how the architectures handle the declared concerns of annotator variability. Further details about these three test sets can be found in Chapter 4.

For this experiment it was decided to use the parameters from the preceding experiments concerning optimizer, batch size, input size, network type and, augmentation techniques. The prior experiments showed that the learning rate and the $\lambda$ of the loss function strongly depend on the dataset size; these parameters were retested using the dataset 5.3 combining annotations by both experts. The auspicious learning rates 3e-05 and 5e-05 in combination with the $\lambda$ values 0.9 and 0.1 were tried. Figure 5.33 shows the Macro F1 performance of the different combinations. From these results, a learning rate of 3e-05 together with $\lambda = 0.9$ weighting for the combined loss function were inferred.

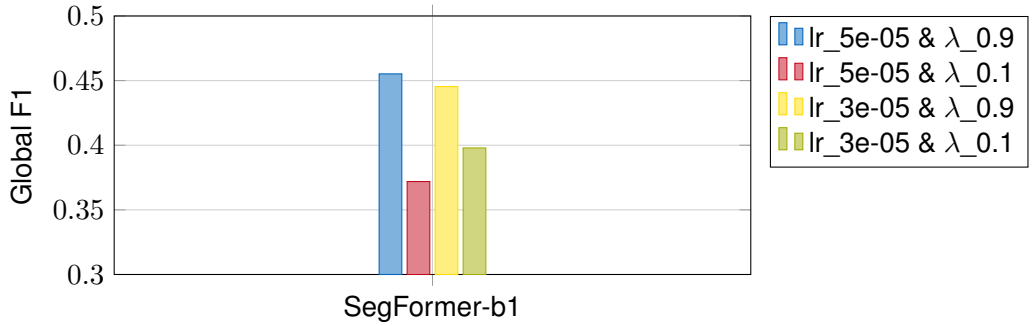



**Figure 5.33** Impact of Learning Rate, $\lambda$ of Combined Loss on Global Macro F1 of SegFormer-b1 with Dataset 5.3

The parameters for the UNet with ResNet18 backbone were set according to the parameters which are known to yield the best results for this dataset size. Figure 5.34 proves that both models were trained until convergence on the dataset including annotations from both experts.

## 5.8 Impact of Inter-Annotator Variability

This experiment provides insight into the models' ability to handle inter-annotator variability, answering sub-research question 7. To assess this, the trained models were evaluated on each of the three hold-out sets. As observed in previous experimental evaluations, the UNet architecture demonstrates superior performance across two of the three test sets, shown in Figure 5.38. However, the performance gap between the two is less pronounced than in earlier experiments. This outcome can be attributed to several factors. Firstly, the overall performance of both models is notably lower than that of previously trained models, likely due to the minimal size of the training dataset. Secondly, the test slides feature complex examples with intricate segmentation boundaries, particularly evident in the finer-grained annotations provided by Annotator B, which contributes to the lower test performance on slides annotated by B. Evaluating the models' performances across different datasets highlights significant variations in segmentation outcomes.

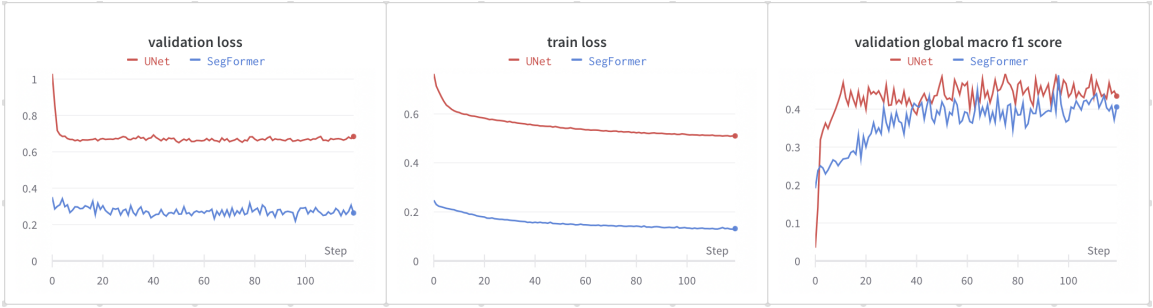



**Figure 5.34** Validation and Train loss of UNet and SegFormer on Dataset 5.3

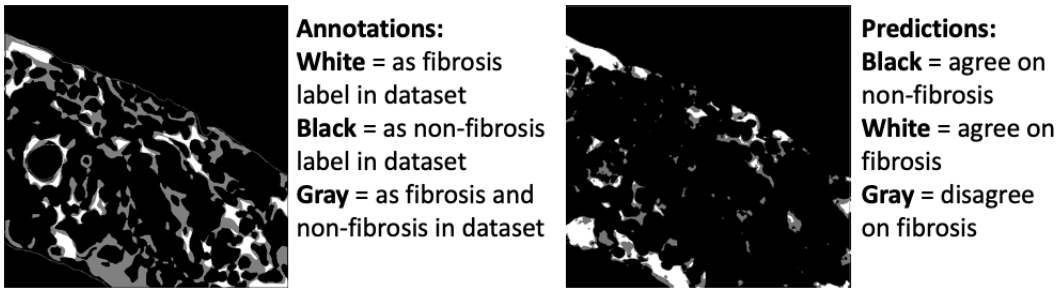



**Figure 5.35** Overlayed Fibrosis Annotation and Prediction

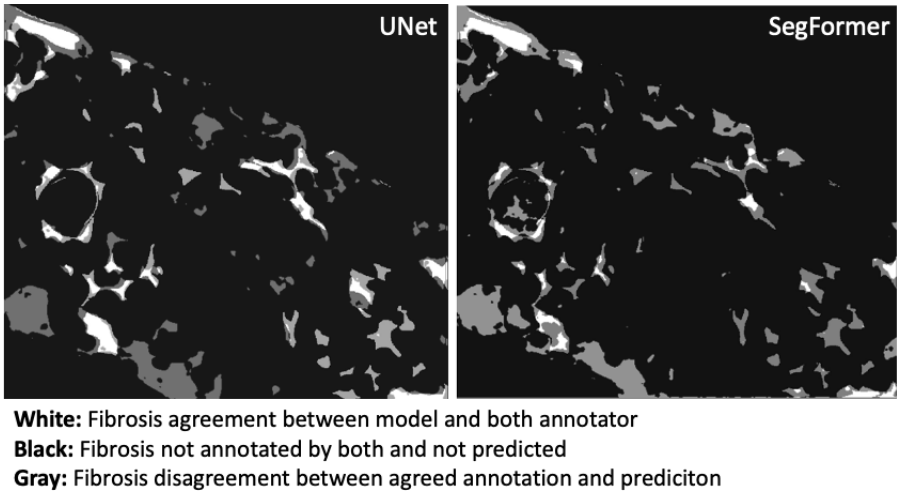



**Figure 5.36** Quantitative Comparison of Predictions on Agreed Annotation

This divergence implies that the model demonstrates enhanced generalization capabilities towards the annotations provided by Annotator A, as evidenced by the higher test scores achieved when evaluated on A's annotations. Conversely, the model exhibits comparatively diminished generalization when confronted with annotations by Annotator B. This discrepancy could be attributed to the finer granularity of annotations present within this dataset, which could challenge the model's segmentation process. Consequently, the assessment of segmentation performance is intricately influenced by the specific dataset they are evaluated on, including in any conclusion the consideration of training data characteristics. These observations emphasize the need for considering dataset characteristics and annotator variability when interpreting segmentation performance metrics. A qualitative evaluation of the two architectures provides a deeper understanding of the model's strategies in tackling inter-annotator variability within H&E segmentation. As depicted in Figure 5.35, on the left the fibrosis class annotations made by the two annotators are visualized, while the right side displays the overlayed predictions of the two architectures. The right channel map shows wide model agreement; however, compared to the annotations, notably intricate fibrosis annotations are absent from the predictions. Contrasting the predictions based on fibrosis agreement, as exhibited in Figure 5.36, it becomes apparent that the UNet encapsulates a notably larger number of fibrosis instances from the agreed annotations. Furthermore, when overlaying the predictions of each model onto the fused fibrosis annotations, as shown in Figure 5.37, it becomes evident that the SegFormer often predicts larger areas within regions with extensive annotations. However, it struggles to identify instances with thin boundaries or fine details accurately. This results in fine-grained instances being misclassified as non-fibrosis predictions. Furthermore, the SegFormer displays increased disparity between fused annotations and predictions, indicating that it tends to include false-positive fibrosis predictions.

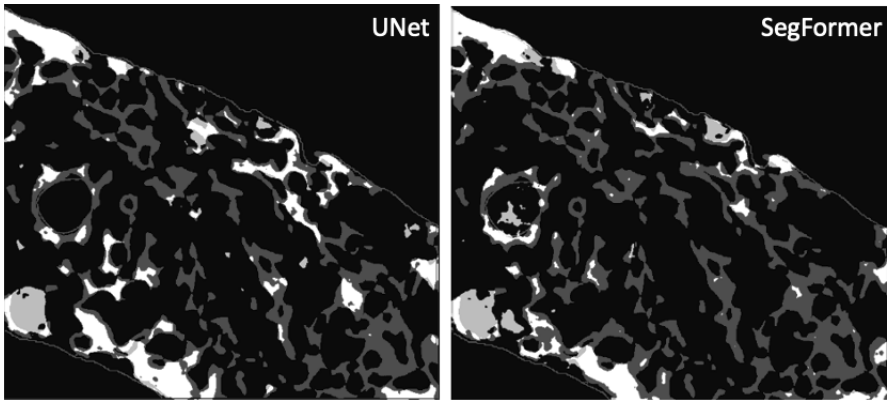



**Figure 5.37** Quantitative Comparison of Predictions on Joint Annotation

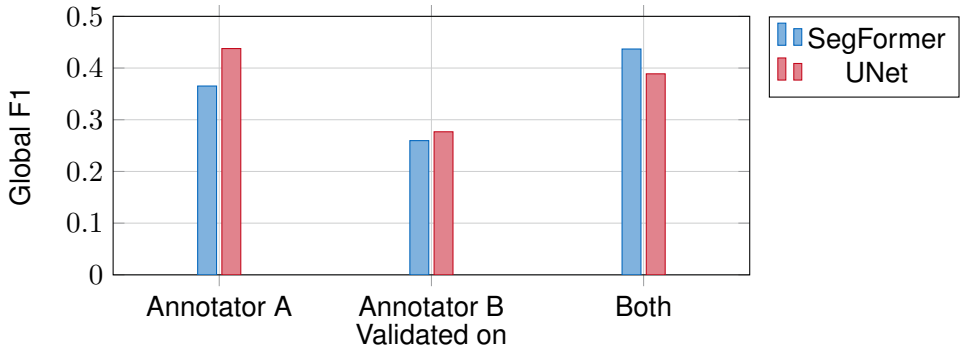



**Figure 5.38** Performance of Models Trained on Dataset 5.3

## 5.9 Impact of Single Style Annotations on Learned Biases

An additional experiment was conducted to assess potential biases arising from training the SegFormer solely on data annotated by a single annotator, addressing research question 8. A model displaying a pronounced inclination toward its training style, especially notable within limited training data, indicates a propensity for the architecture to overfit to that specific style. Such overfitting could reduce generalization capacity and diminish performance on diverse, unseen data. Illustrated in Figure 5.39 and 5.40, which compares the segmentation performance of both architectures on hold-out sets trained with annotator-specific datasets, no conclusive patterns emerge. The models trained on annotator B's annotations do not exhibit superior performance on the hold-out set annotated by B. Conversely, the models trained on annotator A's annotations outperform those trained on B's annotations. While these two models excel on A's hold-out set, they also perform strongly on B's and the combined test set. This suggests that while the models are more influenced by annotator A's style, they do not manifest a significant bias. This could be linked to the complexity of B's annotations, which feature fine-grained segmentation boundaries, as elaborated in Chapter 4. A quantitative analysis was executed on models trained exclusively on dataset 5.1, containing annotations solely by annotator A. A comparison of predictions to the ground truth annotations by this annotator, as depicted in Figure 5.41, reveals that neither model accurately predicts regions resembling the ground truth. Remarkably, neither model successfully predicts the "glomeruli" class, considered one of the more accessible classes for H&E segmentation and present in all training slides. This observation emphasizes the conclusion that the training dataset lacks the necessary density, and representation to yield meaningful predictive outcomes.

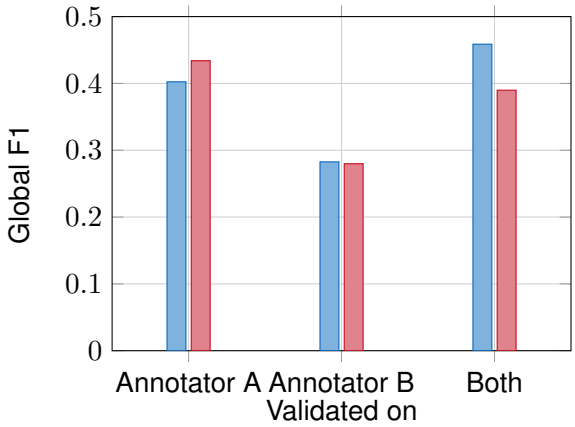



**Figure 5.39** Performance of Models Trained on Dataset 5.1

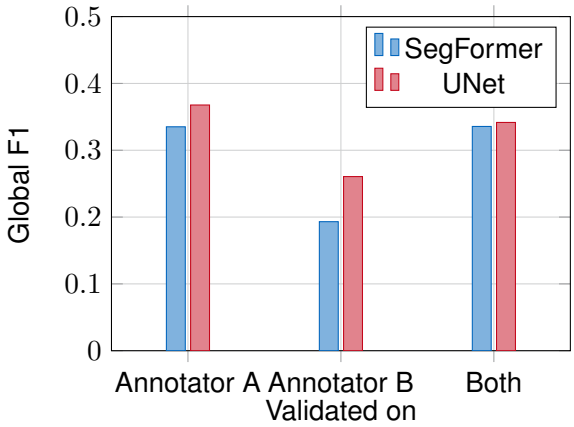



**Figure 5.40** Performance of Models Trained on Dataset 5.2

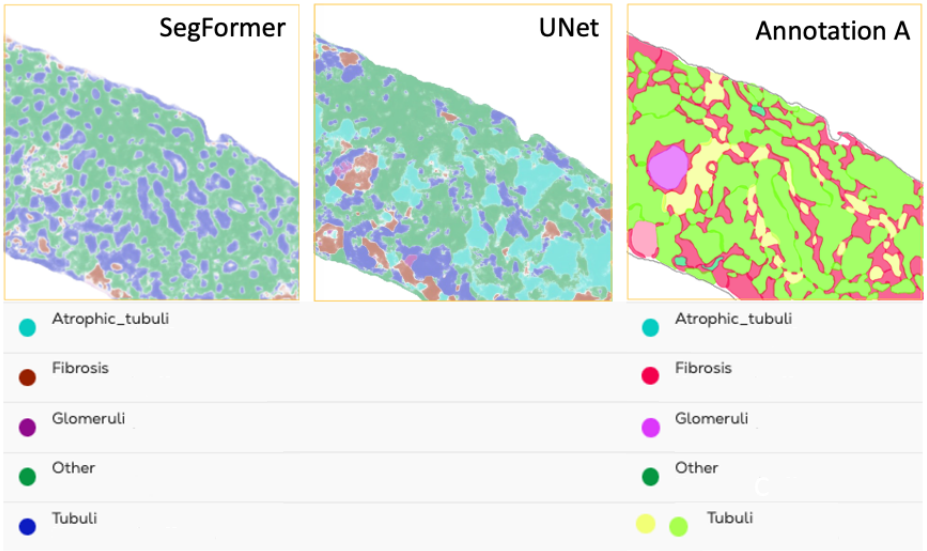



**Figure 5.41** Qualitative Analysis of Models Trained on Annotator A

# 6 Discussion

The objective was to assess the performance of SegFormer, compare it to a state-of-the-art UNet model, and provide insights into its suitability for integrating into a continuous deployment pipeline.

## 6.1 Contributions

### 6.1.1 Applied Pathological Pipeline Integration

Integrating the SegFormer architecture into the applied histopathological context was attainable through a series of adaptations to the original implementation. Through code modification, the model can seamlessly integrate into the deployment pipeline, can be seamlessly integrated into the deployment pipeline, facilitating architecture training with existing datasets, varied loss functions, optimizers, and class weighting strategies. This transition delineated the process of integrating SOTA models from open-source repositories into an existing histopathological AI pipeline. The learnings collected from this integration offer the potential to effectively evaluate advancements in computer vision techniques within the context of H&E segmentation, thus bridging the existing gap between conventional computer vision models and the specific requisites of medical histopathology segmentation.

### 6.1.2 Optimal Architecture and Parameter Initialization

The primary focus of this study revolved around the optimization of architecture and machine learning parameter initialization to achieve optimal segmentation performance. The integration of established machine learning parameters from advanced models, in conjunction with task-specific adaptations for SegFormer, facilitated the identification of effective parameter combinations. The weighting of loss function and choice of learning rate within a combined loss framework was determined to influence segmentation performance significantly. A higher weighted Cross Entropy loss displayed superior performance on multiple datasets. Only on a slimmed binary dataset did a model trained with a higher share of Dice loss outperform considering quantitative Macro F1. The results suggest that the choice of $\lambda$ value within the combined loss functions for H&E segmentation is not straightforward concerning the performance measured on Global Macro F1 and needs to be thoroughly tested for every dataset. The best performing learning rates are set to be within the space of ]1e-04, 3e-05[. While additional input augmentation during training and optimizer strategies did exhibit an impact, their effects were comparatively less pronounced, rendering them suitable for fine-tuning rather than initial setup. The investigation revealed that smaller network sizes sufficed, likely attributed to the relatively modest dataset size. Contrary to initial hypotheses, the empirical evidence indicated that larger input patches did not improve performance. However, it is essential to acknowledge that the extensive solution space limited the conclusive outcomes of this testing, primarily due to time and resource constraints.

### 6.1.3 Quantitative and Qualitative Comparison with UNet

A comprehensive comparison was conducted between SegFormer and a state-of-the-art UNet implementation in terms of both quantitative and qualitative performance aspects. Regarding quantitative performance metrics, SegFormer did not surpass the macro F1 score achieved by UNet. Notably, SegFormer's performance on more extensive datasets approached that of the state-of-the-art model, demonstrating its potential on larger training data. Clear trends emerged regarding recall and precision. While recall is comparable between the two architectures, the precision of SegFormer is notably distant from that of UNet.

SegFormer's higher recall than precision signifies its ability to effectively detect positive instances while also introducing numerous false positives. This points to a trade-off where the model's heightened sensitivity is counterbalanced by reduced specificity in identifying true cases, driven by an elevated false positive rate. Qualitatively, distinctions were observed in the uncertainty exhibited by the two models. Uncertainty in SegFormer predominantly emerged in complex regions, highlighting the architecture's capability to capture intricate details that may require expert interpretation for accurate assessment. Patchy predictions in SegFormer might be attributed to the pipeline being optimized for UNet inference, which makes comprehensive conclusions for whole-slide images inconclusive. Moreover, the prevalence of Cross-Entropy loss during training, which calculates pixel-wise loss, as opposed to Dice loss which accounts for the entire prediction region, could explain the observed patch patterns. Inconsistencies at boundaries were noted in SegFormer's predictions, which could also be linked to the incorrect overlapping of patches during inference. The pruned performance of the SegFormer architecture could be attributed to its emphasis on hierarchical feature extraction and attention mechanisms, potentially prioritizing broader contextual information over fine-grained local details. This could result in the loss of low-level details during processes such as the sequence reduction in self-attention calculations and the overlapping patch merging. The latter process could explain the uncertainty and missing of noncontinuous features between adjacent patches. Additionally, reducing Key and Value vectors in self-attention may overshadow the preservation of local details, explaining the model's missing capability to capture smaller-scale patterns.

Given the importance of localized information in histopathological image segmentation, the prevalent global attention focus in the self-attention mechanism might not be optimally suited for capturing small-scale patterns inherent in these images. The Decoder module might encounter challenges in recovering precise local information during upsampling due to the inherent reduction step in self-attention and the absence of skip connections.

Another potential elucidation for the inferior overall performance could be ascribed to the inherent characteristics of the input image. The high dimensionality and substantial image size intrinsic to H&E scans necessitate the utilization of image patching during the dataset building. This constrains the SegFormer's exposure to the entirety of the image during training. This limitation could impede the architecture's comprehensive exploitation of attention mechanisms.

### 6.1.4 Exploring Input Augmentation and Perturbation Effects

The study aimed to assess the robustness of SegFormer and UNet architecture by applying various input transformations and evaluating macro F1 score and accuracy at different perturbation levels. Although UNet achieved higher macro F1 scores, it struggled to generalize under different levels of input transformations. The results suggest that SegFormer exhibited greater overall robustness to input perturbations, making it a preferable choice when dealing with noisy or imperfect input data. Additional robustness experiments concerned the impact of loss function on the SegFormer architecture, concluding that the share of loss function within the combined loss during the training process does not have a high impact on the overall generalizability of the model. The experiments provided valuable insights into the generalizability of both models for real-world H&E segmentation applications.

### 6.1.5 Inter-Annotator Variability Assessment

The evaluation of inter-annotator variability concerning segmentation performance and generalizability revealed several key insights. Despite the lack of significant quantitative differences attributed partly to the limited dataset size, distinct patterns emerged between the predictions of SegFormer and UNet. Notably, SegFormer demonstrated a tendency to predict coarser regions including annotations from either annotator, whereas UNet exhibited a preference for making predictions where the annotators agree. This prompts the query of whether this phenomenon contributes to SegFormer's previously noted characteristics of higher recall and lower precision during training by capturing positive instances possibly labeled inconsistently by the annotators. However, in parallel, SegFormer exhibited elevated uncertainty, particularly in areas where annotators disagreed. This suggests that despite its tendency to overpredict in some

instances, the architecture is aware of potential inaccuracies, as evidenced by uncertainty. The analysis of bias remained inconclusive due to the limited dataset and the relatively low interpretability of the predictions. Nonetheless, an emerging hypothesis from this experimentation suggests that both architectures might learn more efficiently from the style of one annotator over the other. This implies that the annotator's style substantial influence on deep learning algorithms, irrespective of the employed architecture. Further investigations with larger datasets are warranted to validate and expand upon these findings.

### 6.1.6 Applicability of SegFormer in Applied Pathology

From an economic perspective, the SegFormer model demonstrates comparable resource demands for the two smaller network sizes. However, its suitability for H&E segmentation is debatable due to the considerable demands for training data and extended training duration until convergence. The limited availability of training data and the requirement for swift re-training to accommodate continuous integration of new datasets, such as corrective and novel samples, adds to the challenges. Moreover, the model's lower performance diminishes its viability for adoption in applied pathology scenarios. The aspects of uncertain regions and high robustness to input perturbation require deeper consideration to leverage their potential benefits and facilitate iterative improvements.

## 6.2 Scope and Limitations

The scope of this research study encompasses a comprehensive exploration of the SegFormer architecture's performance and applicability in the context of histopathological image segmentation. This study aims to assess the architecture's strengths, weaknesses, and potential contributions to medical image analysis through a series of systematic experiments and analyses. By examining quantitative and qualitative aspects and comparing those to a reference CNN-based model with a ResNet18 backbone, the research seeks to uncover the architecture's capabilities and limitations in capturing intricate structures within histopathological images. While the study has provided valuable insights, certain limitations should be acknowledged. The findings' portability and reproducibility might be constrained by their dependence on an internal pipeline and specific dataset. Additionally, the scope of parameter exploration was somewhat constrained due to the vast parameter space and the focus on commonly used functions. Due to feasibility constraints, highly specialized parameters, such as customized loss functions or optimizers, were not extensively tested. These limitations, while present, do not diminish the significance of the study's contributions, and they underscore potential avenues for further research and refinement. Future research could address these limitations by conducting more extensive parameter exploration, utilizing diverse and publicly available datasets, and finding more efficient ways to perform inference to ensure broader applicability and reduce biases in the analysis.

# 7 Conclusion

The objective of this study was to adapt the SegFormer network architecture for histopathological semantic segmentation and compare its performance to a reference UNet implementation. This Transformer-based architecture addresses the issue of absent long-range dependencies and lack of robustness in Encoder-Decoder CNN architectures by incorporation attention modules. The quantitative comparison shows a similar performance of the two architectures when applied to larger dataset sizes. However, on smaller datasets, the Transformer model encountered difficulties surpassing the baseline performance, a recurrent behavior observed in Transformer architectures. The qualitative assessment revealed discrepancies in accurate fine structure segmentation, wherein the SegFormer model displayed pronounced uncertainties for complex structures and an overarching deficiency in achieving precise segmentation at a fine-grained level. This is most likely due to the absence of skip connections, causing the Decoder to encounter challenges in accurately restoring detailed local information during the upsampling process. Moreover, employing a single patch from the whole H&E slide image as input might impede the optimal utilization of its global attention mechanism.

Furthermore, the architectures display similar quantitative performance concerning inter-annotator variability, suggesting the Transformer's ability to accommodate diverse annotation styles. Notably, the qualitative predictions of the models differ; UNet's predictions tend to exhibit more fine details while effectively encompassing annotations where both experts concur. Conversely, SegFormer's predictions display coarser features over all annotations, displaying a deficiency in accurately covering the annotations where consensus exists between experts. Due to limited labels, exploring biases from single-annotator training produced inconclusive results for both architectures.

Lastly, SegFormer exhibits notable resilience in the face of input fluctuations, thereby substantiating its capability for generalization, a characteristic attributed to the incorporation of the self-attention mechanism. This study underscores the significance of automating histopathological segmentation and emphasizes the necessity for in-depth research into computationally efficient Transformer-based methods to surmount the limitations of existing solutions. Future work should investigate the incorporation of holistic information from entire histopathological slides into the training paradigm. This holds particular significance in maximizing the inherent benefits of self-attention when evaluating the performance of any Transformer-based model for the task of H&E segmentation.

# Bibliography


[1] R. Brown, "What is the difference between image segmentation and classification in image processing?" https://medium.com/cogitotech/what-is-the-difference-between-image-segmentation-and-classification-in-image-processing-303d1f660626#:~:text=Image%20segmentation%20is%20the%20process,accurate%20view%20of%20an%20image, 2019, accessed: 2023-08-2.

[2] A. Valizadeh, M. Shariatee *et al.*, "The progress of medical image semantic segmentation methods for application in covid-19 detection," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[3] "Medical image segmentation," https://www.synopsys.com/glossary/what-is-medical-image-segmentation.html, accessed: 2023-04-24.

[4] B. Aldughayfiq, F. Ashfaq, N. Jhanjhi, and M. Humayun, "Yolov5-fpn: A robust framework for multi-sized cell counting in fluorescence images," *Diagnostics*, vol. 13, no. 13, p. 2280, 2023.

[5] N. Sharma and L. M. Aggarwal, "Automated medical image segmentation techniques," *Journal of medical physics*, vol. 35, no. 1, pp. 3–14, 2010.

[6] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in mri images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.

[7] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. De Vries, M. J. Benders, and I. Išgum, "Automatic segmentation of mr brain images with a convolutional neural network," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1252–1261, 2016.

[8] C. Cernazanu-Glavan and S. Holban, "Segmentation of bone structure in x-ray images using convolutional neural network," *Adv. Electr. Comput. Eng*, vol. 13, no. 1, pp. 87–94, 2013.

[9] M. Silveira, J. C. Nascimento, J. S. Marques, A. R. Marçal, T. Mendonça, S. Yamauchi, J. Maeda, and J. Rozeira, "Comparison of segmentation methods for melanoma diagnosis in dermoscopy images," *IEEE journal of selected topics in signal processing*, vol. 3, no. 1, pp. 35–45, 2009.

[10] R. Chrástek, M. Wolf, K. Donath, H. Niemann, D. Paulus, T. Hothorn, B. Lausen, R. Lämmer, C. Y. Mardin, and G. Michelson, "Automated segmentation of the optic nerve head for diagnosis of glaucoma," *Medical image analysis*, vol. 9, no. 4, pp. 297–314, 2005.

[11] V. Fortunati, R. F. Verhaart, F. van der Lijn, W. J. Niessen, J. F. Veenland, M. M. Paulides, and T. van Walsum, "Tissue segmentation of head and neck ct images for treatment planning: a multiatlas approach combined with intensity modeling," *Medical physics*, vol. 40, no. 7, p. 071905, 2013.

[12] "Medical image segmentation," https://paperswithcode.com/task/medical-image-segmentation, accessed: 2023-04-24.

[13] J. D. Bancroft and C. Layton, "The hematoxylins and eosin," *Bancroft's theory and practice of histological techniques*, vol. 7, pp. 173–186, 2012.

[14] L. Bonaldi, A. Pretto, C. Pirri, F. Uccheddu, C. G. Fontanella, and C. Stecco, "Deep learning-based medical images segmentation of musculoskeletal anatomical structures: A survey of bottlenecks and strategies," *Bioengineering*, vol. 10, no. 2, p. 137, 2023.

[15] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual review of biomedical engineering*, vol. 19, pp. 221–248, 2017.

[16] F. Renard, S. Guedria, N. D. Palma, and N. Vuillerme, "Variability and reproducibility in deep learning for medical image segmentation," *Scientific Reports*, vol. 10, no. 1, pp. 1–16, 2020.

[17] J. Latif, C. Xiao, A. Imran, and S. Tu, "Medical imaging using machine learning and deep learning algorithms: a review," in *2019 2nd International conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE, 2019, pp. 1–5.

[18] J. Xu, X. Luo, G. Wang, H. Gilmore, and A. Madabhushi, "A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images," *Neurocomputing*, vol. 191, pp. 214–223, 2016.

[19] A. Madabhushi and G. Lee, "Image analysis and machine learning in digital pathology: Challenges and opportunities," *Medical image analysis*, vol. 33, pp. 170–175, 2016.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[21] Z. Zeng, W. Xie, Y. Zhang, and Y. Lu, "Ric-unet: An improved neural network based on unet for nuclei segmentation in histology images," *Ieee Access*, vol. 7, pp. 21 420–21 428, 2019.

[22] Y. Chen, T. Li, Q. Zhang, W. Mao, N. Guan, M. Tian, H. Yu, and C. Zhuo, "Ant-unet: Accurate and noise-tolerant segmentation for pathology image processing," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 18, no. 2, pp. 1–17, 2021.

[23] "aignostics gmbh," https://www.aignostics.com, 2023.

[24] "Pathai," https://www.pathai.com, 2023.

[25] C. S. Perone and J. Cohen-Adad, "Promises and limitations of deep learning for medical image segmentation," *J Med Artif Intell*, vol. 2, no. 1, pp. 1–2, 2019.

[26] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3286–3295.

[27] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[28] L. Zhang, X. Wang, D. Yang, T. Sanford, S. Harmon, B. Turkbey, B. J. Wood, H. Roth, A. Myronenko, D. Xu *et al.*, "Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2531–2540, 2020.

[29] A. Gillioz, J. Casas, E. Mugellini, and O. Abou Khaled, "Overview of the transformer-based models for nlp tasks," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2020, pp. 179–183.

[30] "Transformers for text classification," https://blog.paperspace.com/transformers-text-classification/, 2023.

[31] "Sentiment analysis using transformers," https://www.analyticsvidhya.com/blog/2022/02/sentiment-analysis-using-transformers/, 2023.

[32] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *Advances in neural information processing systems*, vol. 32, 2019.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[34] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[35] "What is a transformer model," https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/, 2023.

[36] "How transformers work," https://towardsdatascience.com/transformers-141e32e69591, 2023.

[37] G. Andrade-Miranda, V. Jaouen, V. Bourbonne, F. Lucia, D. Visvikis, and P.-H. Conze, "Pure versus hybrid transformers for multi-modal brain tumor segmentation: a comparative study," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 1336–1340.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[39] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[40] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[41] "convolutional neural network (cnn)," https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network#:~:text=A%20CNN%20is%20a%20kind,the%20network%20architecture%20of%20choice., 2023.

[42] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[43] "Vision transformers or convolutional neural networks? both!" https://towardsdatascience.com/vision-transformers-or-convolutional-neural-networks-both-de1a2c3c62e4, 2023.

[44] "What is inductive bias in machine learning," https://www.baeldung.com/cs/ml-inductive-bias, 2023.

[45] J. Liang, C. Yang, M. Zeng, and X. Wang, "Transconver: transformer and convolution parallel network for developing automatic brain tumor segmentation in mri images," *Quantitative Imaging in Medicine and Surgery*, vol. 12, no. 4, p. 2397, 2022.

[46] "Vision transformers vs. convolutional neural networks," https://medium.com/@faheemrustamy/vision-transformers-vs-convolutional-neural-networks-5fe8f9e18efc, 2023.

[47] Y. Bai, J. Mei, A. L. Yuille, and C. Xie, "Are transformers more robust than cnns?" *Advances in neural information processing systems*, vol. 34, pp. 26 831–26 843, 2021.

[48] U. Zidan, M. M. Gaber, and M. M. Abdelsamea, "Swincup: Cascaded swin transformer for histopathological structures segmentation in colorectal cancer," *Expert Systems with Applications*, vol. 216, p. 119452, 2023.

[49] F. Hörst, M. Rempe, L. Heine, C. Seibold, J. Keyl, G. Baldini, S. Ugurel, J. Siveke, B. Grünwald, J. Egger *et al.*, "Cellvit: Vision transformers for precise cell segmentation and classification," *arXiv preprint arXiv:2306.15350*, 2023.

[50] K. He, C. Gan, Z. Li, I. Rekik, Z. Yin, W. Ji, Y. Gao, Q. Wang, J. Zhang, and D. Shen, "Transformers in medical image analysis: A review," *Intelligent Medicine*, 2022.

[51] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[52] F. Fan, L. Ritschl, M. Beister, R. Biniazan, B. Kreher, T. M. Gottschalk, S. Kappler, and A. Maier, "Simulation-driven training of vision transformers enabling metal segmentation in x-ray images," *arXiv preprint arXiv:2203.09207*, 2022.

[53] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images," in *International MICCAI Brainlesion Workshop*. Springer, 2021, pp. 272–284.

[54] H. Yang and D. Yang, "Cswin-pnet: A cnn-swin transformer combined pyramid network for breast lesion segmentation in ultrasound images," *Expert Systems with Applications*, vol. 213, p. 119024, 2023.

[55] J. D. Hipp, A. Fernandez, C. C. Compton, and U. J. Balis, "Why a pathology image should not be considered as a radiology image," *Journal of pathology informatics*, vol. 2, 2011.

[56] M. Zaitsev, J. Maclaren, and M. Herbst, "Motion artifacts in mri: A complex problem with many partial solutions," *Journal of Magnetic Resonance Imaging*, vol. 42, no. 4, pp. 887–901, 2015.

[57] D. Karimi, H. Dou, and A. Gholipour, "Medical image segmentation using transformer networks," *IEEE Access*, vol. 10, pp. 29 322–29 332, 2022.

[58] L. Barisoni, K. J. Lafata, S. M. Hewitt, A. Madabhushi, and U. G. Balis, "Digital pathology and computational image analysis in nephropathology," *Nature Reviews Nephrology*, vol. 16, no. 11, pp. 669–685, 2020.

[59] A. Taha, "High resolution images and efficient transformers," https://ahmdtaha.medium.com/high-resolution-images-and-efficient-transformers-92db6f8803f7, 2023, accessed: 2023-08-3.

[60] P. Kleczek, J. Jaworek-Korjakowska, and M. Gorgon, "A novel method for tissue segmentation in high-resolution h&e-stained histopathological whole-slide images," *Computerized Medical Imaging and Graphics*, vol. 79, p. 101686, 2020.

[61] P. Bándi, R. van de Loo, M. Intezar, D. Geijs, F. Ciompi, B. van Ginneken, J. van der Laak, and G. Litjens, "Comparison of different methods for tissue segmentation in histopathological whole-slide images," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, 2017, pp. 591–595.

[62] K. Ikromjanov, S. Bhattacharjee, Y.-B. Hwang, R. I. Sumon, H.-C. Kim, and H.-K. Choi, "Whole slide image analysis and detection of prostate cancer using vision transformers," in *2022 international conference on artificial intelligence in information and communication (ICAIIC)*. IEEE, 2022, pp. 399–402.

[63] Z. Li, Y. Cong, X. Chen, J. Qi, J. Sun, T. Yan, H. Yang, J. Liu, E. Lu, L. Wang *et al.*, "Vision transformer-based weakly supervised histopathological image analysis of primary brain tumors," *IScience*, vol. 26, no. 1, 2023.

[64] B. Guo, J. Jonnagaddala, H. Zhang, and X. S. Xu, "Predicting microsatellite instability and key biomarkers in colorectal cancer from h&e-stained images: Achieving sota with less data using swin transformer," *arXiv e-prints*, pp. arXiv–2208, 2022.

[65] H. Chen, C. Li, G. Wang, X. Li, M. M. Rahaman, H. Sun, W. Hu, Y. Li, W. Liu, C. Sun *et al.*, "Gashis-transformer: A multi-scale visual transformer approach for gastric histopathological image detection," *Pattern Recognition*, vol. 130, p. 108827, 2022.

[66] Y. Liu, Y. Zhu, Y. Xin, Y. Zhang, D. Yang, and T. Xu, “Mestrans: Multi-scale embedding spatial transformer for medical image segmentation,” *Computer Methods and Programs in Biomedicine*, vol. 233, p. 107493, 2023.

[67] qubvel, “unet.py,” https://github.com/qubvel/segmentation_models/blob/master/segmentation_models/models/unet.py, 2020.

[68] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, “Scaling local self-attention for parameter efficient visual backbones,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 894–12 904.

[69] J. Brownlee. (2019) A gentle introduction to dropout for regularizing deep neural networks. https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/. 14.08.2023.

[70] sgugger, “transformers/src/transformers/models/segformer/modeling_tf_segformer.py,” https://github.com/huggingface/transformers/tree/main/src/transformers/models/segformer, 2023.

[71] “Documentation for tf.keras.model,” https://www.tensorflow.org/api_docs/python/tf/keras/Model, 2023.

[72] “Hydra open-source python framework,” https://hydra.cc/docs/intro/, 2023.

[73] “Imfusion suite,” https://www.imfusion.com, 2023.

[74] “Googlecloud,” https://cloud.google.com/compute/docs/gpus, 2023.

[75] “Documentation for prefect,” https://docs.prefect.io/2.11.3/, 2023.

[76] “Documentation for ray,” https://www.ray.io, 2023.

[77] Rocketknight1, “transformers/src/transformers/models/segformer/modeling_tf_segformer.py,” https://github.com/huggingface/transformers/blob/v4.30.0/src/transformers/models/segformer/modeling_tf_segformer.py#L750.

[78] Vădineanu, D. M. Pelt, O. Dzyubachyk, and K. J. Batenburg, “An analysis of the impact of annotation errors on the accuracy of deep learning for cell segmentation,” in *Proceedings of The 5th International Conference on Medical Imaging with Deep Learning*, 2022, pp. 1251–1267.

[79] S. Sudhakar, V. Prabhu, A. Krishnakumar, and J. Hoffman, “Mitigating bias in visual transformers via targeted alignment,” *arXiv preprint arXiv:2302.04358*, 2023.