

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Enhancing Automatic Polyp Segmentation Generalization Using Self-Supervised Learning

Carla Sofia Barros Monteiro



Master in Biomedical Engineering

Supervisors:

Valentina Corbetta

Dr. Wilson Silva

Dr. Luís Teixeira

July 21, 2025

Enhancing Automatic Polyp Segmentation Generalization Using Self-Supervised Learning

Carla Sofia Barros Monteiro

Master in Biomedical Engineering

July 21, 2025

UN Sustainable Development Goals

This chapter provides an analysis on how the presented work may contribute to the United Nations Sustainable Development Goals (SDGs). In total, this includes 17 goals aiming to address current world's pressing challenges, such as lack of equality, climate change, poverty, etc.

We can frame this dissertation in the following SDG goals:

SDG 3 Ensure healthy lives and promote well-being for all at all ages.

SDG 8 Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all.

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation.

SDG 12 Ensure sustainable consumption and production patterns.

Table 1 presents an overview of the specific goals that this dissertation addresses among the above SDG goals.

Table 1: Sustainable Development Goals addressed by the work developed in this dissertation, specifying the contributions and targets it addresses.

SDG	Target	Contribution	Performance Indicators and Metrics
3	4	Improving polyp segmentation generalizability moves AI segmentation systems closer to effective implementation in clinics, which can be crucial to reduce the current rate of missed polyp detections. By also improving the performance of these models, we can have a better delineation of the lesion, thus providing a better guidance for extraction.	The referred contributions could be measured by an increase in the rate of early polyp identification and extraction, followed by a decrease in colorectal cancer-related deaths.

Continued on next page

Table 1 – continued from previous page

SDG	Target	Contribution	Performance Indicators and Metrics
	8	Improving polyp segmentation performance and generalizability can contribute to make colonoscopy procedures more efficient, affordable, and accurate. By integrating AI-driven tools for polyp segmentation, healthcare systems may reduce costs, enhance accessibility to quality diagnostics, and improve patient outcomes, particularly in underserved or resource-limited settings.	This can be measured by a reduction in the cost per patient diagnosis, as well as the increase in the number of people diagnosed early due to more affordable or scalable diagnostic technology.
	b	By improving polyp detection, this work may inform the development of more targeted treatment and prevention strategies for non-communicable diseases. Furthermore, by increasing generalizability, we also make these strategies more applicable and adaptive to different data distributions, facilitating its use in further research.	This can be measured in a reduction of the adaptation time and by a higher use as base technology to develop clinical strategies for non-communicable disease treatment.
8	2	The integration of automatic polyp detection systems can enhance productivity in healthcare, leading to faster diagnoses and better resource allocation.	This can be measured by diagnosis speed, and reduction of the amount of medical resources consumed per diagnosis (for example, reducing the need for further manual image analysis).
	3	The development and implementation of AI-based healthcare technologies could promote innovation in the medical field, contributing to economic growth and the creation of high-skill jobs in tech-driven healthcare sectors.	This can be measured by increased job creation and economic growth that follows the integration of these solutions (for example, by promoting the creation of new startups focused on AI-enhanced diagnostic strategies).
9	5	By advancing the use of deep learning for polyp segmentation, this work directly contributes to technological innovation in the healthcare sector, improving diagnostic methods and treatment pathways.	This can be measured by the rate of adoption of these models in hospitals and clinics, as well as the further scientific research promoted by the contributions of this dissertation.

Continued on next page

Table 1 – continued from previous page

SDG	Target	Contribution	Performance Indicators and Metrics
12	6	By automating polyp detection, we can contribute to reducing resource consumption and costs.	This can be measured through resource efficiency in medical diagnostics, cost reduction in diagnostic practices, and environmental impact.

Abstract

Automatic polyp segmentation using deep learning models has great potential to reduce missed polyp detections and improve colorectal cancer outcomes. However, these models often pick up patterns from the training data that are unrelated to the polyps themselves. For example, they might focus on specific annotation styles introduced by the imaging device or on demographic and anatomical variations across different populations. These factors can influence the model predictions without providing any meaningful information about the lesions, thus making them biased and leading to poor performance on new data. Consequently, such models struggle when faced with distribution shifts, which hinders their proper integration into clinical settings and limits their potential to improve diagnostic accuracy.

This dissertation explores methods to address these biases, aiming to enhance the reliability of deep learning models across diverse data distributions and to promote their effective use in clinical applications. The primary goal is to enhance polyp segmentation by increasing feature robustness, encouraging models to focus on meaningful polyp-related features, while reducing reliance on irrelevant cues. This is explored through the application of color invariance mechanisms and self-supervised pretraining, which has been shown to produce more meaningful feature representations when compared to label-based methods. The robustness of these features is further enhanced through the self-attention mechanisms used in Vision Transformers. By integrating these methods, we propose a simple yet effective architecture that leverages high-quality learned features to improve current state-of-the-art generalization performance.

In sum, the contributions of this thesis aim to improve the performance of polyp segmentation models across varying data distributions, bringing them closer to practical integration in clinical settings.

Keywords: Deep Learning, Generalization, Medical Bias, Vision Transformers, Self-Supervised Learning, Polyp Segmentation, Domain Shift, Data Variability

Acknowledgments

This dissertation was developed in collaboration with the Trustworthy AI group from the Radiology Department of the Netherlands Cancer Institute (NKI) in Amsterdam and AIT4LIFE at Utrecht University (UU). I am grateful to everyone from both groups for the warm welcome and insightful discussions, particularly to my office mates, whom I can now call friends. I am thankful for the many people I met who made my time in the Netherlands memorable and ordinary workdays fun. I would especially like to thank my supervisors, Wilson Silva and Valentina Corbetta, for listening to my questions, supporting my ideas and side quests, and calming my always-too-many worries. Their support throughout my master's thesis kept me motivated, and through our discussions and my involvement in their work, I am now considering pursuing a PhD as my next life step.

I would like to thank my family, who have always believed in me and supported me throughout my academic journey.

I also wish to express my gratitude to my friends for always listening to my worries, making me laugh in tough times, and offering constant support. Finally, a big thank you to my boyfriend, who's always been my biggest supporter through all my challenges and successes.

“There are no shortcuts in evolution”

Louis D. Brandeis

Contents

1	Introduction	1
1.1	Background	1
1.2	Medical Bias and Generalization Issues	3
1.3	Motivation	3
1.4	Structure of the Dissertation	4
2	Literature Review: Polyp Segmentation	5
2.1	Convolutional Neural Network-Based Methods	6
2.2	Pure Transformer-Based Methods	9
2.3	Hybrid Methods	12
2.4	Current Challenges	13
2.5	Relevance to this Dissertation	15
3	Literature Review: Self-Supervised Learning	16
3.1	Key Concepts	16
3.2	Image-Level vs Dense Self-Supervised Learning	20
3.2.1	Image-Level Self-Supervised Learning	20
3.2.2	Dense Self-supervised Learning	23
3.3	Current Challenges	24
3.4	Relevance to this Dissertation	25
4	Preliminary Experimental Work	26
4.1	Dataset: PolypGen	27
4.1.1	Chromoendoscopy	28
4.2	Baseline Models	29
4.2.1	U-Net	29
4.2.2	DeepLabv3+	29
4.3	Evaluation Metrics for Segmentation Problems	30
4.4	Assessing the Chromoendoscopy Bias	31
4.4.1	Methodology	31
4.4.2	Results and Discussion	33
4.5	Impact of Color Preprocessing on Baseline Model Performance	37
4.5.1	Data Preparation	37
4.5.2	Methodology	37
4.5.3	Results and Discussion	38
4.6	Conclusions	42

5	Color Invariance Methods	43
5.1	CycleGAN for Paired Dataset Creation	43
5.1.1	Implementation Details	45
5.1.2	Qualitative Results	45
5.2	Color Transfer	47
5.3	Random Color Erasing	48
5.4	Ablation of the Different Color Invariance Techniques	49
5.5	Conclusions	50
6	Advancing Segmentation Performance with Self-Supervised ViT Representations	52
6.1	Analysis of Key Features from DINO-ViT	53
6.1.1	Semantic Correspondences	53
6.1.2	Evolution of Key Descriptors Across Layers	55
6.1.3	Similar Image Retrieval	56
6.2	Part co-segmentation	59
6.2.1	Ablation Study on Group Specificity	60
6.2.2	Ablation Study on Extractor Settings	64
6.2.3	Quantitative Results	64
6.2.4	Evaluation of the Multiple Sisters Approach Against a Fully Supervised Baseline	70
6.2.5	Discussion	72
6.3	KeyViTSeg: Supervised Polyp Segmentation	74
6.3.1	Ablation Study on the Extractor Configurations	75
6.3.2	Generalization Tests	75
6.3.3	Statistical Evaluation	79
6.3.4	Computational Cost Analysis	81
6.3.5	Qualitative Results	81
6.4	Conclusions	83
7	Conclusions	85
	References	87
A	Segmentation with Self-Supervised ViT Representations Results	95
A.1	Part-co-segmentation results	95
A.1.1	Global	95
A.1.2	Clusters	96
A.1.3	Classes	100
A.1.4	Sisters	102

List of Figures

1.1	Representation of two of the most common types of polyps, followed by their corresponding segmentation masks [1]. In (a) we show examples of pedunculated polyps, while in (b) we show sessile polyps.	1
1.2	Representation of the two polyp types according to their growth pattern. In (a) we show adenomatous, while in (b) we show serrated [2].	2
2.1	U-Net Architecture [3]	6
2.2	UM-Net architecture [4].	8
2.3	Vision Transformer (ViT) architecture. (a) Linear embedding and transformer encoder; (b) Multi-head self-attention (MSA); (c) Scaled dot-product attention, (d) Final linear transformations and output [5].	10
2.4	Vision Transformer (ViT) attention layer representation. Self-attention is computed when $Y=X$ [6].	11
3.1	Pretext tasks and downstream tasks. The model is trained on a pretext task using data-extracted labels (in other words, in a self-supervised way). The knowledge acquired by the model after being trained on this pretext task is transferred to the downstream (or target) task. This figure was inspired in a similar representation [7].	17
3.2	Pretext tasks used to enhance model training: a) Colorizing: Inferring colors for grayscale images; b) Inpainting: Filling in missing parts of an image; c) Jigsaw Puzzle: Rearranging shuffled image patches; d) Predicting Relative Position: Identifying the arrangement of image patches; e) Predicting Rotation Angle: Determining the angle of image rotation. Inspired by a similar representation [8].	18
3.3	Contrastive learning simplified. Representations of positive pairs of images (augmentations belonging to the same class) are attracted whereas representations of negative pairs of images (belonging to different classes) are repelled. Inspired by a similar representation [9].	19
3.4	Self-distillation with no labels for one pair of views [10]. x_1 and x_2 represent a pair of views of the same image.	24
4.1	Visual examples of some of the data biases represented in the PolypGen dataset, including different light conditions, the inclusion of medical instruments in the images, labels, and chromoendoscopy data.	26
4.2	Examples of image/mask pairs for each center of the PolypGen dataset.	27
4.3	Distribution of chromoendoscopy and regular colonoscopy single-frame samples across centers. Blue or "chromo" is chromoendoscopy, whereas pink or "non-chromo" is regular colonoscopy.	28
4.4	DeepLabV3+ Architecture [11]	29

4.5	A 3×3 depthwise separable convolution decomposes a standard convolution into two parts: (a) a depthwise convolution, where each input channel is filtered independently, and (b) a pointwise convolution that combines the results from the depthwise step across all channels. In their work, Chen et al. investigate separable convolutions, where the depthwise convolution utilizes atrous convolution with a rate of 2, as shown in (c) [11].	30
4.6	Visual representation of sample type and patient type distribution throughout the used splits.	32
4.7	Example of the three training configurations being evaluated on the chromo test set.	32
4.8	Qualitative examples of the chromoendoscopy ID test setting with the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).	34
4.9	Qualitative examples from the chromo OOD test set using the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).	34
4.10	Qualitative examples of the colonoscopy ID test setting with the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).	35
4.11	Qualitative examples of the colonoscopy OOD test setting with the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).	35
4.12	Second train-val-test splits used for the color pre-processing experiments.	37
4.13	Considered color transformations.	38
4.14	Visual comparison of the convolution operation computed in the U-Net model with the dilated convolution present in DeepLabV3+.	40
5.1	CycleGAN architecture. G_{AB} translates images from domain A to domain B and G_{BA} performs the inverse mapping. D_A is the discriminator for domain A.	44
5.2	Examples of $\mathbf{A} \rightarrow \mathbf{B}$ image generation. The $\mathbf{A}$ domain corresponds to the normal colonoscopy data domain, and the $\mathbf{B}$ domain represents the chromoendoscopy domain.	46
5.3	Examples of BA image generation. A domain corresponds to the normal colonoscopy data domain, and B domain represents the chromoendoscopy domain.	46
5.4	Examples of the color transfer operation between same-technique samples.	47
5.5	Examples of the color transfer operation between different-technique samples.	48
5.6	Example of Random Color Erasing (RCE) in a polypgen sample.	48
6.1	Visual examples of semantic correspondences between two PolyGen instances. For each pair, each colored point on the left image corresponds to semantic match to the same colored point on the right image.	54
6.2	Visual analysis of the change in descriptor PCA representation through progressively deeper layers of the ViTs8 and ViTs16 pretrained backbones. Here we represent these extracted features from two example images.	56
6.3	Examples of nearest neighbor search results in the reference pool. (a), (b) and (c) mark each target/reference pair found.	57
6.4	Three examples of polyp descriptors-based similarity search. (a), (b) and (c) denote different target/reference pairs found.	57

6.5	Part co-segmentation pipeline. The group of input images (a) goes through the feature extractor, retrieving the key descriptors (b) and saliency maps (c) from a specified layer of the ViT architecture. All descriptors are then clustered using K-Means (d). Each cluster is then assigned the class "background" or "foreground" based on saliency, forming co-segments (e). A new clustering is applied exclusively to the foreground descriptors (f), resulting in foreground part-co-segments [12].	59
6.6	Train set clusters t-SNE representation (left) and origin center of each sample (right).	60
6.7	t-SNE representation of the inference clusters for the ablation study.	61
6.8	Examples of samples assigned to each considered class.	61
6.9	Multiple Sisters Pipeline. The target image (a) is assigned k nearest neighbors via NN search (b). k+1 paired combinations are then formed (c), and part co-segmentation is performed on each pair (d). The resulting parts are separated and aligned (e) and a weighted binary mask along with an uncertainty map are generated (f).	62
6.10	Quantitative In-Distribution (ID) results using 2 stages of clustering. The Y axis represents the dice score (Equation 4.3), and the X axis represents the group divisions, ranging from less specific (global) to most specific (sister variants). Each colored line represents an extractor configuration. Extractors with self-supervised pretraining are colored in pink dotted lines, whereas those with supervised training are depicted in blue dash-dot lines for better interpretation.	65
6.11	Quantitative In-Distribution (ID) results using 3 stages of clustering.	66
6.12	Quantitative Out-Of-Distribution (OOD) results using 2 stages of clustering. . . .	66
6.13	Quantitative Out-Of-Distribution (OOD) results using 3 stages of clustering. . . .	67
6.14	Qualitative results across extractors and data divisions. Columns represent extractors and rows division types. A = ImageNet pretrained Vision Transformer (ViT)	68
6.15	Qualitative failure/underperformance examples. Columns represent extractors and rows division types. A = ImageNet-pretrained ViT , B = GastroNet-pretrained ViT .	68
6.16	Example for a single pair. The second column shows the first clustering performed on the dense descriptors of both images, where correspondent colors mark descriptors of the same cluster. The third column depicts the final clustered parts, where, similarly, same color indicates same semantic part. Finally, the last columns show the segmented polyp mask and ground truth for the target image.	69
6.17	Examples of predictions and uncertainty maps for the 'multiple sister pairs' approach.	69
6.18	Logistic-based Polyp Part Selection. For each semantic part, the descriptors of each patch are evaluated by a logistic regression classification model that outputs the probability of it being a polyp patch. In the end, the predictions for all-part patches are combined, forming a global probability per part.	70
6.19	Qualitative test set examples for the multiple sisters approach vs U-Net, including corresponding uncertainty maps. Cooler colors translate lower uncertainty, whereas hotter color indicate higher uncertainty.	72
6.20	Representation of two sister combinations for each example. First combination being the target and itself, and second with one of the retrieved sisters.	72

6.21	KeyViTSeg pipeline. The input image (a) is passed through a ViT encoder (b). Key features are extracted from the multi-head attention block of the last layer (11) of the ViT extractor (c). A projection layer reshapes the ViT features to match the spatial dimensions expected by the CNN decoder (d). The reshaped features are input to the decoder (e), generating a segmentation mask (f).	74
6.22	Boxplot showing Dice Score Coefficient (DSC) distribution across models. . . .	79
6.23	Qualitative results for the Extreme Single Domain Generalization (ESDG) tests. .	82
6.24	Examples of failed segmentations.	83
A.1	Qualitative results for each cluster (rows) across different feature extractors (columns). .	98
A.2	Qualitative failure results for each cluster (rows) across different feature extractors (columns).	99
A.3	Qualitative results for each class (rows) across different feature extractors (columns). .	101
A.4	Qualitative failure results for each class (rows) across different feature extractors (columns).	102

List of Tables

1	Sustainable Development Goals addressed by the work developed in this dissertation, specifying the contributions and targets it addresses.	i
4.1	U-Net performance on chromoendoscopy samples (in-distribution and out-of-distribution). The best performance is highlighted in bold	33
4.2	U-Net performance on regular colonoscopy samples. Best performance is depicted in bold	34
4.3	U-Net performance on both types of samples. The best performance is depicted in bold	35
4.4	DeepLabV3+ performance on chromoendoscopy samples. Best performance is depicted in bold	36
4.5	DeepLabV3+ performance on regular colonoscopy samples. Best performance is depicted in bold	36
4.6	DeepLabV3+ performance on both types of samples. Best performance is depicted in bold	36
4.7	U-Net results for the color pre-processing experiments using the first splits. The best performance is highlighted in bold , while results surpassing the baseline (no transformation) are <u>underlined</u>	39
4.8	DeepLabV3+ results for the color pre-processing experiments using the first splits. The best performance is shown in bold , and performances surpassing the baseline (no transformation) are <u>underlined</u>	40
4.9	U-Net results for the color pre-processing experiment using the second splits, built with the entire PolypGen dataset. The best performance is shown in bold , and performances surpassing the baseline (no transformation) are <u>underlined</u>	41
4.10	DeepLabV3+ results for the color pre-processing experiment using the second splits. The best performance is shown in bold , and performances surpassing the baseline (no transformation) are <u>underlined</u>	41
5.1	Results of U-Net and UM-Net Performance with Color Transfer (CT) and Random Color Erasing (RCE). Best performance is depicted in bold , and the second-best performance is <u>underlined</u>	49
5.2	Results of U-Net and UM-Net Performance with Color Transfer (CT) and CycleGAN Generated Samples (GS). Best performance is depicted in bold , and second-best performance is <u>underlined</u>	50
6.1	Relative improvement (in %) added from the multiple sisters approach compared to the initial sisters grouping.	67

6.2	DSC results from the 5-fold cross-validation comparing the multiple sisters approach (K=6) with the fully supervised baseline. Best performance is presented in bold	71
6.3	Ablation results for varying extractor settings. The best performance is represented in bold , and the second-best is <u>underlined</u>	75
6.4	Comparison between KeyViTSeg and UM-Net Decoder Architectures	76
6.5	Results for the Domain Generalization Test (DG test) and the Extreme Single-Domain Generalization Test (ESDG test), comparing the proposed KeyViTSeg model and the baselines. The subscripts presented on the model names indicate the fold of the test. Evaluation of $Model_c$ under a DG test indicates that the model was trained on all centers excluding c for evaluation. Similarly, $Model_c$ for an ESDG test indicates that the data from c was used for model training, leaving all other domains for testing. Best performance is depicted in bold	78
6.6	Friedman test results	80
6.7	Wilcoxon signed-rank test results for different model pairs	80
6.8	Evaluation of model size and inference time. Best result is depicted in Bold	81
A.1	Part Co-segmentation quantitative results globally (in-distribution and out-of-distribution). In the "Finetuned on" column SSL stands for Self-Supervised Learning, more particularly DINO training and is followed by the number of training epochs; SL means Supervised-Learning and is equally followed by the number of epochs. The best performance is bold , and the second-best is <u>underlined</u>	95
A.2	Part Co-Segmentation global quantitative results for cluster division (in-distribution and out-of-distribution). The best performance is bold , and the second-best is <u>underlined</u>	96
A.3	Performance results per cluster in-distribution (ID) with two stages of clustering for the different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents the type of supervision training (Self-Supervised Learning) and 100 the number of epochs it was trained for. The best performance per cluster is bold , and the second-best is <u>underlined</u>	96
A.4	Performance results per cluster in-distribution (ID) with three stages of clustering for different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents self-supervised learning trained for 100 epochs. The best performance per cluster is bold , and the second-best is <u>underlined</u>	96
A.5	Performance results per cluster out-of-distribution (OOD) with two stages of clustering for different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents self-supervised learning trained for 100 epochs. The best performance per column is bold , and the second-best is <u>underlined</u> . In the out-of-distribution cluster assignment, some clusters do not have assigned samples. For these, the missing values are indicated by "-".	97
A.6	Performance results per cluster out-of-distribution (OOD) with three stages of clustering for different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents self-supervised learning trained for 100 epochs. The best performance per column is bold , and the second-best is <u>underlined</u> . Missing values are indicated by "-".	97

A.7	Part Co-Segmentation quantitative results for class division (in-distribution and out-of-distribution). The best performance is marked in bold , and the second-best in <u>underlined</u>	100
A.8	Performance results per class in-distribution (ID) with two stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is bold , and the second-best is <u>underlined</u>	100
A.9	Performance results per class in-distribution (ID) with three stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is bold , and the second-best is <u>underlined</u>	100
A.10	Performance results per class out-of-distribution (OOD) with two stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is bold , and the second-best is <u>underlined</u>	101
A.11	Performance results per class out-of-distribution (OOD) with three stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is bold , and the second-best is <u>underlined</u>	101
A.12	Performance results for the paired sister images approach with $N = 1$. The best performance is bold , and the second-best is <u>underlined</u>	102
A.13	Performance results for the multiple sisters approach considering 6 sisters. The best performance is bold , and the second-best is <u>underlined</u>	103

Abbreviations and Symbols

- CL** Contrastive Learning. xvi, 8, 12, 19–23
- CLAHE** Contrast-Limited Histogram Equalization. xvi, 38–41
- CNN** Convolutional Neural Network. xvi, 5, 6, 9, 10, 12–14, 20, 64, 74
- CRC** Colorectal Cancer. xvi, 1, 2
- CT** Color Transfer. xvi, 47, 49, 50
- DG** Domain Generalization. xvi, 76, 77, 79
- DINO** Distillation with No Labels. xvi, 4, 23–25, 52, 53, 55, 59, 64, 70, 73, 74, 76, 77, 83–85
- DL** Deep Learning. xvi
- DSC** Dice Score Coefficient. xvi, 30, 33, 49, 77, 79
- ESDG** Extreme Single Domain Generalization. xvi, 76, 77, 79, 81
- FFNN** Feed-Forward Neural Network. xvi, 10, 11
- FN** False Negative. xvi, 30
- FOV** Field Of View. xvi, 9
- FP** False Positive. xvi, 30
- GAN** Generative Adversarial Network. xvi, 17, 45
- ID** In-Distribution. xvi, 32–34, 36, 37, 39, 65, 66
- k-NN** k-Nearest Neighbor. xvi, 56–58, 61
- MHSA** Multi-Head Self-Attention. xvi, 11, 74
- MLP** Multilayer Perceptron. xvi, 11, 22
- NN** Neural Networks. xvi, 6
- OOD** Out-Of-Distribution. xvi, 12, 32–36, 39, 43, 49, 66, 84

RCE Random Color Erasing. xvi, 48–50

SL Supervised Learning. xvi, 17, 21

SOTA State-Of-The-Art. xvi, 7–9, 12, 15, 16, 22, 24, 43, 53, 75, 77, 84, 86

SSL Self-Supervised Learning. xvi, 3, 4, 16, 17, 20, 23–25

TN True Negative. xvi, 30

TP True Positive. xvi, 30, 31

UDA Unsupervised Domain Adaptation. xvi, 8

ViT Vision Transformer. xi, xvi, 4, 11–13, 15, 24, 25, 51–56, 59, 64, 67–69, 73–76, 83–85

Chapter 1

Introduction

1.1 Background

Colorectal Cancer (CRC) is the third most prevalent cancer type globally, accounting for 10% of all new cancer cases. Unfortunately, it is also the second biggest contributor to cancer-related deaths. Polyps are the source of approximately 95% of colorectal cancers, originally non-cancerous lesions that develop in the surface of the colon or rectum. Although the survival rate exceeds 90% in the early stages of the disease, it drops significantly to 5% in the later stages. This highlights a strong need for effective screening and detection of these lesions at an early stage, thus preventing them from progressing to a malignant state [13, 14].

Polyps can be classified as sessile or pedunculated according to the nature of their connection to the colon or rectum wall. 85% of colorectal polyps fall into the category of sessile polyps, having a broad base and being directly attached to the colon or rectum. On the other hand, pedunculated polyps have a narrow stalk that connects them to the rectum wall (Figure 1.1) [15].

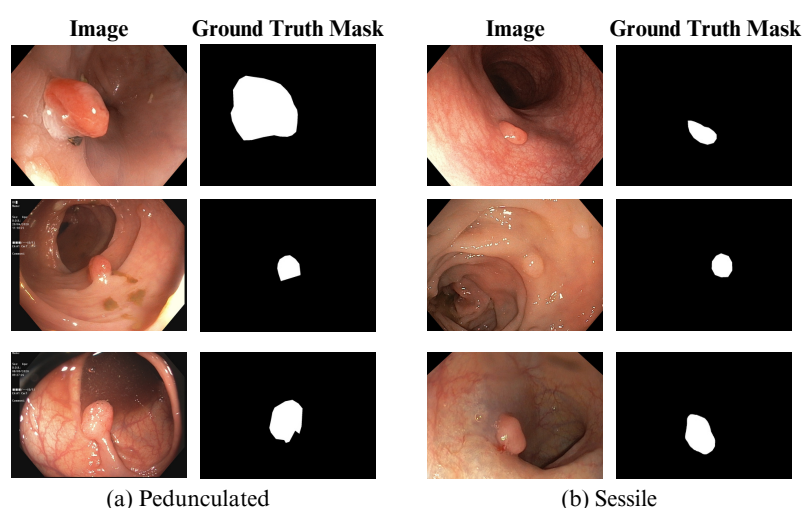


Figure 1.1: Representation of two of the most common types of polyps, followed by their corresponding segmentation masks [1]. In (a) we show examples of pedunculated polyps, while in (b) we show sessile polyps.

Polyps can also be classified according to their growth pattern into two main types: adenomatous (or adenoma), which are the most common precursors to cancer, and serrated, which feature a saw-tooth-like pattern when viewed under a microscope (Figure 1.2). If polyps become cancerous, cancerous cells invade blood and lymph vessels, which are responsible for removing fluids and byproducts from the colon and rectum. Early detection of these lesions through screening tests is, therefore, crucial for their identification and removal before they progress to cancer.

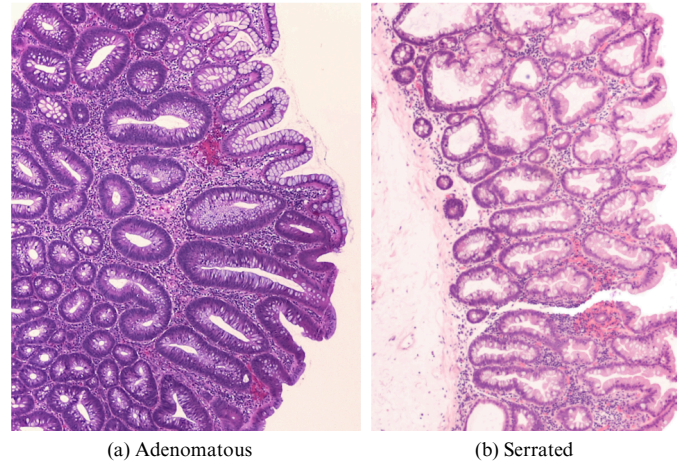


Figure 1.2: Representation of the two polyp types according to their growth pattern. In (a) we show adenomatous, while in (b) we show serrated [2].

The gold standard technique for polyp detection is colonoscopy, which has been effective in reducing cancer rates but, unfortunately, has a predictable miss rate for adenomas of about 6–27% [13]. Polyps can be easily overlooked if they are outside the visible area or due to the high speed of the colonoscope. The detection of polyps is also quite challenging due to their varying shapes, sizes, appearances, and colors. Due to this, the colonoscopy technique is highly dependent on the expertise of the doctor performing it, resulting in a significant rate of missed or inconsistent detections amongst professionals [14].

Recent advances in deep learning for computer vision applications have shown great potential in medical imaging analysis, achieving performance levels similar or superior to experts. Thus, these technologies offer promising opportunities to enhance the precision and reproducibility of CRC diagnoses by being integrated in the clinical workflow, addressing some of the limitations associated with human interpretation [16].

Deep learning techniques for polyp diagnosis can focus on simple lesion detection, which involves localizing the lesion within the image, or can extend to pixel-level classification, where the boundaries of the polyp are fully delineated. Although the latter is more complex than simple detection, it offers a much more precise identification of the lesion and can provide valuable guidance for biopsy removal. However, despite their potential, deep learning solutions face significant challenges when implemented in clinical settings, particularly in terms of medical bias and generalization.

1.2 Medical Bias and Generalization Issues

Deep learning models are highly data-hungry, significantly benefitting from exposure to large amounts of data. Obtaining such large, representative real-world datasets is challenging, especially in medicine, where sensitive data is protected by strict privacy regulations that limit its access [17]. Nevertheless, even when models are trained on multicenter datasets, they often struggle to generalize across different data distributions. This variability, known as *domain shift*, arises from factors such as differing scanning devices, acquisition settings, protocols, and patient anatomies [18]. These differences introduce patterns in the data that are not related to its clinically relevant features but are specific to data distribution. These patterns are then captured by the model during training, thus leading to an inherent *bias*. A model is considered biased when it makes predictions based on incorrect or incomplete assumptions. In this case, the model often relies on distribution-specific patterns instead of meaningful domain-transversal features of the lesion, which it fails to identify in other distributions, leading to incorrect predictions [19].

A specific example of such bias in medical imaging is the chromoendoscopy bias, which arises due to the visual variance introduced by a blue contrast agent during the colonoscopy exam. This contrast agent creates a visual contrast between the colon tissue and the lesion, useful for better clinical identification, especially when the polyp has a relatively small size [20]. However, in the deep learning field, this introduces color variability in the data, which can impact model performance and reduce its ability to generalize to different clinical domains.

Although transfer learning and domain adaptation methods have been suggested to mitigate these biases, they often rely on large annotated datasets, which are both time-consuming and expensive to acquire, highlighting the need for label-independent frameworks to improve the robustness of learned features [18].

1.3 Motivation

As discussed, early detection of colorectal lesions is essential for improving patient survival rates and overall care. However, current colonoscopy screenings are heavily dependent on the clinicians' expertise, leading to an elevated rate of missed detections. Deep learning models have shown potential to improve these detections and, therefore, patient survival [13, 14], but their clinical use is limited by challenges such as their lack of generalizability. This creates an increasing demand for the search and development of more reliable and clinically trustworthy AI solutions.

Recently, **Self-Supervised Learning (SSL)** techniques have been widely explored both for their label independence, thus allowing the inclusion of larger amounts of data in model training, as well as their ability to extract more generalizable and robust representations of the data [21, 22]. In **SSL**, the model is trained to solve a pretext task guided by signals inherent in the data, which can then be applied to downstream tasks like object detection or semantic segmentation. Because this task doesn't depend on manual labels, it avoids label bias and its limitations. As a result,

the learned features are more resistant to data variations, improving generalizability. Since these representations come directly from the data, they are often more interpretable as well [22, 23].

This dissertation aims to contribute to medical research by addressing the unexplored chromoendoscopy bias and developing a generalizable polyp segmentation solution by leveraging **SSL** features. The use of **SSL** allows for more efficient data utilization, bypassing the limitations of manual labeling and providing a more interpretable representation of the data, thus offering a promising approach to overcoming the generalization challenges faced by current models.

1.4 Structure of the Dissertation

Besides the introduction, this dissertation contains 6 more chapters. As we explore the enhancement of polyp segmentation through **SSL**, there are two chapters dedicated to the literature review. Chapter 2 provides an overview of key works in the polyp segmentation field, while Chapter 3 presents the current relevant concepts and methodologies in **SSL**.

Chapter 4 presents the dataset used, along with an introduction to two benchmarking architectures employed in the preliminary experiments. These experiments include a practical assessment of the chromoendoscopy bias, i.e., how the introduction of chromoendoscopy samples during training affects model performance and capacity to generalize. Within this chapter we also explore traditional color preprocessing techniques to address it, giving rise to their limitations and motivating Chapter 5. Here, we explore more advanced color invariance techniques, aiming to reduce the models' focus on color cues of the data during training for improved color bias mitigation. Chapter 6 shifts focus from chromoendoscopy to broader medical biases, exploring the semantic power and robustness of **ViT** features pretrained with the **Distillation with No Labels (DINO)** framework to enhance polyp segmentation methods. Finally, Chapter 7 summarizes the main findings of this dissertation, as well as future work directions. In Appendix A, we present quantitative and qualitative results that complement those presented in the main body of Chapter 6.

Chapter 2

Literature Review: Polyp Segmentation

Image segmentation plays a fundamental role in computer vision by partitioning an image into various regions based on its distinct characteristics. It can be divided into two subtasks: semantic segmentation and instance segmentation. In semantic segmentation, each pixel in an image is labeled, representing the object class to which it belongs. This is essentially a classification task executed at the pixel level. In contrast, instance segmentation differentiates between distinct instances of the same object class, giving each instance a unique label. For example, in an image with two cats, semantic segmentation labels all cat pixels as 1, while instance segmentation assigns cat 1 the label 1 and cat 2 the label 2. There is also the possibility of combining these two approaches, giving rise to panoptic segmentation. This third approach considers each component of the global scene by giving both a semantic class label and an instance label, which greatly improves overall scene perception. Revisiting the cat image example, pixels of cat 1 would be labeled as 1, while pixels of cat 2 would be labeled as 2 [24].

Polyp segmentation aims to detect and segment polyps automatically in colonoscopy frames as an auxiliary measure for clinicians to reduce the missed polyp rate in such exams. Conventional methods focus mainly on low-level and known features like texture, shape, color distribution, and edge characteristics. Due to their simplicity, they tend to fail when faced with more complex scenarios. As a result, and with the rise of deep learning, new approaches began to appear. For polyp segmentation, the most commonly used deep learning approaches include **Convolutional Neural Networks (CNNs)**, encoder-decoder architectures, Recurrent Neural Networks (RNNs), attention mechanisms, and generative models [25].

This chapter presents an overview of past works that have shaped the use of computer vision techniques for image segmentation, with a focus on polyp segmentation applications. We organize past works based on the type of architecture employed, dividing them into **CNN**-based methods, transformer-based methods, and hybrid approaches [14]. In each subsection, we further group works based on the main features or specific limitations addressed. In the end of the chapter, we also provide an overview of the main challenges still faced in the field and future research directions.

2.1 Convolutional Neural Network-Based Methods

CNNs introduce convolutional and pooling layers to traditional Neural Networks (NNs). Convolutional layers apply filters (also called kernels) capable of capturing local patterns or textures to the input images. This information is obtained due to the convolution operation between the filters and the image at various locations. The filters slide through the image, and, in each filter position, the dot product between the overlapped image area and filter is computed, producing feature maps. The displacement between successive filter positions on the image is called the stride. Therefore, a stride of 2 indicates that the kernel moves two pixels each time it slides over the image. The dimension of these feature maps is reduced thanks to the pooling layers (typically max pooling) that retain only the main information they hold. To reconstruct segmentation maps from this compressed representation, deconvolution or upsampling layers are often used [26].

The U-Net [3] architecture (Figure 2.1) introduced a strong foundation for biomedical image segmentation. This model first processes the input in a contracting path where it undergoes multiple convolutional layers that progressively downsample the images until they reach the maximum number of channels defined. These deep feature maps are then processed by the expanding path, which progressively upsamples the encoded information. The final convolution maps the upsampled channels to the desired number of classes (2, in the case of binary segmentation). Throughout the expanding path, skip connections allow the integration of low-level encoder features and high-level decoder features, recovering some of the lost spatial information lost during downsampling.

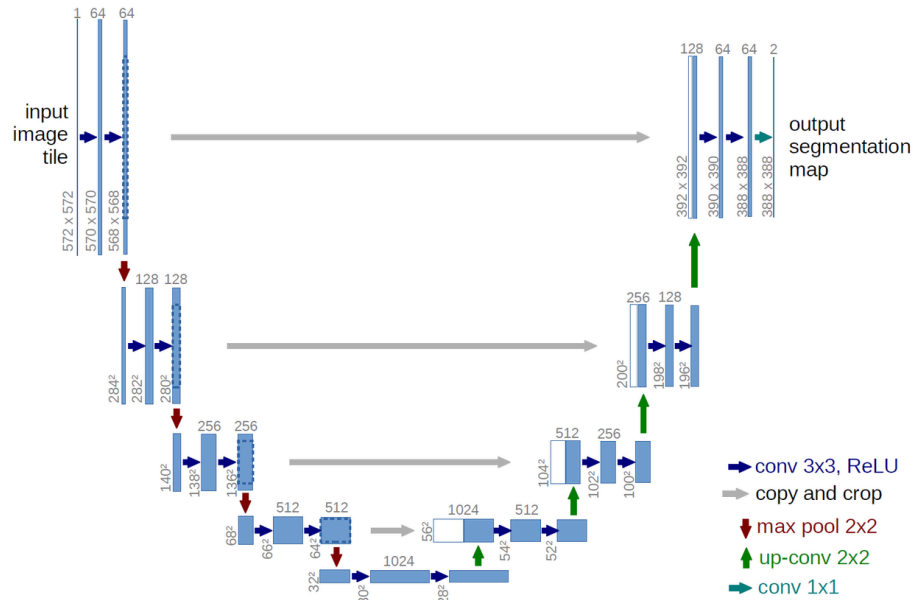


Figure 2.1: U-Net Architecture [3]

Skip connections improvements and Multi-scale feature fusion. Many models were inspired by U-Net and achieved notable performance improvements. UNet++ [27], for example, follows the original encoder-decoder architecture, introducing a redesign of the original skip connections

and the integration of deep supervision. It replaces the direct concatenation between same-level features of the antagonistic paths by dense skip connections, promoting a semantic refinement of encoder features by approximating them from what is expected by the decoder, thus improving baseline results. This architecture was not specifically designed for polyp segmentation; however, it was tested on a colonoscopy dataset in the original paper, and it is often used as a baseline comparison for new **State-Of-The-Art (SOTA)** polyp segmentation methods. A-DenseUNet [28] builds on U-Net++, using densely connected convolutional blocks within the encoder, allowing for a higher network depth while avoiding exploding gradients. Additionally, it uses attention blocks after each encoder layer to provide information on which sections of the network to keep more focus on. MSNet [29] further incorporates multi-scale feature fusion by computing differential features between adjacent layers and combining them in a pyramidal manner. Furthermore, it introduces LossNet, a training-free perceptual supervision network built on pretrained classification models. LossNet supervises the segmentation predictions by comparing the features of the prediction and true label at different scales. PEFNet [30] modifies the original U-Net architecture by using an EfficientNetV2 module with positional embedding. It promotes the output of each upsampling step to match the shape of the corresponding downsampling step, thus improving the quality of the merged information. The positional embedding merge block is an improved way of merging the encoded and decoded information at each step [31]. These contributions tackle U-Nets limitations on spatial information loss, leading to a better segmentation performance.

Multi-Task-Learning. Psi-Net [32] employs ResNet50 [33] as a shared encoder with three parallel decoders trained simultaneously for semantic segmentation, contour prediction, and distance map estimation. The distance map task predicts, for each pixel, its distance to the nearest object boundary. Combining these three related objectives in a joint loss enables effective multi-task learning, enriching the learned features.

Residual-based architectures and attention mechanisms. One of the issues regarding U-Net based models is the loss of spatial information that comes as a consequence to the progressive downsampling in the contracting path. In order to reduce this issue, several models tried to introduce residual connections that retain spatial information along layers. ResUNet++ [34], for example, is built on top of the Deep Residual U-Net (ResUNet) [35], combining residual and squeeze-and-excitation blocks. The latter models relationships between channels, adaptively weighting their importance. Additionally, ResUNet++ uses atrous spatial pyramid pooling to obtain input information at multiple scales. ACSNet [36] replaces traditional skip connections with channel attention that adaptively selects feature information from context extraction and global extraction modules [36]. PraNet [37] mimics clinicians' approach to annotating polyps by first making a rough estimation of the polyp area and then refining it using a reverse attention mechanism. This mechanism progressively focuses on ambiguous regions by erasing areas already detected, improving segmentation accuracy. CoInNet [38] uses two feature-refinement modules: CoIn attention and Anomaly Boundary Approximation (ABA). CoIn attention combines convolution and

involution operations to capture features more flexibly and efficiently, with involution offering lower computational cost. The statistical attention unit extracts pixel-wise statistics to generate an attention map. ABA focuses on refining boundary predictions through recursive feature fusion, crucial for accurate segmentation. UM-Net [4], the current **SOTA** model for polyp segmentation, is an enhanced version of ICGNet by the same authors and utilizes ResNet34 as its encoder to extract hierarchical features across five levels, with channel sizes ranging from 64 to 512. To optimize computational efficiency, the three highest-level features are compressed to 64 channels using 1×1 convolutions. The architecture incorporates two key modules: the Reverse-Contour Guidance module (RCG), which uses a Convolutional Block Attention Module (CBAM) on the lowest-level features to emphasize critical details and improve boundary detection, and the Adaptive Local-Global Context Module (ALGM), which processes features from levels two to five to capture both local and global multi-scale context, enhancing the detection of polyps of varying sizes. These outputs are then combined with decoder features to recover spatial details, and the Hybrid Pyramid Pooling Fusion Module (HPPF) is applied to fuse multi-level decoder features, effectively aggregating multi-scale information (Figure 2.2).

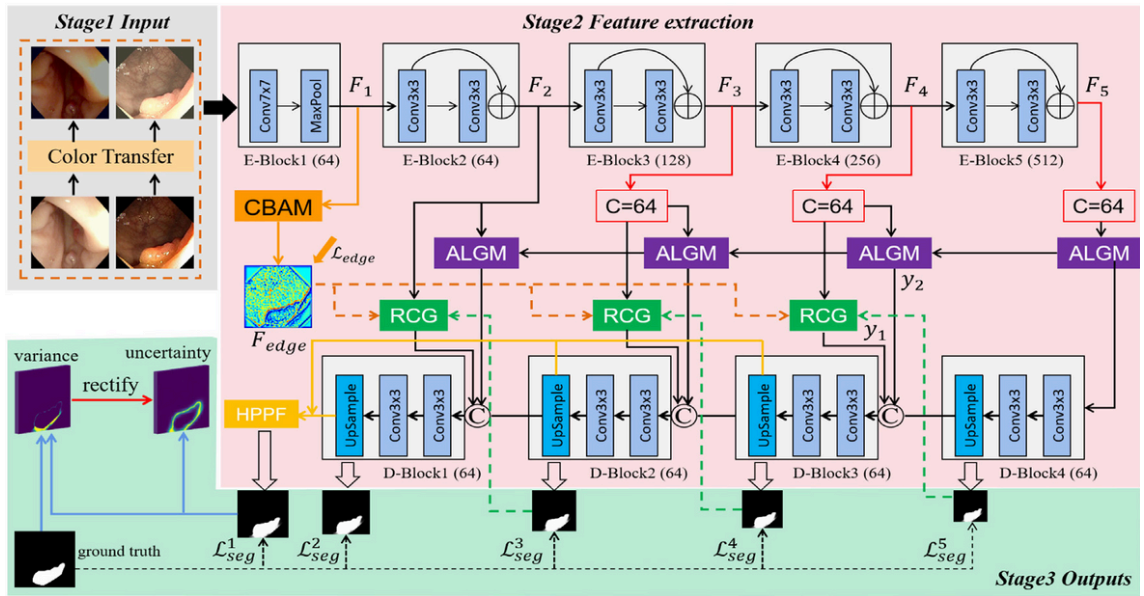


Figure 2.2: UM-Net architecture [4].

Addressing domain shift and generalization. Recognizing the challenge of domain variability in colonoscopy images, domain adaptation methods such as CoFo [39] apply **Unsupervised Domain Adaptation (UDA)** with Fourier transforms and adversarial contrastive learning to learn a content-style disentangled embedding space, enhancing feature robustness against domain shifts. FSM [40] addresses domain adaptation in a source-free scenario, where source images are inaccessible due to privacy regulations. Using a Fourier-style mining generator, it generates images from the source domain based on statistical data from a pretrained model. A **Contrastive Learning (CL)** framework is then used to reduce the domain gap. TRFR-Net [41] employs domain-invariant

feature decomposition with adversarial learning to extract domain-agnostic features. S2ME [42] combines ensemble learning and mutual teaching with a ResNet50 backbone, using spatial and spectral branches to capture domain-specific patterns and supervise each other in a weak supervision setup.

Capturing context and spatial dependencies. Due to limited receptive fields, traditional CNN struggle with capturing long-range dependencies. DeepLabV3 [43] tries to mitigate this issue by using atrous convolution. This convolution variation allows for a flexible **Field Of View (FOV)** of filters, increasing their receptive field without affecting the spatial resolution of the feature maps. This allows for their use at different rates and posterior fusion of this multi-scale information. PolypSeg [44] is a context-aware network with a U-Net backbone applied in colonoscopy video segmentation. This network learns context-aware information by taking advantage of multi-scale context extraction and aggregation. It further includes channel attention, providing focus to polyp-relevant channels and suppressing the less important channels.

Adaptative U-Net. After the introduction of U-Net, many models tried to introduce new modules and architectural modifications of its base to better adapt to their specific task. This generated an uncountable amount of U-Net variants and papers that, while introducing complexities that enhance task-specific performance, are not generalizable to other medical image domains. No-new U-Net (nnUNet) [45] was introduced as an adaptive architecture, offering flexibility on dataset sources and types within the segmentation task. Its focus is on maintaining U-Net simplicity, adding only minor changes to baseline 2D and 3D U-Net. Their network is able to adjust to various input patch sizes, number of pooling layers, and batch size depending on image resolution, anisotropy, and GPU memory constraints. Furthermore, it includes preprocessing strategies tailored to the input dataset and postprocessing techniques based on connected component analysis that remove unlikely structures based on the properties found in the training data. In their paper, they demonstrated competitive and **SOTA** results on the Medical Segmentation Decathlon challenge across 10 diverse datasets without relying on manual tuning, proving their method to be highly generalizable.

2.2 Pure Transformer-Based Methods

The main limitation regarding CNN-based models is their lack of ability to efficiently capture long-range dependencies in the images. This is primarily a consequence of the local nature of the convolution operation. The receptive field in CNNs, i.e., the image region that the network is processing at each layer, corresponds to the size of the used kernels, which is typically 3x3 or 5x5 and, therefore, quite limited. Due to this, it fails at capturing relationships between objects that are far apart in the images, which poses a strong contextual limitation to these models. Transformers, on the other hand, have a self-attention mechanism where the relative relevance of one image patch to all others is assessed and quantified, translating spatial relationships between objects or object

parts in different regions of the image (Figure 2.3). So, contrarily to what CNNs do, transformers explicitly model global and relational context. This advantage has made this category of models quite popular in current computer vision applications.

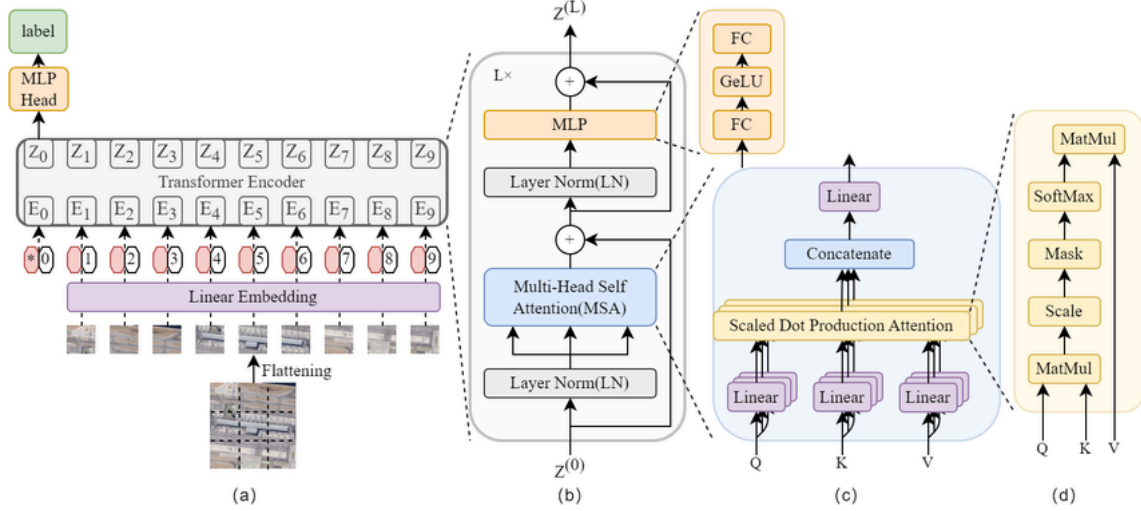


Figure 2.3: Vision Transformer (ViT) architecture. (a) Linear embedding and transformer encoder; (b) Multi-head self-attention (MSA); (c) Scaled dot-product attention, (d) Final linear transformations and output [5].

In transformer-based models, the image is first partitioned into smaller, fixed-size patches, which are then flattened into one-dimensional vectors and converted into embeddings. These embeddings are treated as tokens, viewed similarly by the transformer as words in the Natural Language Processing (NLP) models. Since the transformer does not inherently have a mechanism to understand how these patches are spatially arranged in the original image, it is necessary to add position embeddings to each patch embedding, encoding their spatial location in the original image. After turning the patches into embeddings and associating their spatial information, they are passed into the transformer encoder with multiple layers, each having two main components: a self-attention mechanism and a **Feed-Forward Neural Network (FFNN)**. In the self-attention module (Figure 2.4), each token (patch embedding) is represented in three ways: a query (Q), a key (K) and a value (V). These representations correspond to mapped transformations of the token. The query vector of a token i represents its question for all other tokens in the image, whereas the key embedding represents an answer for other tokens. On the other hand, the value indicates the real content to be combined according to the computed attention score. So, in order to determine how relevant other tokens are to token i , we compute the similarity between Q_i (the question token i is asking) and the keys of all tokens (the answers they provide). The softmax function (Equation 2.2) then normalizes these scores into a probability distribution, translating the relative importance of each patch to i . Ultimately, the attention scores are applied to compute a weighted sum of the values for updating patch i , emphasizing patches with higher relevance to i .

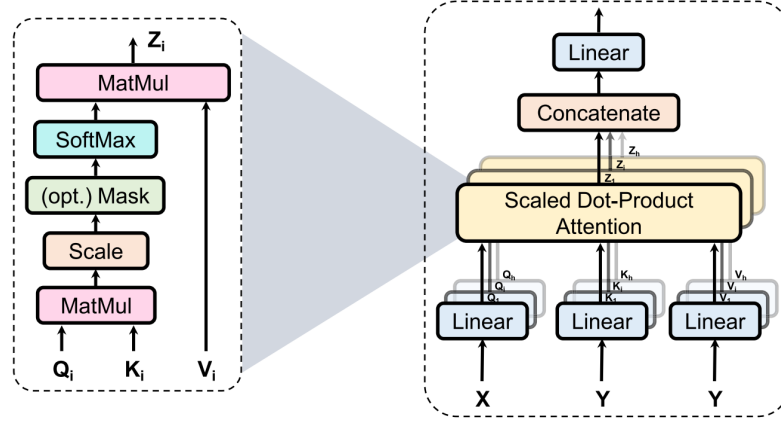


Figure 2.4: Vision Transformer (ViT) attention layer representation. Self-attention is computed when $Y=X$ [6].

The softmax function is defined as:

$$\text{Softmax}(\mathbf{z}_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (2.1)$$

where $\mathbf{z}$ is a vector, and z_i represents the i -th element of the vector.

The attention score is thus calculated as follows (equation 2.2):

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (2.2)$$

In **Multi-Head Self-Attention (MHSA)**, the process is repeated across multiple heads, each capturing different relationships between the patches by using different mapping functions to calculate Q , K , and V . These results are then passed through the **FFNN**, introducing non-linearity and enabling the model to learn more complex transformations.

In the context of polyp segmentation, transformer-based models, such as SegFormer [46], PTMF-Net [47], and PETNet [48], have been developed to harness these benefits. By using the self-attention mechanism of transformers, these models can better capture the contextual relationships between distant objects in the image, which is particularly important in colonoscopy images where polyps may be located far apart or surrounded by complex anatomical structures. For instance, SegFormer [46], introduced by Xie et al., combines a hierarchical Transformer encoder with a lightweight **Multilayer Perceptron (MLP)** decoder. The input image is first divided into 4×4 patches, which are fed to the Transformer encoder. The usage of small patches is beneficial for finer spatial resolution and, therefore, for the dense prediction task. The multilevel features obtained with the encoder are then fed to the **MLP** decoder to predict the segmentation mask. In contrast to the original **ViT**, which outputs a single feature map at a fixed resolution, the hierarchical encoder generates both high-resolution fine features and low-resolution coarse features.

This allows the model to combine detailed information with broad context, improving flexibility in segmentation tasks.

PTMF-Net [47] integrates a pyramid transformer encoder with a multibranch decoder. The encoder captures multi-scale features, which are subsequently refined and combined. To enhance texture representation, it applies spatial and channel attention mechanisms. Additionally, a boundary region extraction module targets uncertain edge regions of polyps. The decoder fuses the extracted multi-level features using the computed attention, preserving detail and context.

PETNet [48] includes a ViT encoder cascaded with a transformer attention layer in four scales. One of the decoders used is an auxiliary Gaussian decoder that represents polyps as probability heatmaps, while the remaining are regular binary decoders. The introduction of Gaussian probability to the segmentation process aims to enhance context comprehension and reduce the number of false positives. The combination of the gaussian and binary decoder branches is combined via soft convolution. This binary decoders ensemble tries to mimic the observation of two physicians in clinics to reduce missed polyp detections.

CTNet [49] has a hierarchical Transformer encoder (MiT) as its main feature extractor. These features are processed by enhancement modules, aiming to improve multiscale and high-resolution feature representation. In order to build a more organized feature space, this network is trained under CL settings. This framework is extensively tested on colonoscopy datasets and found to have good generalization abilities on the polyp segmentation task.

2.3 Hybrid Methods

Some methods try to combine CNNs and transformers and take advantage of the strengths of both, namely the local detail captured by CNNs and the long-range dependencies acquired by the latter. For example, Fu-TransHNet [50] comprehends a CNN branch and a Transformer branch for feature extraction. The paralleled extracted features are fused under the same scale, taking advantage of the local and global information that the two branches extract [31].

Domain Adaptation. Colonoscopy data exhibits high variability, which makes it challenging for models to generalize well to unseen domains, as they tend to learn domain-specific patterns. Polyformer [51] tackles this problem by integrating a Polypformer layer into existing networks like U-Net, enabling effective adaptation to new domains with minimal fine-tuning while retaining source domain knowledge. It achieves SOTA results in few-shot settings, narrowing domain gaps in heterogeneous data. TransNetR [52] improves generalizability through an encoder-decoder design using a ResNet50 encoder combined with transformer residual blocks in the decoder. This approach achieves top performance on Out-Of-Distribution (OOD) datasets and maintains real-time processing speeds, essential for clinical use. SSFormer [53] addresses the generalization issue of deep learning applications to unseen data. The typical CNN encoder is replaced by a combination of PVTv2 and SegFormer. The decoder is built to give local emphasis to the features

(such as textures and edges) and a refocus mechanism that prevents the model from dispersing its attention.

Boundary Refinement. Several transformer-based polyp segmentation methods explicitly incorporate modules or loss functions aimed at improving the accuracy and clarity of polyp boundaries, critical in polyp segmentation. For example, ColonFormer [54] integrates CNNs and Transformers to learn a hierarchical feature representation with multi-scale information. The encoder processes the image at different resolutions, resulting in varying feature map shapes that are then aggregated by the decoder. The decoder is designed in a hierarchical pyramid structure, enabling a gradual refinement of the feature maps. To further refine the segmentation predictions, it includes a refinement module based on a reverse attention mechanism.

TransFuse [55] combines CNN and Transformer branches in parallel and fuses the information extracted by both with a BiFusion module. This module uses both self-attention mechanisms and multi-modal fusion methods.

Multi-scale Feature Fusion. The extraction and fusion of multi-scale features for more accurate detection of the different shapes and sizes of polyps has been extensively explored by several approaches. Polyp-PVT [56] uses a Pyramid ViT encoder based on the PVTv2 architecture that outputs multi-scale feature maps. A Cascaded Fusion Module (CFM) is used to find the high-level polyp information in the features and fuse them based on convolution operations. A Camouflaged Identification Module (CIM) follows to extract important polyp information from the low-level features, such as texture and edges. Finally, a Similar Aggregation Module (SAM) uses attention to merge the pixel-wise information effectively [31]. LAPFormer [57] integrates a hierarchical transformer backbone. Instead of adding the features from the different stages directly, the decoder concatenates and linearly transforms them. After feature fusion, the most important features are selected based on attention and refined. It also uses skip connections to further integrate previous encoder information during decoder processing. In PPFormer [58], both a shallow CNN encoder and a deep Transformer encoder are used for feature extraction. It generates predictions at different levels, using a CNN encoder for lower-level features such as boundaries, introducing them to the transformer features for segmentation refinement.

2.4 Current Challenges

Even with notable progress in automatic polyp segmentation, several common challenges are still unresolved.

Polyp shape and variability. Polyps have irregular shapes and sizes, with significant 3D variation even within the same type. Additionally, the surrounding regions can be complex due to factors such as underdistension of the colon, residual stool, fluid, and anatomical structures such

as haustral folds, making segmentation more difficult [59]. In order to improve robustness and sensitivity to the complex boundaries between polyps and surrounding tissues, that sometimes lead to missed detections, several works have introduced edge feature extraction modules. Usually, since CNN low level features are highly focused on boundaries and texture-like patterns, these works use the features from these early layers for boundary refinement. Even though these are superficial features, they are still the result of convolution operations that intertwine edges with other features, making them hard to decouple. Other works explore frequency information [60, 61, 31] to distinguish object edges from its body. .

Lack of explainability. Another big issue comes from the fact that the majority of designed deep learning models don't prioritize interpretability. The lack of comprehension of the underlying considerations of these "black-box models" raises concern regarding their trustworthiness, which poses a severe setback in their integration in real-life clinical scenarios. It is important for the model to be able to provide clear and transparent explanations on the factors that lead to its decision in human diagnosis [14].

Privacy violation. There are many privacy restrictions regarding sensitive medical data. This makes it hard to assemble a significant amount of data for current data-hungry deep learning models. Several alternatives have been proposed to address this issue, such as using data augmentation techniques for synthetic data generation and employing federated learning. Federated Learning (FL), also known as collaborative learning, is a decentralized strategy for training machine learning models. In FL, there is no requirement to exchange raw data from the client devices to the central server. Instead, the model updates (typically parameters or gradients) are aggregated by the server and then shared with the clients, allowing them to update their local models without sharing raw data [14, 62].

Domain shift and Generalization. Existing polyp segmentation models are often built under the premise that the training and test data have the same data distribution. Frequently, colonoscopy image datasets have a high heterogeneity with varying factors such as illumination and image quality. By dividing the dataset in train and test sets, there is no guarantee that the different light and quality conditions are equally represented, which gives rise to domain adaptation techniques. Therefore, it is crucial that these models demonstrate the capacity to deal with variability, such as screening of different patients or using different devices [14].

Adversarial attacks. It has also been observed that deep models are susceptible to adversarial attacks, which, while imperceptible to the human eye, can significantly impact model performance. A change as little as altering a pixel of the image may induce a previously effective model to make a wrong prediction with a high level of confidence [63]. This confidence can be problematic in clinical settings as the doctor might rely and trust on the AI's output [14].

Computational cost and Real-Time Constraints. Another concern is the complexity of new models. Contemporary and advanced models are typically large and complex, which often requires a large amount of computational power and memory. For this model to be applied in clinical settings or on mobile devices, which don't usually have the required resources or cloud systems. To tackle this, researches have been looking for ways to design lightweight models that are efficient yet smaller and therefore suitable for less powerful hardware. Furthermore, the introduction of these models in real clinical settings would require them to be able to process data in real-time [14].

Dataset Limitations and Bias. Medical datasets are often limited and biased due to strict regulations and high collection costs, leading to segmentation methods that struggle to generalize. Multi-center data still inherits biases like lighting issues, varying polyp sizes, and annotation differences. Therefore, there is a need for robust models that focus on intrinsic lesion characteristics, ensuring they generalize effectively across diverse data sources.

2.5 Relevance to this Dissertation

In this chapter, we provided a comprehensive overview of key works in polyp segmentation. Since not all methods discussed are central to this dissertation, this section highlights the limitations addressed in this work and outlines the **SOTA** methods used in the experimental sections.

The main limitation we address is the lack of generalizability in existing models, worsened by the data heterogeneity commonly found in medical datasets.

UM-Net [4], the current **SOTA** model in polyp segmentation, serves as the main benchmark throughout this work. It is used in ablation studies in Chapter 5 and as a baseline for comparison in the proposed model in Chapter 6. Additionally, we compare the proposed approach with classic U-Net [3] and the competitive nnU-Net [45], both of which provide a solid performance foundation. DeepLabV3+ [11] serves as a baseline model for preliminary experiments in Chapter 4. Finally, the **ViT** self-attention mechanism, a key focus of this dissertation, plays a pivotal role in the experiments detailed in Chapter 6.

Chapter 3

Literature Review: Self-Supervised Learning

Supervised models rely on human expert intervention to generate "ground truths," or the correct labels for each data instance. In contrast, **SSL**, a subset of unsupervised learning, does not require labeled datasets. Instead, it extracts pseudo-labels from the inherent structure or patterns in the data, using these inferred signals for training. While **SSL** shares similarities with supervised methods, it relies on automatically generated labels that do not require human input, classifying it as an unsupervised approach. This is particularly useful when annotated data is scarce or costly to obtain, making **SSL** a time- and cost-effective alternative for training **SOTA** models. Furthermore, it has also been emerging as an effective mechanism to obtain meaningful feature representations of the data. By providing the model a pretext task such as input reconstruction, the model is led to compose compact representations of the images that retain the key information that allows successfully completing the task [22].

In this chapter, we present fundamental concepts of **SSL** and review influential works in the field. We categorize existing methods into image-level and dense approaches, based on the granularity of the learned representations. Image-level methods generate a single representation for an entire image, while dense methods produce distinct representations for each patch or pixel. Each category can be further divided into instance-level and group-level methods, depending on what the model discriminates. Instance-level methods focus on distinguishing individual samples—such as different images or patches—from one another. In contrast, group-level methods learn shared representations within groups, separating categories based on common features. For example, instance-level approaches differentiate between two cat images, whereas group-level approaches contrast images of cats against images of dogs, learning group-specific features.

3.1 Key Concepts

There are two important types of tasks in **SSL**: pretext and downstream tasks. Pretext tasks are designed to generate "pseudo-labels" from the unlabeled data. The term "pretext" indicates that

these tasks are not valuable on their own, but they help the model learn useful data representations for later use in downstream or target tasks. Because of this, pretext tasks are also referred to as representation learning. After models are pre-trained on large datasets using **SSL**, they are typically fine-tuned for a specific target task, which involves **Supervised Learning (SL)**, where the learned parameters are adjusted to fit the smaller, labeled dataset for the specific task. There is an alternative approach that applies the learned feature representations, or "applicable embeddings", directly on the downstream task, without the need for fine-tuning. A diagram representing this transfer learning operation can be seen in Figure 3.1. Some examples of pretext tasks can be seen in Figure 3.2 [64].

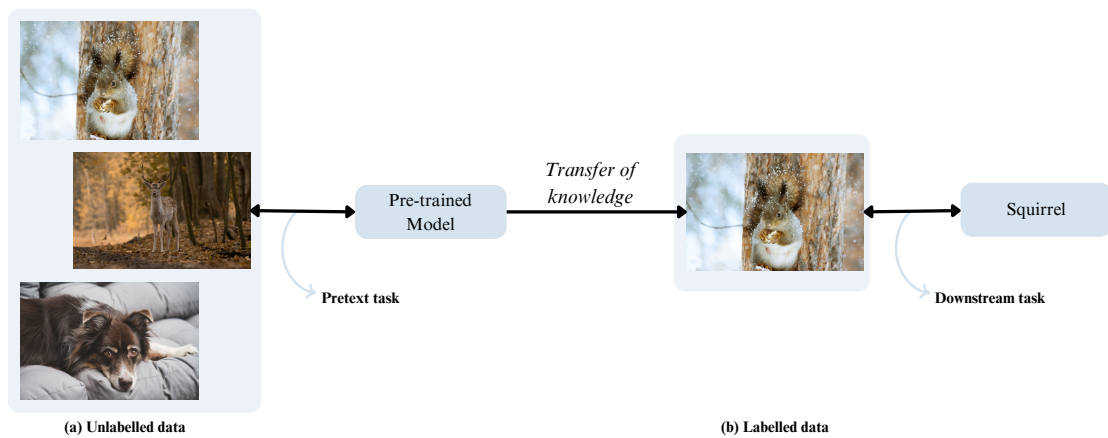


Figure 3.1: Pretext tasks and downstream tasks. The model is trained on a pretext task using data-extracted labels (in other words, in a self-supervised way). The knowledge acquired by the model after being trained on this pretext task is transferred to the downstream (or target) task. This figure was inspired in a similar representation [7].

We can divide **SSL** approaches into three categories: predictive, generative, and contrastive.

Predictive approaches. Predictive learning aims to understand semantic and contextual information about the data by predicting geometric transformations applied. For example, we can create different rotated views of the same image and define the learning task as prediction of the applied rotation angle [65].

Generative approaches. In generative learning approaches, the model learns to create new data that looks similar to the training data. By generating this data, the model also learns the important patterns and structure within the data. A good example is **Generative Adversarial Networks (GANs)**, where two models—the generator and the discriminator—work together. The generator starts with random noise and creates an image; the discriminator tries to distinguish whether the image is real, i.e., from the actual dataset, or fake, produced by the generator. As the generator improves in creating more realistic images, the discriminator gets better at telling the difference.

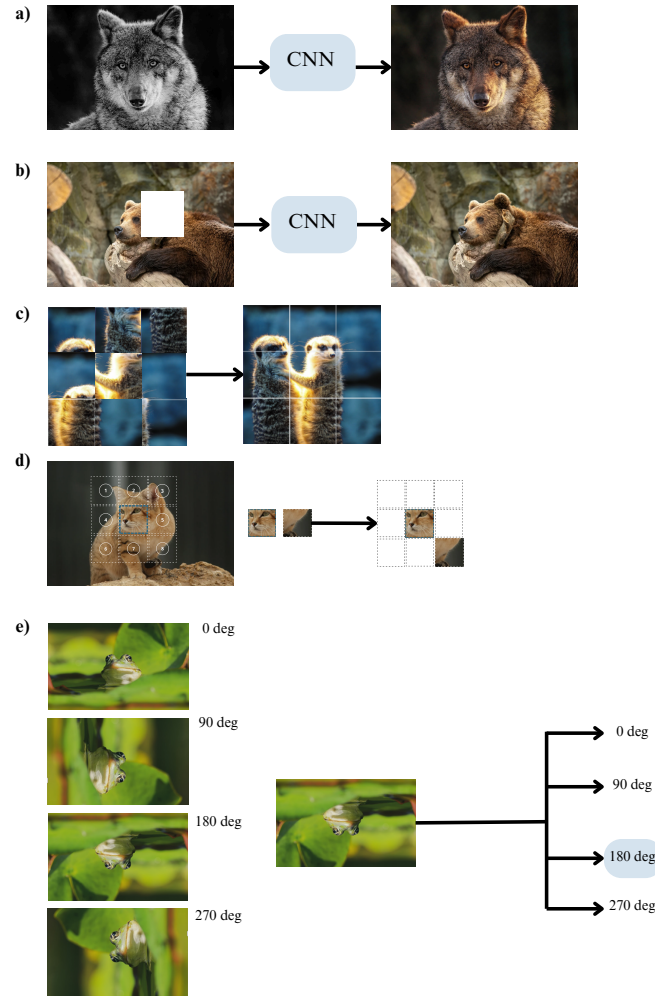


Figure 3.2: Pretext tasks used to enhance model training: a) Colorizing: Inferring colors for grayscale images; b) Inpainting: Filling in missing parts of an image; c) Jigsaw Puzzle: Rearranging shuffled image patches; d) Predicting Relative Position: Identifying the arrangement of image patches; e) Predicting Rotation Angle: Determining the angle of image rotation. Inspired by a similar representation [8].

This cycle continues until the generator produces highly realistic results. In this example, the internal layers of the generator and discriminator capture meaningful and descriptive features of the data through adversarial learning that can then be used in a downstream task.

Masked Language Modeling (MLM) approaches are prevalent in NLP, where the model is trained to predict hidden tokens (like words) in a sentence based on the visible ones. Masked Image Modeling (MIM) is a parallel concept and a variation of the Denoising AutoEncoder (DAE). In this generative approach, a portion of the input is hidden randomly during training (such as a selected patch), and the model learns to predict the missing part using the visible sections of the image. The SimMIM model is an example of the usage of MIM learning [66].

Contrastive learning. On the other hand, **CL** is based on the use of positive-negative pairs. Positive pairs are representations of images that belong to the same class, as indicated in Figure 3.3, whereas negative pairs are the contrary. It is a discriminative model that has the purpose of differentiating between augmented views of an image and learning from those variations. There are several methods for image augmentation, including cropping, resizing, distorting color, adding Gaussian noise, cutting a portion of the image, sobel filtering, and Gaussian blur, amongst others. The goal is to approximate similar samples in a latent space while pushing dissimilar samples apart by using a contrastive loss. The latent space is a transformed space where the model represents the data. This type of learning is heavily inspired by the ways humans learn. By effectively pulling together similar samples and pushing apart dissimilar ones, it captures the underlying structure of the data, making it useful in various applications [8, 67].

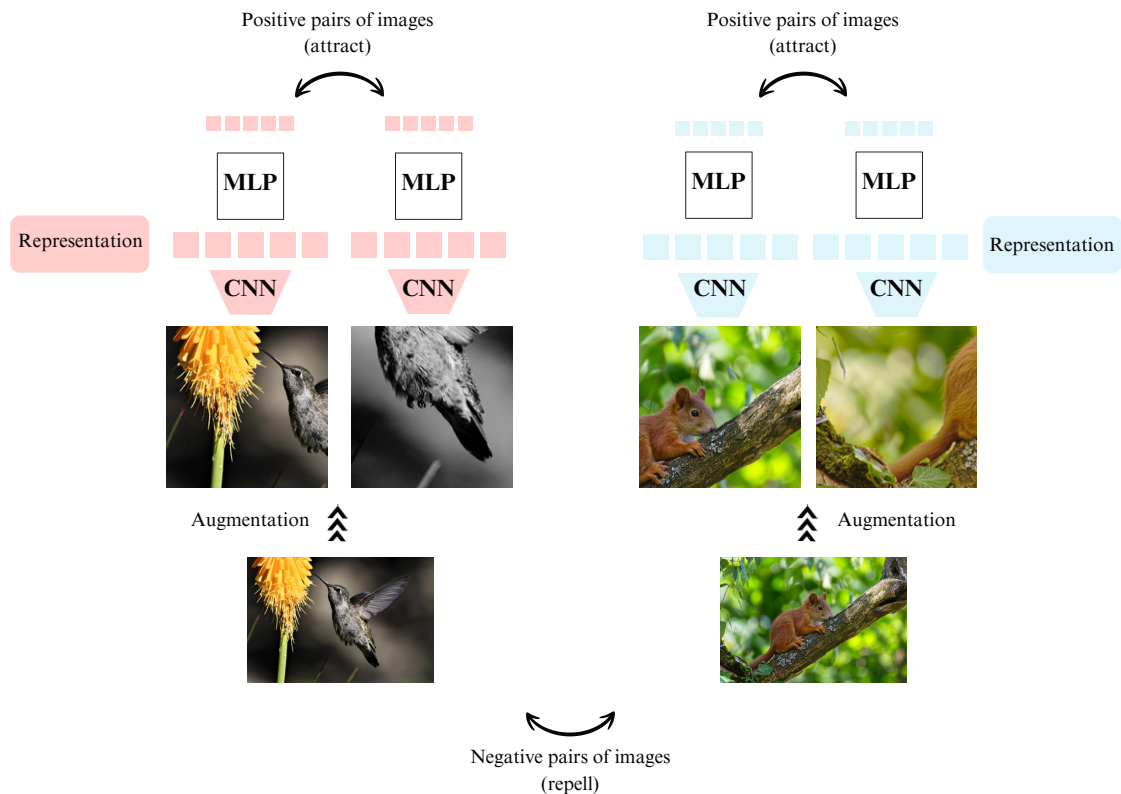


Figure 3.3: Contrastive learning simplified. Representations of positive pairs of images (augmentations belonging to the same class) are attracted whereas representations of negative pairs of images (belonging to different classes) are repelled. Inspired by a similar representation [9].

Feature evaluation. The quality of the features learned from the pretext task needs to be evaluated. This can be done by fine-tuning, where the pretext task's learned weights are used as a starting point and updated during training on a new model, or by applying a linear classifier to a small labeled dataset while keeping the learned features unchanged.

3.2 Image-Level vs Dense Self-Supervised Learning

We can divide current **SSL** methods into image-level or dense **SSL**.

3.2.1 Image-Level Self-Supervised Learning

Image-level **SSL** aims to learn meaningful feature representations for entire images without requiring human annotations. It can operate at the instance level or at the group level [68].

3.2.1.1 Instance-level Methods

Instance-level methods focus on distinguishing between individual instances of images rather than just their class. The goal is for the model to recognize different augmentations or transformations of the same image as belonging to the same instance, not just to a general category (like "cat"). For example, different views or augmentations of the same image are treated as different versions of that specific instance, and the model is trained to assign them the same representation. On the other hand, different images, even if they belong to the same class (e.g., different images of cats), are treated as separate instances with distinct representations. Common instance-level pretext tasks include predicting augmentations and **CL**. The core idea behind these tasks is that by learning to predict or reverse the transformations applied to images (e.g., rotations, crops, color changes), the model gains a deeper understanding of the underlying structure and features of the images.

An example of a self-supervised method for image rotation prediction is RotNet [69]. For each training image, a set of rotational transformations is applied, where the image is rotated by multiples of 90 degrees. The network is then trained to identify which specific rotation was applied to each image using a cross-entropy loss function. The core principle is that the neural network must detect and localize significant parts of the object within the image while understanding how their orientation changes with rotation. This is achieved by minimizing the distance between the feature maps of the original image and its rotated versions, allowing the network to learn rotation-invariant features. By doing so, the model can better understand the underlying structure of the objects, regardless of their orientation. The learned features can subsequently be applied to real-world tasks like object detection and image segmentation.

A model can also predict relative position based on intra-image context and learn useful feature representations. The image is divided into patches, and random pairs are formed. The convolutional neural network is then trained to identify the relative position of the second patch of the pair to the first. By doing so, the model learns to recognize objects and their constituent parts and, therefore, the semantic context of the image [70].

Another example is a **CNN** trained to solve jigsaw puzzles as a proxy task. In this method, the image is divided into patches that are randomly rearranged so that their spatial distribution no longer matches the correct, original geography of the image. By employing this pretext task, the

model is acquiring semantically relevant knowledge on the feature maps of object parts and their correct spatial arrangement [71].

On the other hand, in **CL**, the model learns to distinguish different images by pulling augmentations of the same class together in the feature space and of different classes apart. This is accomplished by employing a contrastive loss. This loss encourages the model to capture features that are invariant to the transformations applied to the images.

The SoCO [72] framework is designed specifically for the subsequent task of object detection. Previous works on image-level contrastive learning treated the entirety of the image as a single instance. This proposed idea looks at each object proposal in the image as an independent instance. SoCO creates several views of the object, changing its size and location in the image, thus helping the model learn several representations of the same object. Contrastive learning is applied here, by looking to approximate the feature maps of the representations of the same objects and separate views of different objects. The authors tested SoCO on a popular dataset (COCO) and found that it performed better than previous methods.

Obtaining effective image representations in SimCLR [73] requires using a variety of data augmentation techniques to strengthen the model’s ability to generalize. These augmentations, such as cropping, color distortions, and flipping, help create diverse views of each image, allowing the model to learn more robust representations. They also found that **CL** with SimCLR benefits significantly from larger batch sizes. Larger batches provide more negative examples (instances of different images) in each mini-batch, which helps the model differentiate between positive pairs (augmentations of the same image) and other images. This leads to more efficient learning compared to traditional supervised methods. Additionally, longer training durations were shown to improve the model’s performance, as more training steps allow for a better refinement of the learned representations. Like in **SL**, SimCLR also performs better with deeper and wider neural networks, as these architectures can capture more complex patterns in the data. Using deeper models helps contrastive learning methods extract richer, more meaningful representations from the images, leading to better performance [74].

Momentum Contrast (MoCo) [75] builds on **CL** by using a query and a key encoder. The first generates representations of the current mini-batch of images (called queries), while the key encoder generates representations of images that act as keys. MoCo’s goal is to match the query representations to the correct key representations (positive pairs), where both the query and key are different views of the same image (for example, different crops or augmentations). The key representations are stored in a dictionary (or queue) that is continuously updated as the model processes new images. When a new mini-batch of images is processed, their representations are added to the dictionary, and the oldest representations are removed, ensuring the model always has a large and diverse set of keys. By using this queue structure, the size of the dictionary (the number of negative samples it can hold) is independent of the mini-batch size. This way, the mini-batch can be small and the dictionary large. The key encoder is updated more slowly than the query encoder using a momentum update strategy. This helps ensure that the representations in the dictionary remain stable and consistent, improving high-quality representation learning over

time.

One key advantage of MoCo over SimCLR is that each query (positive pair) is compared against all negatives stored in the queue, not just the negatives in the current mini-batch. In contrast, SimCLR only uses the negatives from the mini-batch, which is why SimCLR requires larger batch sizes to provide enough negative examples. MoCo can achieve similar or better performance with smaller batches, thanks to its large and evolving queue of negatives.

A common issue in CL approaches is collapsing. Collapsing occurs when the model learns to produce similar representations for different inputs, therefore mapping them into the same point in the latent space. In this case, the model is not learning meaningful differentiations between images, thus leading to poor performance. SimSiam [73] is a Siamese Representation Learning model that applies stop gradients to tackle the collapsing issue. In this model, the two produces augmentations of the input image x are feed to an encoder f . One output of the encoder is sent to an MLP predictor and directly compared to the representation of the paired augmentation. The stop-gradient operation prevents gradients from affecting the model in the area where it's applied, ensuring that the two representations do not collapse in the feature space. Overall, this model operates similarly to SimCLR but does not use negative pairs [74].

BYOL [76] learns image representations without using negative pairs, unlike contrastive methods, showing SOTA results on ImageNet and other benchmarks. It uses two networks with the same architecture, an online network and a target network. It computes two augmented views for every input image. The online network then takes one augmented image and, from it, tries to predict the representation that the target network produced from the other view. A regression loss is used to minimize the prediction of the online network and the target network. The target network is updated by using a slow-moving average of the online network's weights. The latter is updated by gradient descent. This work proves that using negative pairs (i.e., contrasting against other images) is not necessary in order to learn meaningful feature representations.

3.2.1.2 Group-level Methods

In instance-level methods, the goal is to learn a representation for each individual image, whereas in group-level methods the focus is to learn representations by looking at groups of images together. Group-level methods learn shared concepts and patterns across images, often by using clustering techniques to group similar images and generate pseudo-labels for training.

CL, which typically compares pairs of image feature maps, can be computationally expensive. To address this, SwAV [23] was introduced. It is a more efficient approach that eliminates the need for explicit pairwise comparisons. Instead of comparing individual images, SwAV groups similar images into clusters, and the model learns by distinguishing between these clusters. During training, SwAV generates different views of the same image and ensures they are assigned to the same cluster. This mechanism operates by swapping prediction roles, with each view responsible for predicting the cluster assignment of the other. By doing this, SwAV achieves the benefits of CL in a more computationally efficient way.

3.2.2 Dense Self-supervised Learning

Recent studies have found that image-level representations are less effective for dense prediction tasks like object detection and semantic segmentation. However, up until recently, most **SSL** approaches were mostly focused on the first. Object detection refers to the identification and localization of a certain object in an image, whereas semantic segmentation involves pixel-level or patch-level classification. The reason for this may be that these representations do not capture detailed spatial information or specific features in order to allow a deep understanding of the structure and different object relationships in an image. Thus, image-level pre-training may lead to the comprehension of holistic features—that is, those that describe the overall content of the image—while neglecting finer details. An example would be a successful cat identification but no further understanding of the different parts that make it a cat [72].

In dense **SSL**, much like in image-level **SSL**, pretext tasks can be considered as instance-level or group-level tasks. Instance-level tasks, such as image reconstruction and **CL**, focus on learning from individual pixels or patches within an image. Group-level tasks, like dense clustering, work by identifying patterns across groups of images. Depending on how these tasks are implemented, they can be used to extract either pixel-level or patch-level representations, allowing the model to learn finer details or broader patterns within the data.

3.2.2.1 Instance-level Methods

By extending and generalizing existing **SSL** methods, DenseCL adapts them for dense-level tasks. In DenseCL, the feature maps are extracted from the input views through some backbone network (e.g. ResNet). After being extracted, the feature maps are passed on to two parallel working projection heads. The global projection head summarizes the image in a global feature vector, whereas the dense projection head preserves local details by creating dense feature vectors for different regions of the image. DenseCL trains both the global and dense projection heads together using a contrastive loss. This loss promotes the model to learn similar representations between different views of the same image and dissimilar for different images at both the global (image-wide) and local (region-specific) levels [77].

If the K-Means algorithm assigns all solutions to a single cluster, the optimization process can result in degenerate solutions. To tackle this issue, SeLa was proposed, addressing the problem by incorporating constraints that prevent representation collapse. This is achieved through the use of equipartitions and a Sinkhorn-Knopp algorithm simplified, ensuring a more balanced distribution of features during training [78].

3.2.2.2 Group-level Methods

The instance level methods can be expanded to group-level in dense prediction tasks.

Caron et al. [10] applied a **DINO** approach to train a transformer backbone in a self-supervised way. **DINO** is a self-supervised method that trains two neural networks simultaneously: a student and a teacher (Figure 3.4). Both networks receive different altered versions (views) of the same

image. The student learns by trying to imitate the teacher’s output, while the teacher’s parameters are updated gradually based on the student’s. This process does not require any labeled data. Special techniques, like output normalization and carefully designed updates, prevent the model from collapsing into trivial solutions. **DINO** is especially effective with **ViTs** and helps the model learn detailed and meaningful image features, such as object outlines, making it useful for various visual recognition tasks.

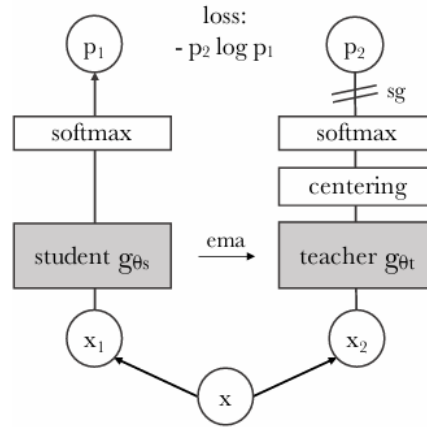


Figure 3.4: Self-distillation with no labels for one pair of views [10]. x_1 and x_2 represent a pair of views of the same image.

Leopart [68] focuses on learning an embedding space where patches containing the same object part form a group. This comes from the intuition that parts are more generalizable than objects because they don’t depend on specific object definitions.

They use a clustering pretext task to group the patch representations. A teacher generates the optimal cluster assignments and the student learns to predict them. Empirical evidence further supports their selection of a clustering pretext task, demonstrating that it outperforms contrastive pretext in dense prediction tasks. The extractor used is a **DINO**-pretrained **ViT** transformer.

The authors show that the representations learned by this method can be transferred to downstream semantic segmentation tasks, achieving better performance than existing **SOTA** methods on benchmarks like Pascal VOC, COCO-Thing, and COCO-Stuff [68].

3.3 Current Challenges

Although proven to be very useful for reducing the need for annotations and for learning more robust and generalizable feature representations of the data, **SSL** techniques pose some challenges.

Pretraining cost. Pretraining self-supervised models requires high computational resources and is quite time-consuming. This can make it challenging for smaller companies or institutions with fewer resources to be able to use these techniques effectively. Furthermore, energy consumption is high, causing environmental concerns [79].

Data requirements and curation. Image-related tasks often benefit from large datasets. Datasets like ImageNet are commonly used, but they have been curated. It’s uncertain whether this curation process might introduce generalizability issues, as the human involvement in organizing and selecting the data could result in a representation that is less reflective of real-world variability [79].

Architecture choice. Larger architectures often lead to better performance; however they are also much heavier to process and train. A common way to tackle this issue is by post-training distillation, which compresses large models into smaller, more efficient ones without significantly compromising performance [79].

Transferability. The goal of **SSL** techniques is to obtain robust features that can be applied to downstream tasks. Although this works well in recognition, when facing more complicated tasks such as dense prediction, performance is not as good. Due to this, there isn’t a benchmark method that is applicable to all types of tasks [79].

Pretask choice. The effectiveness of **SSL** is very dependent on the chosen pretext task. For text-related applications, masked modeling is quite effective. In vision, clustering and instance discrimination have been shown to provide advancements. The main consideration to be made is that the pretext task chosen should best relate to the data and capture its core characteristics in order to properly represent it in a robust manner [79].

3.4 Relevance to this Dissertation

In this dissertation, **SSL** is primarily relevant for the work developed in Chapter 6, where we extensively assess the use of **DINO ViT** features to enhance the robustness of polyp segmentation.

Chapter 4

Preliminary Experimental Work

As discussed in Chapter 1, data heterogeneity is a key factor affecting model performance and generalizability, limiting the adoption of AI solutions in clinical workflows.

This dissertation focuses on PolypGen [1], a multi-center dataset highly representative of common colonoscopy biases - different lighting conditions, populations, screening devices and thus labels in the images, blurring, visible medical devices, etc. (Figure 4.1).

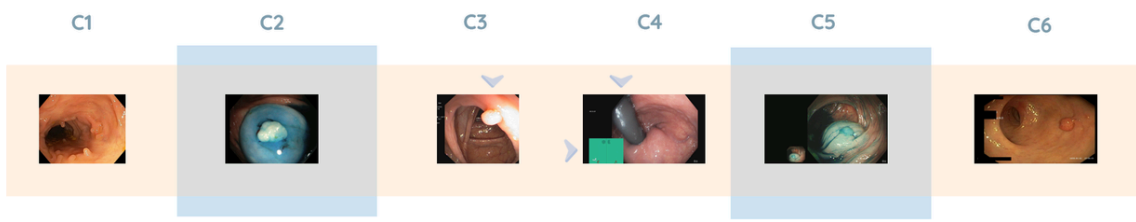


Figure 4.1: Visual examples of some of the data biases represented in the PolypGen dataset, including different light conditions, the inclusion of medical instruments in the images, labels, and chromoendoscopy data.

This chapter focuses specifically on verifying and addressing the chromoendoscopy bias, as it poses a significant source of variability in colonoscopy data and is often overlooked. We first introduce PolypGen and its main characteristics, followed by a more in-depth explanation of the chromoendoscopy technique and how it may hinder model performance. We also present main segmentation evaluation metrics, to be considered throughout the dissertation. The experimental part of this chapter is therefore focused on proving the existence of the chromoendoscopy bias, thus justifying the need for solutions to mitigate it. After this verification, we explore if traditional color preprocessing techniques can properly deal with this bias, or if there is a need to move to more advanced methods.

With this, the questions addressed by this chapter can be summarized into:

- Does chromoendoscopy introduce a bias that impacts generalizability?
- If so, can simple color preprocessing already reduce this bias?

4.1 Dataset: PolypGen

This section provides an overview of PolypGen’s main characteristics that make it the ideal choice for the purpose of this dissertation. Furthermore, it introduces in more depth the chromoendoscopy technique and its characteristics that make it a bias worth exploring.

The PolypGen dataset includes data from over 300 patients across six centers—Ambroise Paré Hospital (Paris), Instituto Oncologico Veneto (Padova), Centro Riferimento Oncologico, IR-CCS (Oslo), Oslo University Hospital, John Radcliffe Hospital (Oxford), and the University of Alexandria. With this, it comprehends different geographical anatomical distributions as well as data variability introduced by center-specific characteristics. Different centers often use different devices or protocols for this screening, all affecting image representation.

PolypGen incorporates both single-frame and sequence data. Single-frame data includes all detected lesions for each patient, while sequences consist of frames that follow a specific lesion or simply the mucosa (negative frames). The incorporation of data from multiple patients and centers enhances the ability to thoroughly test the generalizability of models across diverse patient populations and clinical settings [1].

The annotations for the dataset were carried out by a team of three researchers. Each researcher was assigned specific groups of frames, which were then cross-validated by senior gastroenterologists for precision. Frames with ambiguities or discrepancies were reviewed and corrected by joint annotation. The annotation process followed a strict protocol to reduce variability, focusing on accurate delineation of polyps, including handling occlusions and variations in polyp morphology [1].

Figure 4.2 has one example of single-frame images for each center in this dataset and its corresponding label, making clear the color and lighting differences, as well as variation in annotation types.

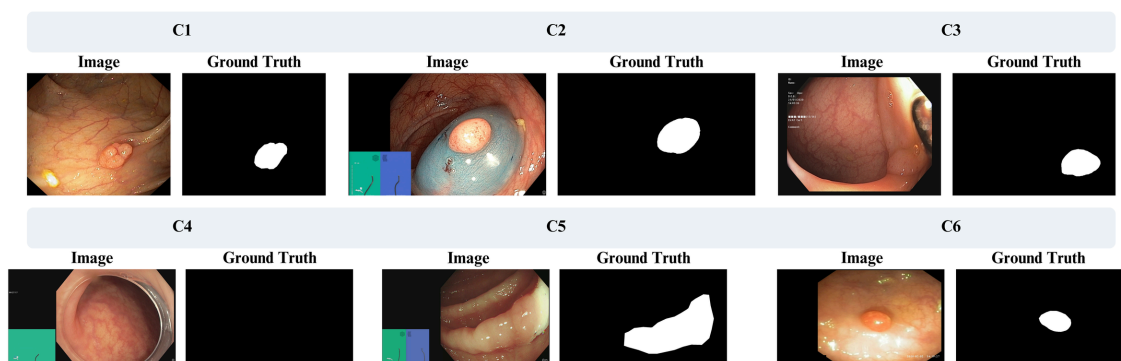


Figure 4.2: Examples of image/mask pairs for each center of the PolypGen dataset.

4.1.1 Chromoendoscopy

Chromoendoscopy is a diagnostic technique that involves the application of stains or dyes to enhance the location, characterization, and diagnosis of the lesion. It complements endoscopy by improving the contrast between normal and abnormal tissue, helping the endoscopist identify precancerous or early cancer regions for targeted biopsies or treatment [20].

The procedure is simple, with minor variations depending on the dye used. First, the area is washed with a catheter, then the dye is applied. Chromoendoscopy agents are divided into absorptive, reactive, and contrast stains. Common dyes used in chromoendoscopy include Lugol's iodine, methylene blue, indigo carmine, and crystal violet. Methylene blue is both in the contrastive and absorptive type, which often makes it preferable to other dyes. Healthy mucosa absorbs the dye more uniformly and abundantly, while areas with active inflammation or neoplasia absorb less. This difference in absorption creates a clearer contrast between normal tissue and lesions, making them easier to identify. Abnormal tissues, with fewer goblet cells and reduced cytoplasm, have a lower uptake of the contrast agent, leading to diminished dye absorption. [20, 80, 81].

With the focus on segmenting polyp instances in colonoscopy images, we focus on the single-frame data of the PolypGen dataset. In Figure 4.3, it is possible to see the distribution of regular colonoscopy and chromoendoscopy single-frame data across centers. Chromoendoscopy is in the minority and lacks representation in some centers, being present in only 283 samples out of 1537.

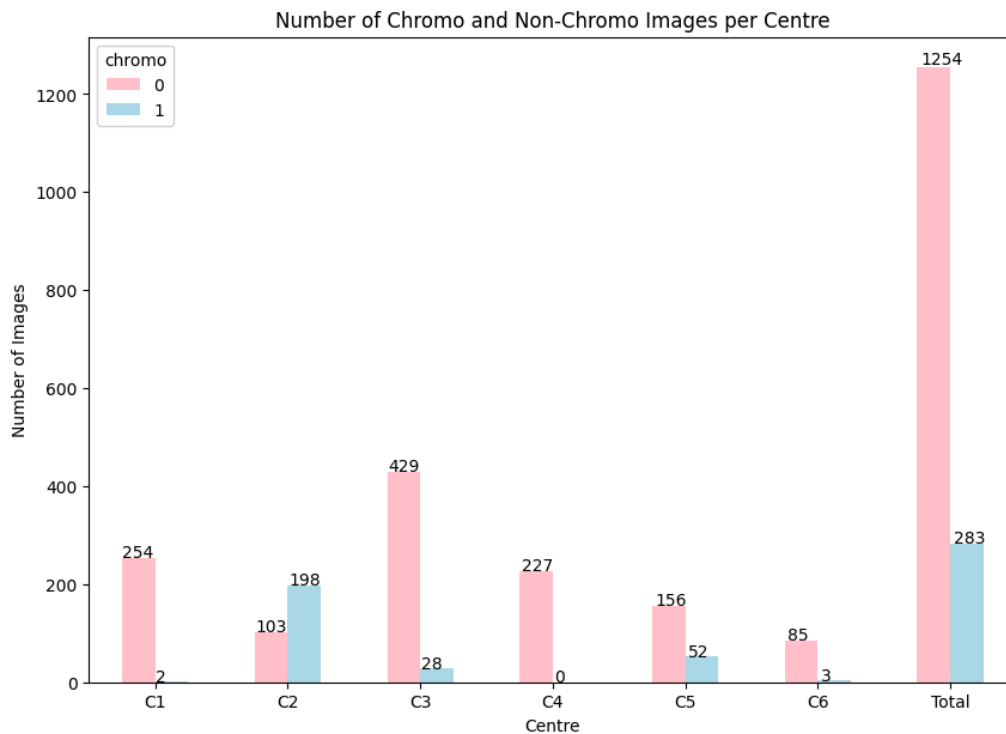


Figure 4.3: Distribution of chromoendoscopy and regular colonoscopy single-frame samples across centers. Blue or "chromo" is chromoendoscopy, whereas pink or "non-chromo" is regular colonoscopy.

4.2 Baseline Models

Having introduced the dataset and its main characteristics, we now describe the baseline models used for the experiments in this chapter.

4.2.1 U-Net

U-Net was introduced in Chapter 2. It consists of two main parts: an encoder (contracting path) and a decoder (expanding path). The encoder captures global context by progressively downsampling the image, and then the decoder upsamples it to provide precise localization for accurate segmentation, capturing fine details [13, 3].

4.2.2 DeepLabv3+

DeepLabV3+ [11], introduced more briefly in Chapter 2, includes spatial pyramid pooling and encoder-decoder structures in its architecture (Figure 4.4).

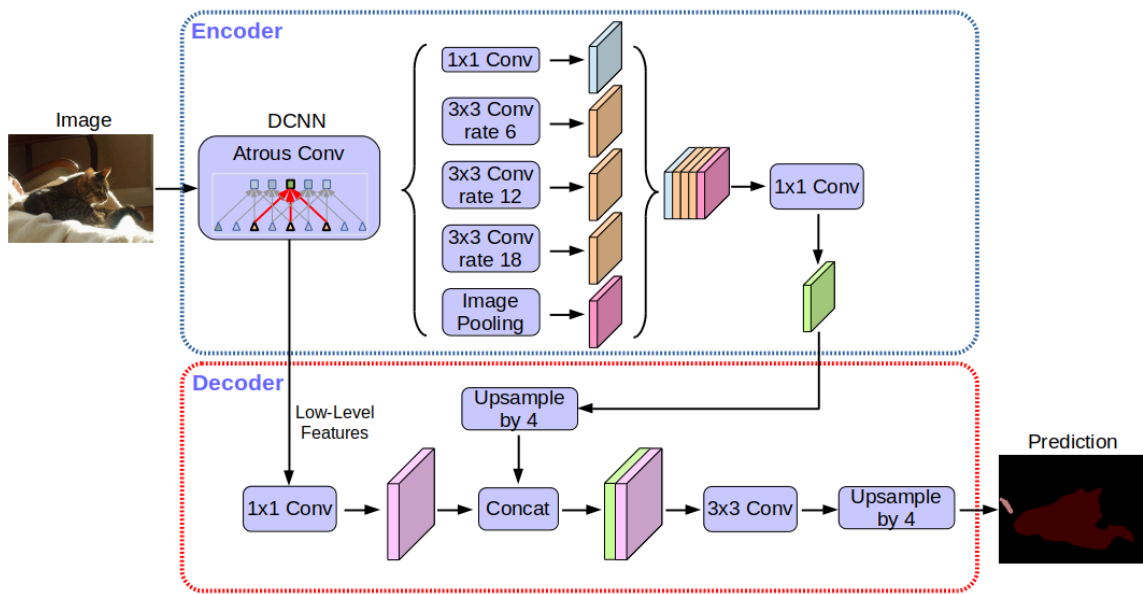


Figure 4.4: DeepLabV3+ Architecture [11]

Spatial pyramid pooling is used to capture contextual information from the input at various scales by distinguishing features through filters or pooling operations at different rates and sizes. The encoder compresses the image to a reduced representation, losing some spatial information in the process, and the decoder upsamples that representation, gradually reconstructing the original image size. In this rebuilding process, the network is capable of recovering sharper object boundaries, thus bettering the segmentation result.

The backbone for the DeepLabv3+ approach is Xception. The Xception model (Extreme Inception) is an improved version of the Inception architecture, designed to make convolution operations more efficient by using depthwise separable convolutions. Instead of applying a single

filter across all input channels in a standard convolution, this model breaks it down in two steps. The first step is defined by a depthwise convolution, where a separate convolution is performed over each channel of the input independently. Then, a pointwise convolution (1x1 convolution) combines the filtered information from all channels, helping the network understand how different features interact [11].

In Figure 4.5, it is possible to see a representation of the employed convolution operations. The atrous rate r is the stride used for input sampling. In regular convolution that we would have $r=1$ [11].

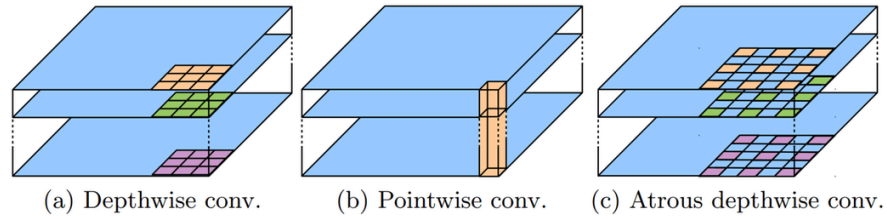


Figure 4.5: A 3x3 depthwise separable convolution decomposes a standard convolution into two parts: (a) a depthwise convolution, where each input channel is filtered independently, and (b) a pointwise convolution that combines the results from the depthwise step across all channels. In their work, Chen et al. investigate separable convolutions, where the depthwise convolution utilizes atrous convolution with a rate of 2, as shown in (c) [11].

To evaluate the performance of these models on polyp segmentation, we adopt standard segmentation metrics detailed in the following section.

4.3 Evaluation Metrics for Segmentation Problems

Common metrics for the evaluation of a segmentation model's performance include the **Dice Score Coefficient (DSC)**, recall, and pixel accuracy [13]. As the semantic segmentation task can be perceived as a pixel-wise classification, we can define

- **True Positive (TP)**: pixel classified as foreground (1) in the ground truth and correctly identified as foreground in the prediction.
- **True Negative (TN)**: pixel classified as background (0) in the ground truth and correctly identified as background in the prediction.
- **False Positive (FP)**: pixel incorrectly identified as foreground (1) in the prediction.
- **False Negative (FN)**: pixel incorrectly identified as background (0) in the prediction.

These four types of prediction pixels are then considered for the assessment of the segmentation metrics presented below.

The **DSC** is given by the formula:

$$DSC = \frac{2 \cdot |A \cap B|}{|A| + |B|} = \frac{2 \cdot TP}{(2 \cdot TP) + FP + FN} \quad (4.1)$$

where A and B denote the sets of pixels corresponding to the predicted segmentation and the ground truth, respectively. $|A \cap B|$ represents the number of common pixels of both sets, while $|A|$ and $|B|$ are the sizes of the respective sets. It is a measure of the overlap of the true class in the prediction and the ground truth mask, ranging $[0, 1]$, where 1 represents a perfect overlap.

The pixel accuracy defines the percentage of pixels of the image that were correctly classified.

$$\text{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.2)$$

Recall represents the proportion of **TP** correctly identified by the segmentation algorithm.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.3)$$

4.4 Assessing the Chromoendoscopy Bias

The primary goal of these experiments is to evaluate how the inclusion of chromoendoscopy samples in the training set affects model performance and generalization. So, in this section, we evaluate the performance of UNet [3] and DeeplabV3+ [11] when trained in three different configurations:

- On chromoendoscopy samples only (also referred to as "chromo" for simplicity);
- On regular colonoscopy samples only (also referred to as "non-chromo" for simplicity);
- On an equally represented combination of samples from both techniques (also referred to as "all" for simplicity).

4.4.1 Methodology

When training exclusively on chromoendoscopy samples, we use 190 chromo samples for the training set and 20 for the validation set. When training exclusively on regular colonoscopy samples, we use 190 non-chromo samples for training and 20 for validation. Finally, when training on samples from both techniques, we combine half of the chromo and half of the non-chromo samples in the training and validation sets. We ensure an equal number of images in each training configuration to properly isolate the type of technique as the primary variable being evaluated. In this way, the model's performance in a given configuration will be attributed to the technique used, rather than a higher number of training samples. In Figure 4.6, we can see the discussed data distribution across training, validation and test sets.

To properly assess generalizability, both **In-Distribution (ID)** and **OOD** testing are performed. Center 5, which has one of the highest representations of chromoendoscopy samples, was excluded from the training and validation sets and used as the **OOD** test set.

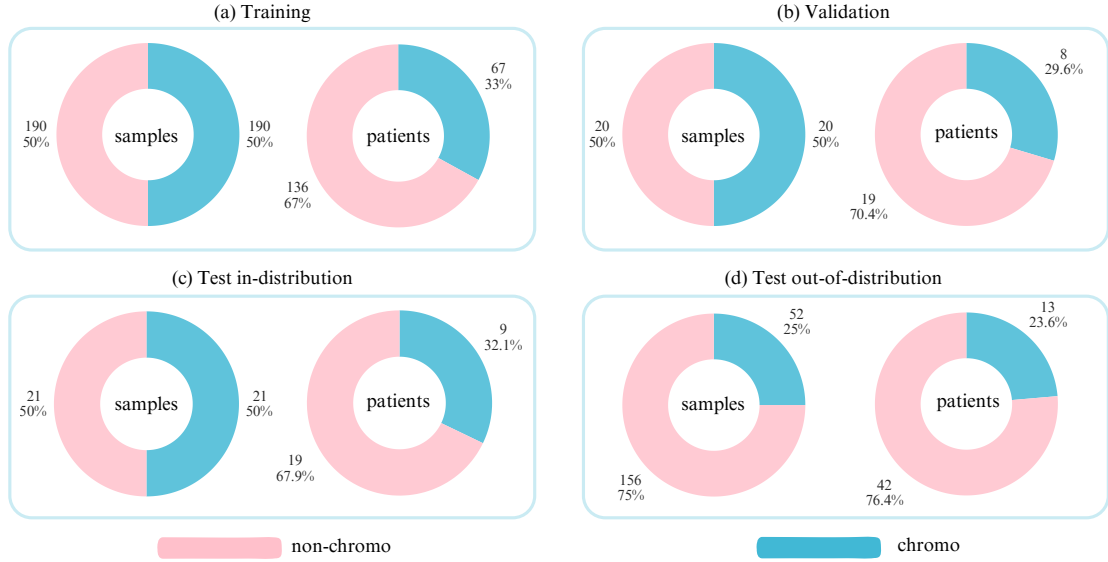


Figure 4.6: Visual representation of sample type and patient type distribution throughout the used splits.

After training, each configuration is evaluated on three separate test sets: chromo, non-chromo, and a combined "all" test set. This allows us to compare the performance of the models across different test conditions. Figure 4.7 illustrates the three training configurations when evaluated on chromoendoscopy samples.

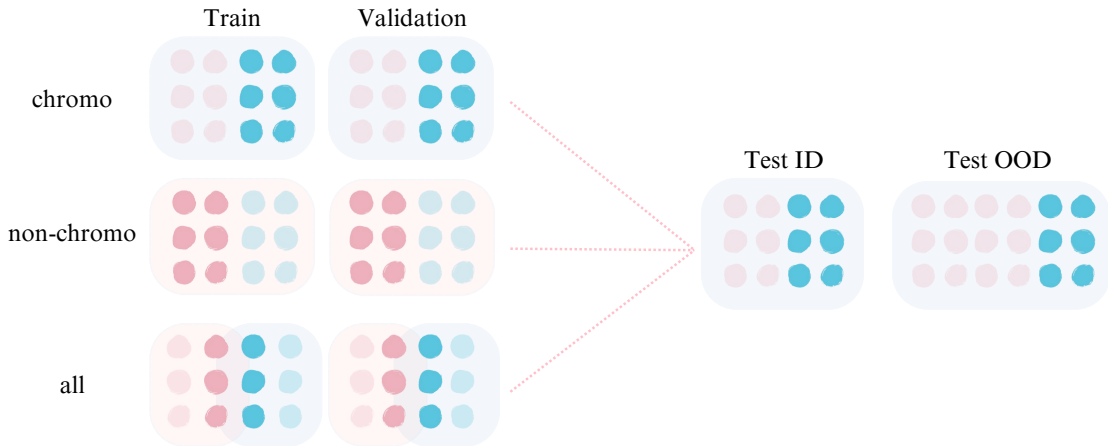


Figure 4.7: Example of the three training configurations being evaluated on the chromo test set.

Implementation details. The training parameters were kept constant for all experiments. The batch size used was 4, learning rate 10^{-6} and the models were trained for 300 epochs with the Adam optimizer [82] on NVIDIA RTX 2080Ti. The loss used was the standard dice loss (Equation

4.4). The inference checkpoint was selected based on the maximum validation **DSC** (Equation 4.3).

$$L_{\text{dice}} = 1 - \frac{2\sum_i p_i g_i + 0.1}{\sum_i p_i + \sum_i g_i + 0.1} \quad (4.4)$$

where:

- p_i denotes the predicted probability for pixel i ,
- g_i represents the ground truth label for pixel i ,
- 0.1 is a smoothing constant used to avoid division by zero.

4.4.2 Results and Discussion

In this subsection, we present and discuss the results of the experiments conducted.

Results with the U-Net baseline are shown in Tables 4.1, 4.2, and 4.3.

In Table 4.1, we compare the performance of the three training configurations on **ID** and **OOD** chromo samples. We can observe that the highest **ID** performance was achieved with chromo training. However, this trend does not hold for the **OOD** test set, where the best performance was observed when the model was trained on regular colonoscopy data. This suggests that the model’s generalizability is greatest when trained without the presence of chromoendoscopy samples. Therefore, the fact that chromo training achieves the best **ID** performance but the lowest **OOD** performance suggests that the model relies too heavily on center-specific features in the chromo configuration, causing poor generalization to data from other centers.

Table 4.1: U-Net performance on chromoendoscopy samples (in-distribution and out-of-distribution). The best performance is highlighted in **bold**.

Trained on	Validated on	Tested on	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
Chromo	Chromo	Chromo	0.5782	0.5399	0.9684	0.2237	0.2667	0.9443
Non-chromo	Non-chromo	Chromo	0.3777	0.4701	0.9074	0.2881	0.3807	0.9450
All	All	Chromo	0.4847	0.4237	0.9623	0.2688	0.2693	0.9547

Figures 4.8 and 4.9 show qualitative examples from the **ID** and **OOD** chromo test sets, comparing the visual performance of chromo and non-chromo training. The visual results reinforce the conclusions extracted with the quantitative evaluation.

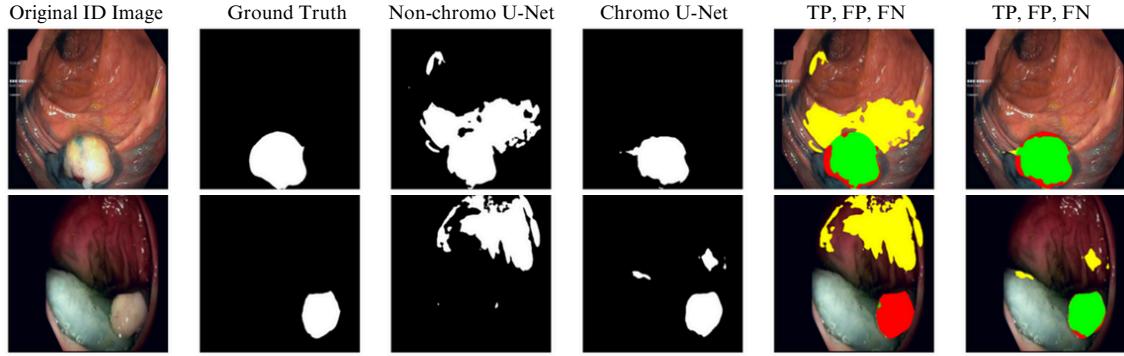


Figure 4.8: Qualitative examples of the chromoendoscopy ID test setting with the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).

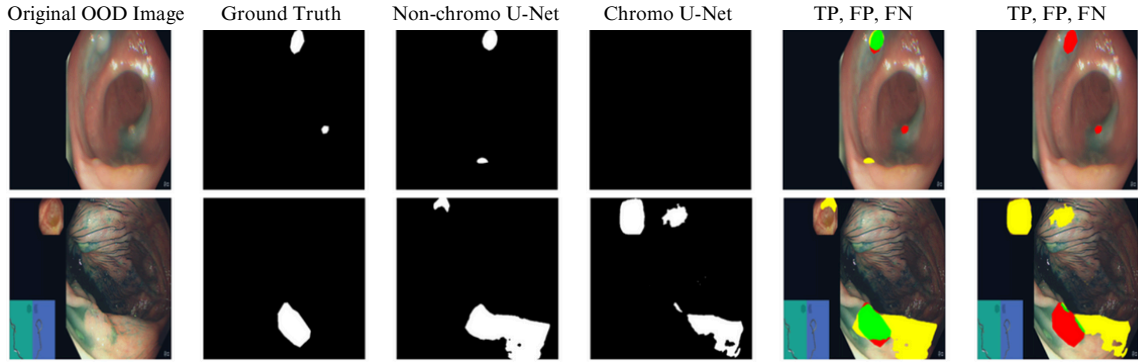


Figure 4.9: Qualitative examples from the chromo OOD test set using the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).

Looking now at Table 4.2, we can see the U-Net results for non-chromo testing. The best performance is obtained for regular colonoscopy data training in both **ID** and **OOD** test sets, and worse performance for the chromo configuration. These results are also consistent with the idea that including chromo samples in training results in a color-related bias, thus reducing generalizability.

Table 4.2: U-Net performance on regular colonoscopy samples. Best performance is depicted in **bold**.

Trained on	Validated on	Tested on	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
Non-chromo	Non-chromo	Non-chromo	0.6070	0.6517	0.9121	0.4085	0.4928	0.9345
Chromo	Chromo	Non-chromo	0.4180	0.3727	0.8955	0.2207	0.2101	0.9352
All	All	Non-chromo	0.5339	0.5670	0.8959	0.2365	0.2523	0.9358

Visual examples of **ID** and **OOD** non-chromo test samples are shown in Figures 4.10 and 4.11, respectively, which reflect the observed trend in the quantitative results.

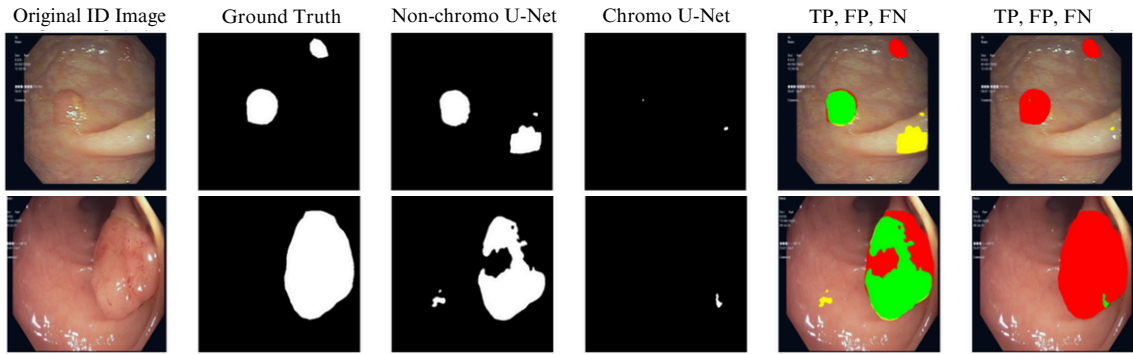


Figure 4.10: Qualitative examples of the colonoscopy ID test setting with the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).

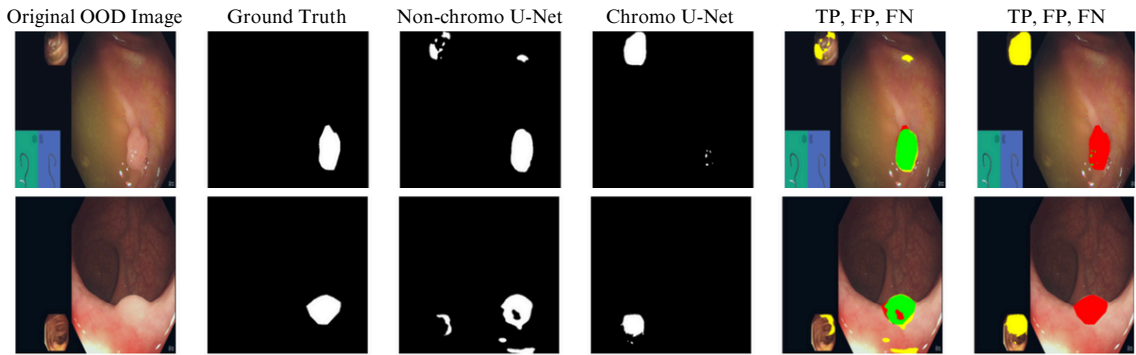


Figure 4.11: Qualitative examples of the colonoscopy OOD test setting with the U-Net model. Green represents true positives (TP), yellow false positives (FP) and red false negatives (FN).

When both types of samples are considered, the best-performing scenario remains non-chromo training, and worst chromo training, as shown in Table 4.3. This further reinforces the idea that including chromoendoscopy samples in the training process negatively impacts generalizability.

Table 4.3: U-Net performance on both types of samples. The best performance is depicted in **bold**.

Trained on	Validated on	Tested on	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
All	All	All	0.4527	0.3836	0.9070	0.2446	0.2565	0.9405
Chromo	Chromo	All	0.3950	0.3712	0.9107	0.2215	0.2242	0.9375
Non-chromo	Non-chromo	All	0.4439	0.5120	0.8991	0.3784	0.4647	0.9371

Tables 4.4, 4.5 and 4.6 the results for the DeepLabV3+ architecture [11]. The same trend is evident in the best-performing scenarios for this model, implying that when trained exclusively on chromoendoscopy data, the model overfits to this specific distribution. It appears to depend heavily on the blue color contrast rather than on more general and meaningful features to make its predictions. This could also explain why, when tested on OOD chromoendoscopy samples, the model performs better when trained solely on regular colonoscopy data.

Table 4.4: DeepLabV3+ performance on chromoendoscopy samples. Best performance is depicted in **bold**.

Trained on	Validated on	Tested on	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
Chromo	Chromo	Chromo	0.5384	0.6273	0.9475	0.1018	0.1226	0.9233
Non-chromo	Non-chromo	Chromo	0.3757	0.5292	0.8820	0.1660	0.2486	0.9432
All	All	Chromo	0.4640	0.4895	0.9554	0.1165	0.2089	0.9160

Table 4.5: DeepLabV3+ performance on regular colonoscopy samples. Best performance is depicted in **bold**.

Trained on	Validated on	Tested on	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
Non-chromo	Non-chromo	Non-chromo	0.4365	0.5171	0.8743	0.2540	0.3143	0.9213
Chromo	Chromo	Non-chromo	0.2313	0.2356	0.8625	0.0965	0.0881	0.9050
All	All	Non-chromo	0.3266	0.3129	0.8767	0.2046	0.2322	0.9163

Table 4.6: DeepLabV3+ performance on both types of samples. Best performance is depicted in **bold**.

Trained on	Validated on	Tested on	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
All	All	All	0.3563	0.3459	0.8999	0.1826	0.2264	0.9163
Chromo	Chromo	All	0.3373	0.3501	0.8932	0.0978	0.0967	0.9096
Non-chromo	Non-chromo	All	0.3584	0.4418	0.8657	0.2321	0.2979	0.9267

Comparing U-Net’s and DeepLabV3+’s results, we can conclude that U-Net consistently outperforms DeepLabV3+ across test sets. The relative drop between **ID** and **OOD** performance is relatively high in both models, showing limited generalizability even in the absence of chromoendoscopy samples during training.

We can conclude that these experiments help validate the bias introduced by chromoendoscopy samples in deep learning-based segmentation models. With this bias confirmed, the next step is to explore whether color-related preprocessing techniques can mitigate this issue and improve model generalization.

4.5 Impact of Color Preprocessing on Baseline Model Performance

The last section ended with the confirmation that chromoendoscopy introduces a significant bias that degrades model performance. In this section, we assess whether the introduction of color-related preprocessing techniques can already help mitigate this bias and improve performance.

4.5.1 Data Preparation

The next experiments are conducted using two different train-validation-test splits. The first split is the same as used in the previous section but used in its entirety, i.e., without chromo / non-chromo sample selection (Figure 4.6). The second split, shown in Figure 4.12, includes a larger number of total samples, ensuring that chromoendoscopy samples and chromoendoscopy patients are evenly represented across all sets. The key difference between the two is the number of chromoendoscopy and colonoscopy samples. In the first split, we ensure an equal number of both techniques in the training, validation, and ID test sets. Since chromoendoscopy is a minority in PolypGen, that led to a lower total number of samples considered. In the second split, we maintain a balanced percentual representation of chromoendoscopy samples across the training, validation, and ID test sets, without focusing on the exact count of samples. It was ensured that same-patient samples are kept within the same split in order to avoid any data leakage.

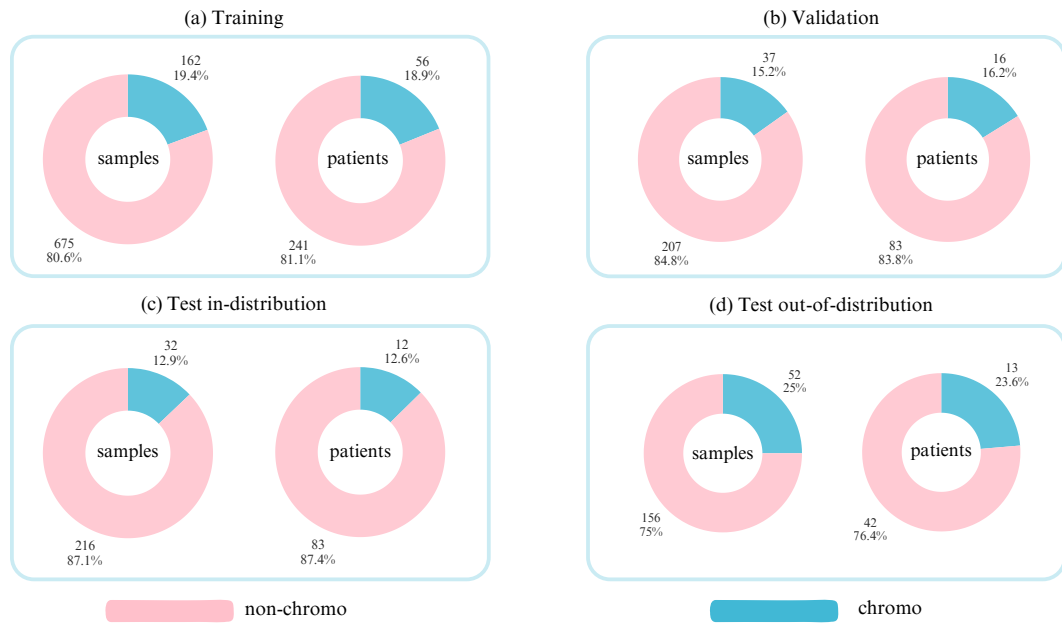


Figure 4.12: Second train-val-test splits used for the color pre-processing experiments.

4.5.2 Methodology

Figure 4.13 shows the explored color pre-processing techniques for these experiments. The original dataset is kept for control comparison. The grayscale transformation removes the color

information from the data, promoting the model to focus more on the structure rather than color information, which could reduce the color bias. HSV separates the image into three distinct components: Hue (representing color type), Saturation (indicating color intensity), and Value (corresponding to brightness). In contrast, in the RGB space, the color information is interdependent across all three channels. This separation may simplify the isolation and modification of the color component with the highest relationship with the blue dye (such as the hue). The rescaling operation adjusts the color and intensity values to a consistent range, helping to standardize the color information across all images. Channel Inversion flips the intensities of each color channel, which could highlight details that otherwise are unrepresented in the original RGB images. **Contrast-Limited Histogram Equalization (CLAHE)** improves the overall contrast of the image by adjusting it in an adaptive way. This transformation helps to mitigate inconsistencies or unwanted variations throughout the image. Grayscale and **CLAHE** were also combined to remove the color information and also enhance the structural detail in the images. Finally, the green channel was isolated as it was empirically demonstrated to exhibit the greatest invariance with respect to the chromoendoscopy technique.

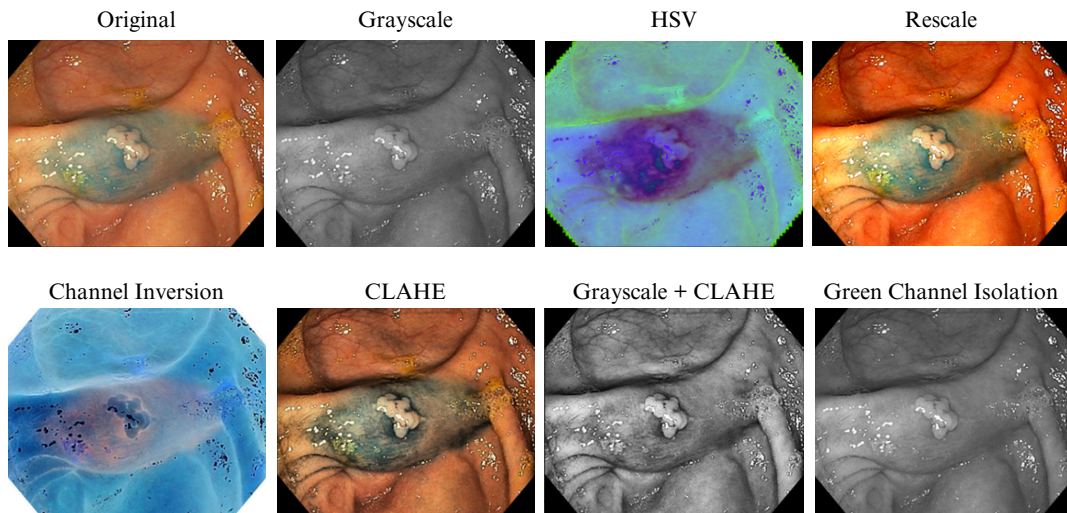


Figure 4.13: Considered color transformations.

Implementation details. Implementation details for this section are the same as in Section 4.4.

4.5.3 Results and Discussion

Table 4.7 presents the results for U-Net trained with the different color preprocessing techniques, using the first introduced splits.

The best performance is observed when applying the grayscale transformation, either as a standalone method or in combination with other transformations. This aligns with the initial intuition that the removal of the color information may promote the model to focus more on the structural information. By shifting its focus from color to structure, the color bias is, therefore, minimized.

CLAHE shows to be a good complementary transformation, enhancing the quality of the structures presented in the images. In general, the pre-processing techniques improve baseline **ID** and **OOD** results, thus proving to be effective for chromoendoscopy bias mitigation when applied as a pre-processing step in U-Net.

Table 4.7: U-Net results for the color pre-processing experiments using the first splits. The best performance is highlighted in **bold**, while results surpassing the baseline (no transformation) are underlined.

Transformation	In-Distribution (ID)			Out-of-Distribution (OOD)		
	Dice	Recall	Accuracy	Dice	Recall	Accuracy
None	0.6119	0.6023	0.9471	0.3803	0.3849	0.9511
Grayscale	0.6857	<u>0.6274</u>	0.9539	0.3798	<u>0.4103</u>	0.9499
HSV	0.6101	<u>0.6099</u>	0.9433	0.2899	0.3325	0.9314
Rescale (0-255 to 50-200)	0.5975	<u>0.6095</u>	0.9413	0.3735	<u>0.4253</u>	0.9440
Channel Inversion	0.5998	0.5793	<u>0.9483</u>	<u>0.3881</u>	0.3844	0.9548
CLAHE	<u>0.6621</u>	<u>0.6745</u>	<u>0.9519</u>	<u>0.3889</u>	<u>0.4308</u>	0.9413
Grayscale + CLAHE	<u>0.6453</u>	<u>0.6376</u>	0.9421	<u>0.3857</u>	<u>0.4079</u>	<u>0.9518</u>
Isolated G + grayscale	0.6063	<u>0.6184</u>	<u>0.9472</u>	0.3786	<u>0.4395</u>	0.9452
Isolated G + grayscale + CLAHE	<u>0.6475</u>	<u>0.6603</u>	<u>0.9482</u>	0.4336	0.4824	0.9508
Isolated G + swap distribution	<u>0.6216</u>	<u>0.6363</u>	0.9431	<u>0.3932</u>	<u>0.4589</u>	0.9473
Isolated G + swap distribution + CLAHE	<u>0.6592</u>	0.6753	<u>0.9512</u>	<u>0.3886</u>	<u>0.4547</u>	0.9477

Table 4.8 shows the results with the DeepLabV3+ architecture [11]. In this case, the color transformations did not show as much improvement as they did with the U-Net architecture. This can be attributed to the architectural differences between the two models and how they process the images. U-Net is a convolutional network designed specifically for segmentation, making it highly responsive to local variations in image features like color. In contrast, DeepLabV3+ utilizes atrous convolution (Figure 4.14), which may make it less sensitive to subtle color variations. Nevertheless, the Grayscale + **CLAHE** setting was still able to surpass the control in the **OOD** case.

Table 4.8: DeepLabV3+ results for the color pre-processing experiments using the first splits. The best performance is shown in **bold**, and performances surpassing the baseline (no transformation) are underlined.

Transformation	In-Distribution (ID)			Out-of-Distribution (OOD)		
	Dice	Recall	Accuracy	Dice	Recall	Accuracy
None	0.5788	0.6138	0.9275	0.2890	0.3023	0.9500
Grayscale	0.4725	0.5390	<u>0.9315</u>	0.2540	0.2786	0.9383
HSV	0.5246	0.4986	<u>0.9309</u>	0.2131	0.2569	0.9272
Rescale (0-255 to 50-200)	0.4478	0.5320	0.9106	0.2449	0.2781	0.9337
Channel Inversion	0.5741	0.5965	<u>0.9303</u>	0.2667	<u>0.3095</u>	0.9393
CLAHE	0.5778	0.5364	0.9375	0.2626	0.2594	0.9484
Grayscale + CLAHE	0.5106	0.6042	0.9225	0.3042	0.3707	0.9375
Isolated G + grayscale	0.4508	0.4789	0.9162	0.2423	0.2998	0.9333
Isolated G + grayscale + CLAHE	0.4936	0.5522	0.9274	0.2855	<u>0.3481</u>	0.9306
Isolated G + swap distribution	0.4821	0.5220	0.9267	0.2425	0.2909	0.9227
Isolated G + swap distribution + CLAHE	0.5041	0.5468	<u>0.9322</u>	0.2431	0.3007	0.9272

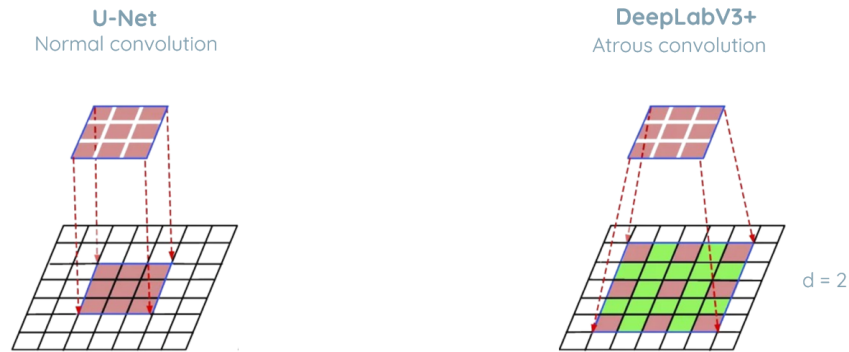


Figure 4.14: Visual comparison of the convolution operation computed in the U-Net model with the dilated convolution present in DeepLabV3+.

Tables 4.9 and 4.10 present the quantitative color pre-processing results for the U-Net and DeepLabV3+ architectures second splits, introduced in this section.

For U-Net, the grayscale transformation and its variations continue to be the best performing pre-processing techniques. However, compared to the first splits, the improvements are less pronounced. This is likely due to the reduced relative presence of chromoendoscopy samples in the second splits, which diminishes the color bias impact. For DeepLabV3+, the grayscale + **CLAHE** combination and single **CLAHE** transformation generally overperform the baseline, proving to be the best pre-processing choices for this architecture.

Table 4.9: U-Net results for the color pre-processing experiment using the second splits, built with the entire PolypGen dataset. The best performance is shown in **bold**, and performances surpassing the baseline (no transformation) are underlined.

Transformation	In-Distribution (ID)			Out-of-Distribution (OOD)		
	Dice	Recall	Accuracy	Dice	Recall	Accuracy
None	0.6595	0.6168	0.9614	0.4707	0.5030	0.9557
Grayscale	0.6457	0.6351	0.9605	0.5203	0.6115	<u>0.9596</u>
HSV	0.6131	0.5547	0.9576	0.3983	0.4254	0.9483
Rescale (0-255 to 50-200)	0.6478	0.5645	0.9585	0.4469	0.4844	0.9553
Channel Inversion	0.6519	0.5941	0.9623	<u>0.5025</u>	<u>0.5252</u>	<u>0.9621</u>
CLAHE	0.6609	0.5987	0.9587	<u>0.4947</u>	<u>0.5002</u>	<u>0.9577</u>
Grayscale + CLAHE	0.6258	0.5615	0.9583	<u>0.4996</u>	<u>0.5128</u>	0.9641
Isolated G + grayscale	0.6515	<u>0.6305</u>	0.9610	<u>0.4781</u>	<u>0.5694</u>	0.9556
Isolated G + grayscale + CLAHE	0.6107	0.5734	0.9577	<u>0.5089</u>	<u>0.5693</u>	<u>0.9614</u>
Isolated G + swap distribution	0.6404	0.5871	0.9597	<u>0.4992</u>	<u>0.5427</u>	<u>0.9630</u>
Isolated G + swap distribution + CLAHE	0.6283	0.6067	0.9581	<u>0.4972</u>	<u>0.5837</u>	<u>0.9592</u>

Table 4.10: DeepLabV3+ results for the color pre-processing experiment using the second splits. The best performance is shown in **bold**, and performances surpassing the baseline (no transformation) are underlined.

Transformation	In-Distribution (ID)			Out-of-Distribution (OOD)		
	Dice	Recall	Accuracy	Dice	Recall	Accuracy
None	0.5662	0.5269	0.9542	0.3717	0.4021	0.9554
Grayscale	0.5093	0.4636	<u>0.9502</u>	0.3452	0.3715	0.9502
HSV	0.5655	<u>0.5443</u>	0.9512	0.3678	<u>0.4394</u>	0.9449
Rescale (0-255 to 50-200)	0.5552	<u>0.5453</u>	0.9502	0.3511	0.3981	0.9434
Channel Inversion	0.5390	<u>0.5312</u>	0.9472	0.3570	0.3988	0.9402
CLAHE	0.5423	0.5250	0.9523	0.3670	<u>0.4425</u>	0.9462
Grayscale + CLAHE	0.4980	0.5496	0.9411	0.3896	0.5050	0.9337
Isolated G + grayscale	0.5094	0.4781	0.9479	0.3415	0.3948	0.9453
Isolated G + grayscale + CLAHE	0.5056	<u>0.5319</u>	0.9430	0.3565	<u>0.4374</u>	0.9368
Isolated G + swap distribution	0.4844	0.4562	0.9422	0.3627	<u>0.4453</u>	0.9362
Isolated G + swap distribution + CLAHE	0.5071	0.4825	0.9463	0.3637	<u>0.4214</u>	0.9453

The results show that segmentation performance can be improved using color-related pre-processing techniques. These methods help reduce the color bias introduced by chromoendoscopy samples, leading to more accurate segmentation. The most effective technique was the combination of grayscale, which shifts the focus from color to image structure, and **CLAHE**, which enhances structural clarity and contrast.

However, comparing the results of U-Net and DeepLabV3+ reveals that the effectiveness of these pre-processing techniques differs between architectures. This suggests that each model may respond differently to color transformations due to their distinct architectures and how they process image data. Therefore, when applying these techniques for bias mitigation, they would require customization to the specific model being used. This is not ideal, as the goal is to have a standard, reliable technique that consistently mitigates bias across different architectures. The variability in performance improvements highlights the challenge of finding a universal solution that works well for all models, emphasizing the need for more generalizable and robust approaches that can effectively handle bias regardless of the underlying architecture.

4.6 Conclusions

This chapter provided an overview of the models, dataset, and evaluation metrics used in the experiments. The preliminary experiments focused on evaluating the bias introduced by chromoendoscopy samples and testing color-related preprocessing techniques as potential solutions to mitigate this bias.

Through these experiments, we confirmed the presence of this bias and investigated whether simple color preprocessing methods could alleviate it. We found that color-related preprocessing techniques can indeed be beneficial with certain combinations, particularly those involving grayscale transformations. However, we also observed that the effectiveness of these methods varies depending on the architecture, which indicates that they cannot be universally applied as a robust solution for bias mitigation. This highlights the need for more reliable solutions to enhance model robustness against this bias, paving the way for exploring more generalizable approaches in the next chapter.

Chapter 5

Color Invariance Methods

Chapter 4 demonstrates the significance of the color bias introduced by chromoendoscopy samples, which hinders model performance, especially when applied to OOD data. The experiments performed also show that traditional color-related preprocessing techniques offer limited and inconsistent improvements across model architectures. This motivated the search for more robust mitigation strategies, giving light to the current chapter, where we investigate color invariance methods.

Color invariance methods are augmentation strategies that aim to promote the model to focus less on the color information of the input during training. The explored strategies include generating a paired dataset using a CycleGAN model, color swapping between input samples, and randomly erasing color information from a portion of the input.

These methods are evaluated and compared using UM-Net [4], the current SOTA model in the polyp segmentation field introduced in Chapter 2. U-Net is also retained for analysis to provide continuity with prior results and to provide a comprehensive analysis of these methods across different architectures. DeepLabV3+, on the other hand, was excluded from further analysis due to its consistent underperformance compared to U-Net, seen in the previous Chapter.

With this, the question we aim to answer during this chapter can be written as:

- Can color invariance methods effectively mitigate the chromoendoscopy bias?

5.1 CycleGAN for Paired Dataset Creation

The CycleGAN model [83] was explored with the purpose of obtaining a paired dataset—for each sample, a chromoendoscopy and normal colonoscopy version. The intuition behind this method was that by training a segmentation model with paired samples in both domains, paralleled with the same segmentation ground truth, we could promote technique invariance and, therefore, increased robustness.

The pipeline is illustrated in Figure 5.1. This example illustrates translation from domain B (chromoendoscopy) to domain A (regular colonoscopy).

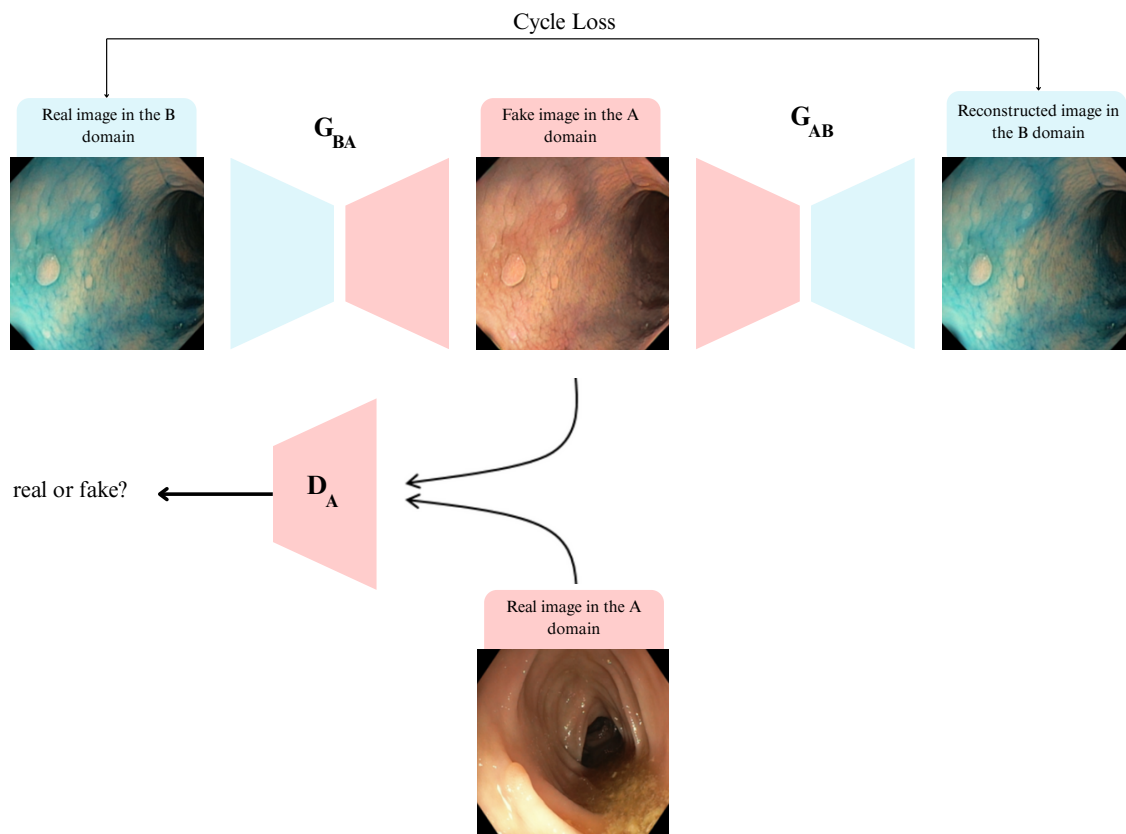


Figure 5.1: CycleGAN architecture. G_{AB} translates images from domain A to domain B and G_{BA} performs the inverse mapping. D_A is the discriminator for domain A.

CycleGAN is a type of Generative Adversarial Network (GAN) for inter-domain translation, designed mainly for unpaired datasets. With this model, it is possible to generate the version of an instance from the source domain as if it were originally from the target domain. A common example would be to generate horse images from a zebra and vice versa. The main novelty it introduces is that it does not require a paired dataset in order to learn the two mapping functions between domains. Unpaired translation is a much harder task than paired translation, since there is no direct correspondence between the images from both domains. The CycleGAN can overcome this difficulty by introducing the cycle consistency and identity losses. The cycle consistency loss evaluates the consistency of the consecutive mapping between the two domains. Considering the example presented on Figure 5.1, the real image from the **B** domain is processed by G_{BA} , that outputs its fake version from domain **A**. This fake version is then passed through the antagonist generator, which should generate an identical image to the original. The cycle loss penalizes the inconsistencies found between the original image **B** and its reconstructed version. If a generator is built to perform a mapping between a source domain and a target domain, when receiving a target domain image as input, it should generate the exact same image. This constraint is enforced by the identity loss.

The adversarial loss is a key feature of GAN architectures, representing the "game" between generators and discriminators. Generators strive to produce more realistic images, while discriminators aim to distinguish real from generated images. The loss penalizes the discriminator for misclassifying images and the generator when its images are correctly recognized as fake.

In sum, a CycleGAN is a junction of two GANs constrained with consistency losses to learn mappings between two different domains, without the need for paired data.

5.1.1 Implementation Details

The training dataset used for the CycleGAN was the same as in Chapter 4 Figure 4.12. It was divided in domain A samples and domain B samples, corresponding to regular colonoscopy and chromoendoscopy, respectively. Mean Squared Error (MSE) loss was used as the adversarial loss. Also called L2 loss, it measures the squared difference between the pixel values of a predicted image and the target image (Equation 5.1). The base implementation was extracted from [PyTorch-CycleGAN GitHub Repository](#).

$$L_2(\hat{Y}, Y) = \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (5.1)$$

The L1 loss was considered for cycle and identity consistency (Equation 5.2).

$$L_1 = \sum_{i=1}^n |y_{true} - y_{predicted}| \quad (5.2)$$

The models were trained on NVIDIA RTX 8000 for 200 epochs using an Adam optimizer, with a batch size of 4 and a learning rate of 2×10^{-4} , decaying linearly from epoch 100. The global loss is the sum of the individual losses, weighted by 5 for identity, 10 for cycle consistency, and 1 for adversarial loss. The generators' architecture follows an encoder-decoder logic based on 9 ResNet blocks. We also used PatchGAN discriminators consisting of 4 sequential convolutional blocks followed by a final classifier.

5.1.2 Qualitative Results

Results of G_{AB} are shown in Figure 5.2. Annotations in the generated images are often blurred or inconsistent, such as displaying a different date compared to the original. Furthermore, the application of the dye is not consistently around the polyp area. Although this is also true for some real chromoendoscopy images, it would still be desirable to promote a more exact application.

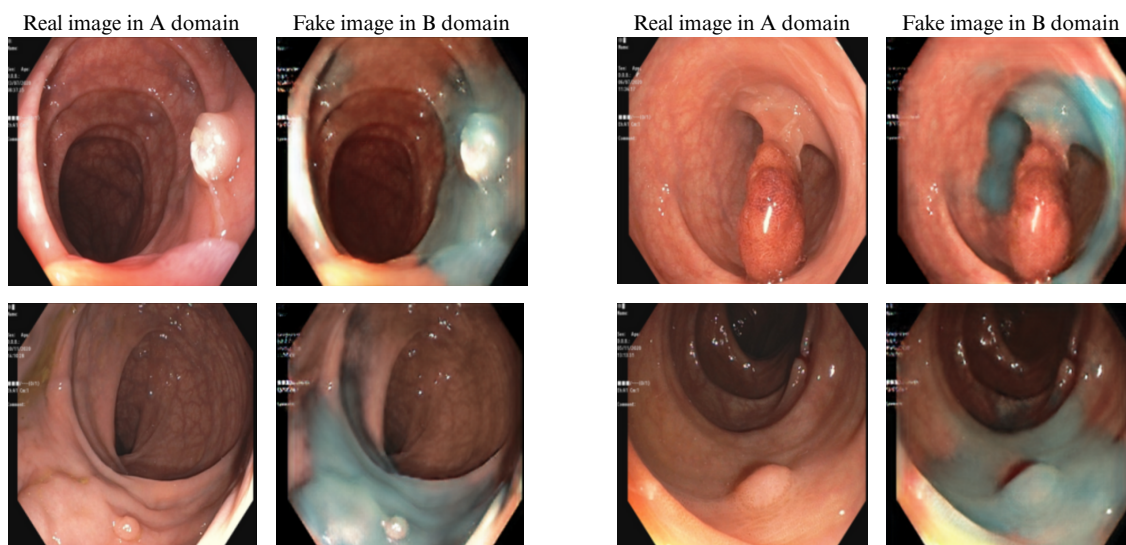


Figure 5.2: Examples of $A \rightarrow B$ image generation. The **A** domain corresponds to the normal colonoscopy data domain, and the **B** domain represents the chromoendoscopy domain.

As for the $B \rightarrow A$ domain mapping, as seen in Figure 5.3, the results were more satisfactory. While some structural loss remains, particularly in the annotations, the generated images are closer to what is expected from a typical colonoscopy.

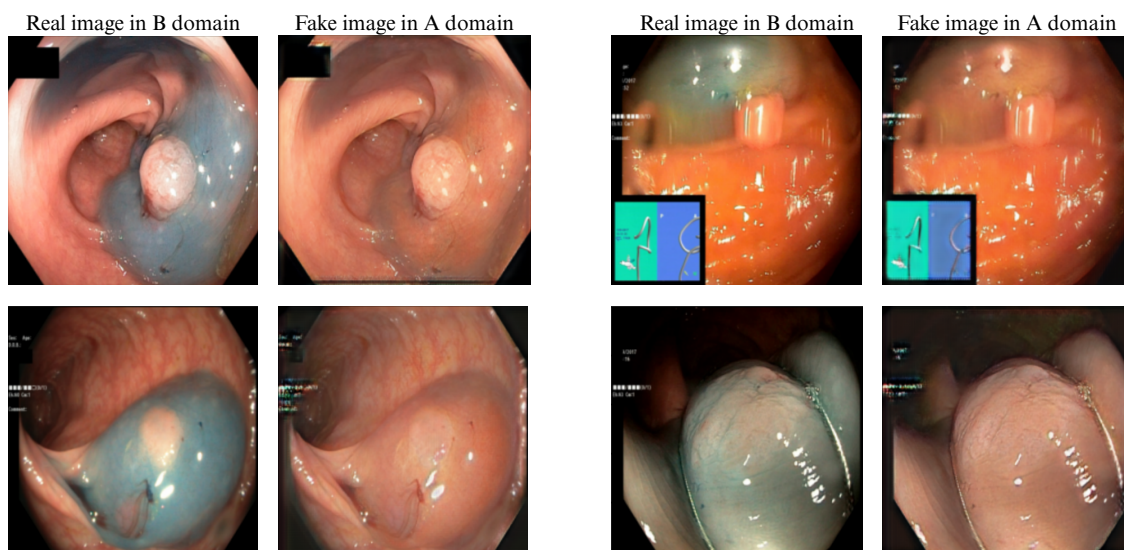


Figure 5.3: Examples of BA image generation. A domain corresponds to the normal colonoscopy data domain, and B domain represents the chromoendoscopy domain.

Although the CycleGAN results exhibit promising domain translation, challenges remain in accurately preserving structural details and annotations, especially in generated chromoendoscopy images. This highlights areas for further possible refinement.

5.2 Color Transfer

The **Color Transfer (CT)** operation, introduced by Xiuquan et al. [4] as an improvement on the UM-Net model, aims to promote color invariance during model training. The operation results in the swapping of color distributions between a pair of images. It is repeated multiple times during training in order to show the model that the color distributions it encounters are independent of the content of the input.

The operation goes as follows:

1. Both images are converted from the RGB space to the LAB space.
2. The mean (μ) and variance (σ^2) of each image are computed.
3. Each image is mapped to the LAB space of the other according to

$$\text{new Lab1} = \frac{\text{Lab1} - \text{mean1}}{\text{std1}} \times \text{std2} + \text{mean2} \quad (5.3)$$

$$\text{new Lab2} = \frac{\text{Lab2} - \text{mean2}}{\text{std2}} \times \text{std1} + \text{mean1} \quad (5.4)$$

4. The image intensities are clipped to the [0, 255] range.
5. The LAB-swapped images are reconverted to the RGB space.

Figure 5.4 shows examples of **CT** applied between two images from the same technique. For regular colonoscopy images, the color style swapping is seamless and produces realistic results. In contrast, the results on chromoendoscopy samples are less visually accurate due to the blue dye. In this particular example, there is also noise introduced by the colored annotations, which ends up being the main factor contributing to a diminished structural clarity in the swapped images.

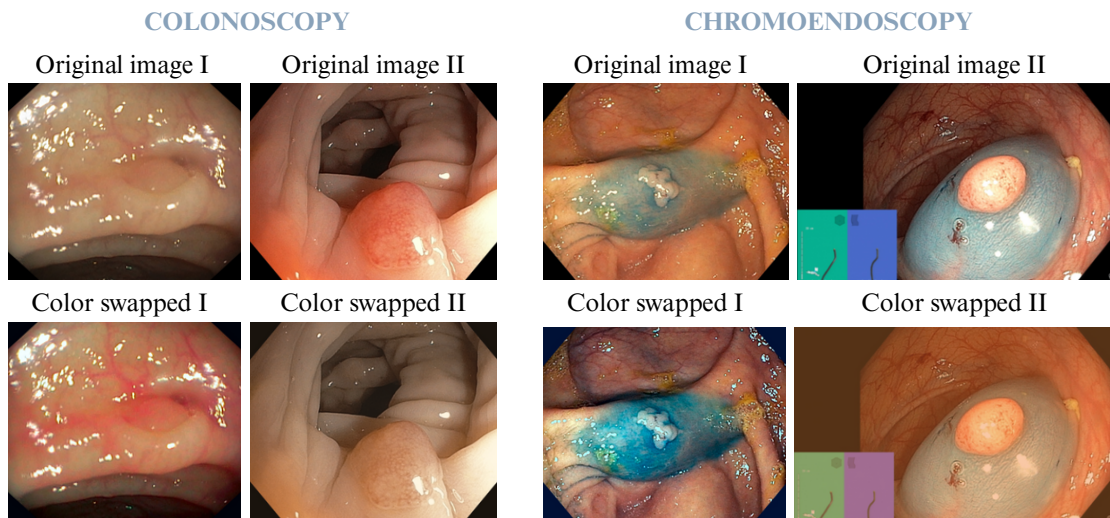


Figure 5.4: Examples of the color transfer operation between same-technique samples.

Finally, we also present an example for inter-technique color transfer in Figure 5.5. The results are more realistic than the latter example; however, there is no effective visual mitigation of the blue dye in the chromoendoscopy image.

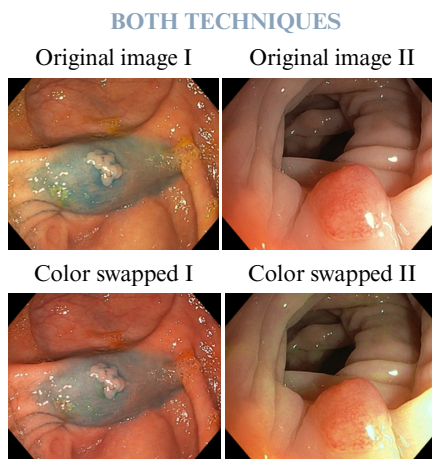


Figure 5.5: Examples of the color transfer operation between different-technique samples.

From the qualitative analysis, we conclude that this technique leads to realistic outputs; however, its performance is influenced by the presence of colored non-anatomical elements. Furthermore, it does not effectively swap chromoendoscopy-related elements as the CycleGAN does.

5.3 Random Color Erasing

Random Color Erasing (RCE) [84] tries to promote color invariance by randomly removing it from the input images. This removal consists of replacing a variable rectangle-shaped portion of the original RGB image by its grayscale version. In the original implementation, this can be performed globally or locally; however, for this experiment, only the local **RCE** was considered. One example of this is present in Figure 5.6.

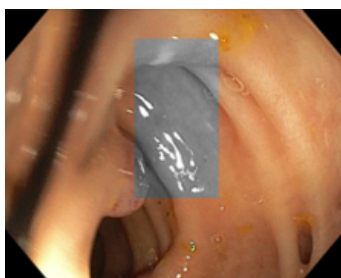


Figure 5.6: Example of Random Color Erasing (RCE) in a polypgen sample.

The intuition behind this is that by randomly erasing color information, the model may learn to focus on more structural elements, such as textures and shapes, rather than relying heavily on color

cues. This, therefore, would make the model more robust to situations where color information may be inconsistent or missing.

5.4 Ablation of the Different Color Invariance Techniques

This section presents an ablation study comparing **CT** with **RCE**, and **CT** against the addition of CycleGAN-generated samples (GS) during training for both U-Net and UM-Net models. It is important to note that only the training set is augmented with generated samples; the validation and test sets remain unchanged. We chose to perform this ablation study in pairs of color invariance techniques, allowing for the fair analysis of their combined effect whilst limiting complexity.

For U-Net, the best performance is generally achieved when both **CT** and **RCE** are applied, whereas the lowest performance is typically when there is no color invariance method applied.

On the other hand, for UM-Net, the addition of **CT** and **RCE** simultaneously does not show a noticeable improvement. It seems to benefit the most from the **CT** operation alone, especially in **OOD**, where the introduction (both alone and with **CT**) of **RCE** hinders performance.

Table 5.1 presents the **CT** and **RCE** ablation results.

Table 5.1: Results of U-Net and UM-Net Performance with Color Transfer (CT) and Random Color Erasing (RCE). Best performance is depicted in **bold**, and the second-best performance is underlined.

Model	CT	RCE	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
U-Net	×	×	0.6595	<u>0.6168</u>	0.9614	0.4707	<u>0.5030</u>	0.9557
	✓	×	0.6873	0.5997	0.9624	<u>0.4942</u>	0.4703	0.9643
	×	✓	<u>0.6734</u>	0.5931	<u>0.9631</u>	0.4734	0.4678	0.9605
	✓	✓	0.6698	0.6285	0.9634	0.5016	0.5214	<u>0.9618</u>
UM-Net	×	×	0.7540	<u>0.7432</u>	0.9690	0.6391	<u>0.6985</u>	0.9725
	✓	×	<u>0.7639</u>	0.7548	0.9685	<u>0.6203</u>	0.7158	<u>0.9715</u>
	×	✓	0.7754	0.7286	0.9726	0.5831	0.6479	0.9700
	✓	✓	0.7593	0.7329	<u>0.9713</u>	0.6158	0.6962	0.9725

Table 5.2 shows the ablation results for the combination of **CT** and generated samples from the CycleGAN.

U-Net appears to benefit from including generated samples during training, especially in **OOD** scenarios and when combined with **CT**. Except for the in-distribution **DSC**, this setting achieves the highest performance. Individually, **CT** provides more benefit than the inclusion of generated samples. Conversely, UM-Net benefits primarily from **CT** alone and generally shows a performance decrease when generated samples are included, either alone or combined with **CT**.

Table 5.2: Results of U-Net and UM-Net Performance with Color Transfer (CT) and CycleGAN Generated Samples (GS). Best performance is depicted in **bold**, and second-best performance is underlined.

Model	CT	GS	In-Distribution (ID)			Out-of-Distribution (OOD)		
			Dice	Recall	Accuracy	Dice	Recall	Accuracy
U-Net	×	×	0.6595	<u>0.6168</u>	0.9614	0.4707	<u>0.5030</u>	0.9557
	✓	×	0.6873	0.5997	<u>0.9624</u>	0.4942	0.4703	<u>0.9643</u>
	×	✓	0.6434	0.5926	0.9609	<u>0.4987</u>	0.4788	0.9634
	✓	✓	<u>0.6725</u>	0.6430	0.9629	0.5166	0.5240	0.9647
UM-Net	×	×	<u>0.7540</u>	0.7432	<u>0.9690</u>	0.6391	0.6985	0.9725
	✓	×	0.7639	0.7548	0.9685	<u>0.6203</u>	0.7158	0.9715
	×	✓	0.7428	<u>0.7533</u>	0.9689	0.6154	<u>0.7140</u>	0.9700
	✓	✓	0.7441	0.7173	0.9694	0.6192	0.6669	<u>0.9718</u>

5.5 Conclusions

In this chapter, we explored color invariance techniques as ways to promote the model to shift its focus from the color cues present in the inputs, thus becoming more resilient to chromoendoscopy bias. To this end, we looked into CycleGAN models for paired dataset generation, having each original sample represented in both technique domains, color distribution swapping between samples, and randomly erasing the color information from a portion of the input.

We first evaluate qualitatively the results of the three techniques on the data and then their application as data augmentation techniques. Here, the CycleGAN showed promising results for inter-domain translation, but challenges remained in preserving structural details, namely in the annotations and spatial application of the blue dye in the chromoendoscopy generated images. Similarly, **CT** produces realistic results but is not robust against the presence of colored non-anatomical structures in the images.

We then provide the results for two ablation studies using these techniques as data augmentation strategies. In the first study, we assess the individual and combined effect of **CT** and **RCE**. Then, we evaluate **CT** with the inclusion of the CycleGAN-generated samples. The results for these two experiments show that for U-Net the combination of strategies enhances generalizability, whereas for UM-Net its original configuration (using **CT** only) is the most beneficial setting.

Overall, while the explored methods show potential in reducing color bias and improving generalization, they are insufficient for addressing the full extent of data variability challenges. So far, the methodologies have been heavily focused on the chromoendoscopy bias. However, as we move forward, it has become clear that we must consider all other biases, as they will continue to restrict generalizability, even if we thoroughly address the chromoendoscopy bias. This chapter

then concludes with the idea that, in order to achieve robust generalization, it is no longer enough to focus solely on this single prevalent bias; broader solutions are needed to tackle the full spectrum of data-related challenges. This is the direction that led us to the next chapter, where we explore deep self-supervised ViT features as robust data representations.

Chapter 6

Advancing Segmentation Performance with Self-Supervised ViT Representations

The primary goal of this dissertation is to enhance polyp segmentation generalizability by increasing robustness to data distribution biases. While our focus so far has been on addressing a specific bias in colonoscopy, i.e., chromoendoscopy, the conclusions presented in Chapter 5 have led us to shift towards a broader approach. We now aim to tackle additional variability sources that hinder effective generalizability. With this goal in mind, this chapter builds on the findings of Amir et al. [12], which demonstrate the potential of self-supervised features extracted from DINO-pretrained vision transformers. Before delving into their findings, it is essential to understand how the ViT architecture computes features through its self-attention mechanism and how the DINO framework leads to the extraction of meaningful and enhanced self-supervised features. The self-attention mechanism and the DINO framework are comprehensively explained in Chapters 2 and 3, respectively.

In this context, Amir et al. [12] conducted a comprehensive study on the features extracted by a DINO-pretrained ViT backbone on the ImageNet dataset, taking advantage of the interpretability of the ViT's self-attention mechanism and the robustness of the representations computed with the DINO framework. They explore the properties of these features as well as their direct application for zero-shot vision tasks, such as semantic correspondences and the segmentation of mutual semantic parts across images. They thoroughly examined the various ViT facets and their representational differences, showing that the "key" facets provide a clearer, more object-centered representation compared to other features. Furthermore, they found that keys and queries preserve positional information in the intermediate layers more effectively than values and tokens. By also comparing the properties of a supervised ViT's feature space and the DINO ViTs, they found that the self-supervised feature space organizes descriptors based on the semantic parts they represent, independently of the original animal classes of the samples, whereas the supervised feature space is organized by class. This shows that DINO ViT descriptors are more robust to

general object category, capturing transversal "object parts" similarly in the descriptor space. In their work, they obtained **SOTA** performances in part co-segmentation, semantic correspondences, and co-segmentation tasks, outperforming other unsupervised works and even some supervised methodologies [12].

Building upon these findings, this chapter argues that self-supervised **ViT** representations, specifically within the **DINO** framework, are ideal for capturing transversal and meaningful features in colonoscopy images, which can improve segmentation performance and robustness in medical images from unseen domains.

The chapter is structured as follows: In Section 6.1.1, we analyze the semantic strength of the **DINO ViT** "key" features and their potential for downstream use in polyp segmentation. In Section 6.2, we explore a part co-segmentation algorithm that offers a lightweight, interpretable, and self-supervised approach for polyp segmentation, validating it against fully supervised U-Net. Finally, in Section 6.3, we present a supervised framework for polyp segmentation and provide an extensive evaluation of its generalization capacity against baseline models. The proposed architecture achieves improved baseline results with a simple and efficient approach that relies on the quality of the extracted features rather than the complexity of numerous specialized modules developed specifically for the polyp segmentation task.

The analyses in this chapter focus on the PolypGen dataset [1], introduced in Chapter 4 as being a challenging multicenter dataset that comprehends multiple forms of data bias. The experiments presented were conducted using mainly PyTorch Lightning and Hydra. Due to the large number of experiments performed, only the main quantitative and qualitative results are presented here. For more detailed information, please refer to Appendix A.

6.1 Analysis of Key Features from DINO-ViT

This section introduces a preliminary analysis of **ViT**'s "key" features and their semantic relevance when extracted from colonoscopy images. We begin by applying a semantic correspondence algorithm [12]. Next, we visualize the progression of the "key" descriptors across **ViT** layers using two self-supervised backbones pretrained in distinct domains. We then analyze the organization and interpretability of the descriptor space by performing image retrieval based on general structural similarity and based on polyp-specific similarity.

The observations from this preliminary analysis lay the foundation for further exploration of these features, motivating their deeper investigation for the downstream task in the following sections.

6.1.1 Semantic Correspondences

As a preliminary exploration of the semantic potential of key facets, our aim is to find semantic correspondences between descriptors of different images. This descriptor matching algorithm [12] proceeds as follows:

1. *Descriptor extraction*: First, "key" features are extracted from two images.
2. *Saliency map extraction and foreground voting*: Saliency maps are then calculated to classify the descriptors as foreground or background. Saliency maps emphasize the most relevant regions of the input image on the basis of the models' focus while making the prediction. Pixels with higher values on the saliency maps have a greater contribution to the final output. Therefore, descriptors with activation above a defined threshold t are classified as foreground, while less activated descriptors are assigned to the background and filtered out.
3. *Matches search*: Next, we calculate the cosine similarity between the foreground descriptors of both images, identifying matches.
4. *Match ranking*: Finally, the matches are grouped with the K-Means algorithm and subsequently ranked for top match selection.

Figure 6.1 shows examples of the descriptor matching results with the ImageNet ViT s8 pre-trained extractor.

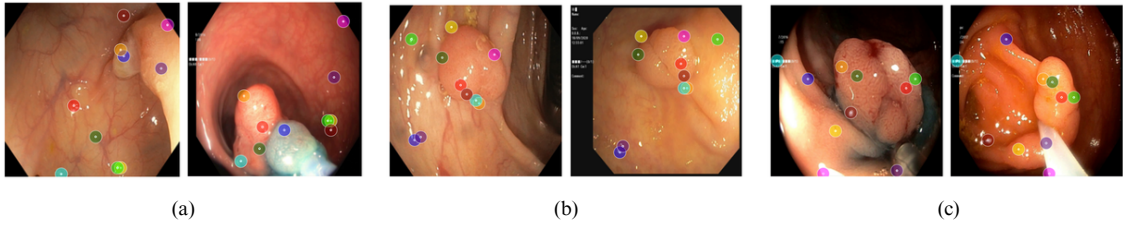


Figure 6.1: Visual examples of semantic correspondences between two PolyGen instances. For each pair, each colored point on the left image corresponds to semantic match to the same colored point on the right image.

The correspondences between images in the same pair ((a), (b) and (c)) prove to be quite interpretable and robust to changes such as polyp size, style and technique. We can point out specific correspondence examples such as the orange points in pair (a), marking the polyp boundary both in the left and right images. For the same pair, in red we can also see marked light reflections on the colon's surface. Similarly, on example (b), we have yellow and dark green on the polyp boundary, in similar positions. In (c), dark blue is pointing to colon surface folds in both images, as well as purple to an inserted medical instrument.

The interpretability of these correspondences shows that the descriptors capture meaningful structural information, including polyp-related details, even without labeled data. This suggests that, despite the domain differences between ImageNet's and PolyGen's images, the learned features can be effectively applied to colonoscopy, enhancing the robustness of downstream applications.

6.1.2 Evolution of Key Descriptors Across Layers

In this analysis, we extract key descriptors from PolypGen images using three DINO-pretrained extractors, all based on the ViT small (ViTs) architecture with 11 layers. The main differences between the extractors lie in the pretraining dataset and the input patch size. We use two ImageNet-based extractors: one with 8x8 pixel patches (ViTs8) and the other with 16x16 pixel (ViTs16). Additionally, we use a ViTs16 GastroNet-pretrained extractor. The ViTs8, with its smaller patches, provides more detailed descriptors but is more computationally intensive. In contrast, ViTs16 with larger patches is more efficient in terms of computation but may capture less fine-grained information.

We chose to contrast the ImageNet and GastroNet backbones because they offer complementary strengths. While ImageNet provides a broad, general representation of visual patterns learned from natural images, GastroNet is specifically pretrained on colonoscopy images, making it more suitable for colonoscopy-related applications. This contrast allows us to explore the impact of domain-specific versus general pretrained features on descriptor quality in the context of colonoscopy imaging. Furthermore, this analysis provides a comparative view of how descriptors evolve through progressively deeper layers of the ViT architecture.

In Figure 6.2 (a), we observe that the ImageNet-pretrained ViTs8 backbone captures generic patterns in the initial layers, while the deeper layers focus on more complex, domain-specific structures. Despite being pretrained on natural images, the model effectively distinguishes polyp areas and subtle colon features, such as blood vessels, particularly in the final layers. Additionally, the Principal Component Analysis (PCA) of these descriptors shows that the same structures across examples are represented by consistent colors, indicating their similar encoding. This consistency demonstrates how these descriptors reliably capture the same features across different images, which is crucial for enhancing generalization across domains. The ability to maintain consistent encoding across different images highlights the robustness of the ImageNet-pretrained ViTs8 and supports its use in downstream tasks, where generalization to unseen medical data is essential.

In (b), we see a similar progression but with loss of structural detail, which is expected due to the larger patch size used in ViTs16.

In (c), the GastroNet-pretrained ViTs16 backbone highlights domain-specific features in the shallow layers. While the GastroNet model excels in capturing domain-specific features, the detailed and more consistent encoding of ImageNet's backbone provides superior generalization potential for broader medical imaging applications.

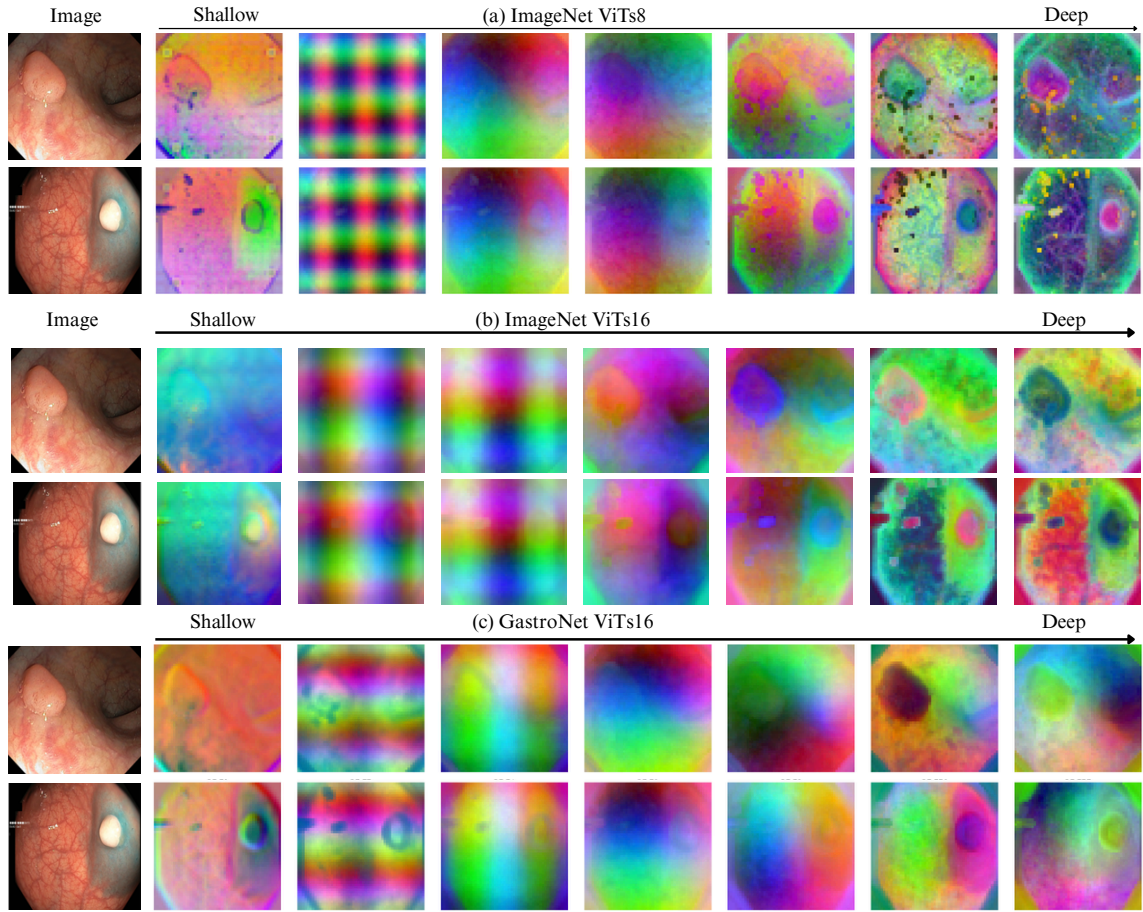


Figure 6.2: Visual analysis of the change in descriptor PCA representation through progressively deeper layers of the ViTs8 and ViTs16 pretrained backbones. Here we represent these extracted features from two example images.

6.1.3 Similar Image Retrieval

So far, matching extracted descriptors has proven effective in providing meaningful semantic correspondences. We have also examined how these descriptors evolve through the layers of the extractors, observing how they consistently highlight important data structures across examples, even when pretrained on non-medical data. In this section, we aim to further investigate the organization of this descriptor space. Specifically, we evaluate whether the consistent and meaningful encoding observed can be leveraged for similar case retrieval, thereby enhancing the interpretability and trustworthiness of these descriptors from a clinical perspective.

We began by extracting descriptors for 60% of the PolypGen images using the ImageNet ViT s8 extractor to construct a representative reference pool for image retrieval. Next, we trained a *k*-Nearest Neighbor (*k*-NN) model on this pool. For a target image, we extract its key descriptors and project them onto the pool, retrieving the nearest neighbor from the reference set. We then wish to evaluate the structural similarity of the pairs obtained, both in general and with a focus on the clinically relevant structures.

6.1.3.1 Based on general similarity

Figure 6.3 shows three examples of target/reference pairs retrieved through the search.

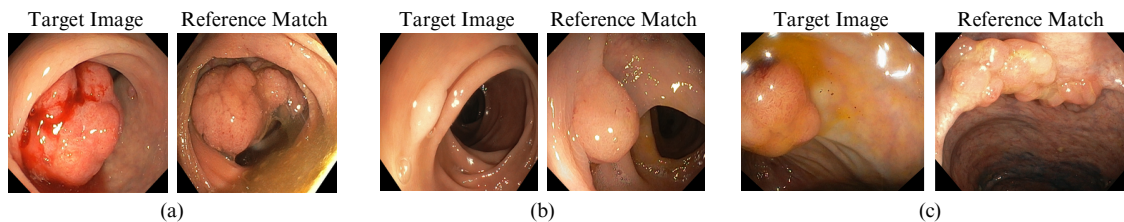


Figure 6.3: Examples of nearest neighbor search results in the reference pool. (a), (b) and (c) mark each target/reference pair found.

The structural similarity between the obtained pairs is evident, with consistent alignments of various structures across the images. This overall similarity reinforces that the descriptors encoded by the model are meaningful, as they capture relevant structural patterns. However, while the **k-NN** model identifies similar images, it does not account for the clinical significance of its structures. For instance, a match may show visually similar patterns, but the clinically relevant factors, such as polyp size and shape, may differ. Therefore, while the retrieval process is effective for identifying similar images, further evaluation is needed to ensure that the retrieved cases are truly relevant in a clinical point of view. Despite this, the observed correlation between shorter distances in the descriptor space and higher observed structural similarity further reinforces that the feature space is semantically meaningful.

6.1.3.2 Based on polyp similarity

Continuing the analysis of the similarities of descriptors between image pairs, we then performed this search with cosine similarity instead of the **k-NN** search. Here, the reference pool from above is filtered to contain only the polyp descriptors of the references. We then compute the cosine similarity between all descriptors of the target image and the descriptors in the pool, selecting the reference with higher similarity to the target. This approach allows us to assess how well the model identifies similar polyps across images, further validating the relevance and consistency of the learned features, while evaluating if such a method could be useful for automatic retrieval of similar clinical cases.

Examples of pairs found are shown in Figure 6.4 below.

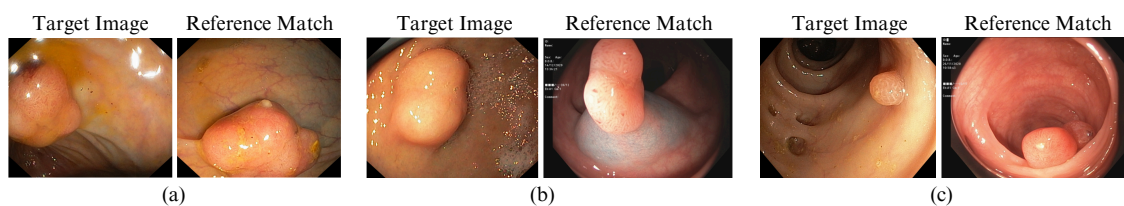


Figure 6.4: Three examples of polyp descriptors-based similarity search. (a), (b) and (c) denote different target/reference pairs found.

Comparing Figures 6.3 and 6.4, we can observe that, in the latter, results are highly driven by similarity in polyp shape and size as intended, regardless of the origin center or structural positions in the image. The target image of pair (c) Figure 6.3 is the same as in pair (a) of Figure 6.4, showing the match differences for the two cases.

6.1.3.3 Limitations

While the similarity searches showed promising results, **k-NN** primarily captures general whole-image similarity rather than ensuring specific part-level similarity. This would have direct implications for its use in similar case retrieval systems: they may retrieve images that are structurally similar overall but not necessarily similar in the clinically relevant regions. Furthermore, we verified that the obtained references are consistently from the same center as the target when there is representation of the target's center in the pool, meaning that these descriptors also capture center-specific characteristics in the data. When matching images based on polyp-specific descriptors, the method works well when polyps occupy a significant portion of the image, ensuring accurate retrieval of similar cases. However, for target images with small or subtle polyps, the background dominates, leading to similarity results based on general tissue characteristics rather than polyp-specific features. This analysis highlights the potential of the the usage of this search for similar-case retrieval; however, it requires improvements to ensure reliability in small-polyp matches.

6.2 Part co-segmentation

This section explores the adapted application of the part co-segmentation pipeline proposed by Amir et al. [12] for polyp segmentation. This zero-shot method segments common parts across images by clustering descriptors, relying on the consistent encoding of related object parts across different images. We aim to evaluate whether this approach can be effectively applied to the colonoscopy domain, offering a simple, lightweight, and interpretable method for segmenting colonoscopy structures, with a focus on polyps. By analyzing different extractor settings and different data grouping strategies, we aim to determine the optimal conditions for applying this method effectively to the end task. After choosing the best settings from these two ablation studies, we move on to a 5-fold cross-validation comparison between our adapted part co-segmentation framework and a fully supervised baseline.

The part co-segmentation pipeline is shown in Figure 6.5 below.

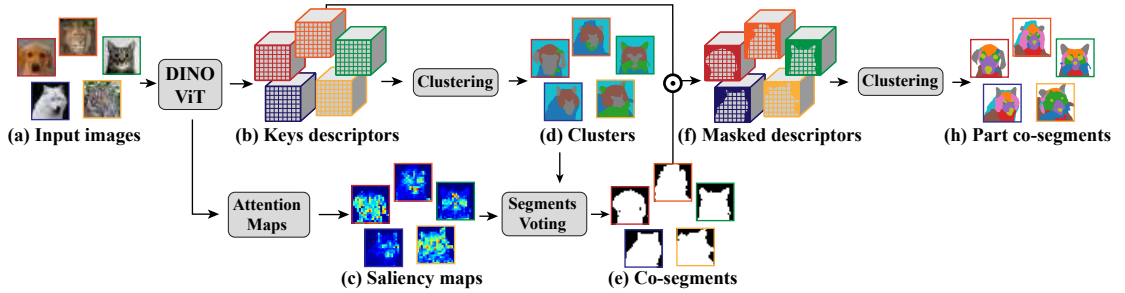


Figure 6.5: Part co-segmentation pipeline. The group of input images (a) goes through the feature extractor, retrieving the key descriptors (b) and saliency maps (c) from a specified layer of the ViT architecture. All descriptors are then clustered using K-Means (d). Each cluster is then assigned the class "background" or "foreground" based on saliency, forming co-segments (e). A new clustering is applied exclusively to the foreground descriptors (f), resulting in foreground part-co-segments [12].

It goes as follows:

1. *Descriptor extraction:* The input images are fed to a **DINO**-pretrained **ViT** to obtain the "key" descriptors.
2. *First clustering:* These descriptors are then clustered using Faiss' K-Means algorithm, trained to determine the optimal number of descriptor clusters for the set of images provided with the elbow method [85].
3. *Foreground descriptors extraction:* Saliency maps extracted from the **ViT** model are used to classify these clusters as either foreground or background. Each image in the input group casts a vote for foreground clusters based on the activation levels in its saliency map. If the activation exceeds a specified threshold t , the vote is cast for the foreground; otherwise, it is considered background. This voting mechanism ensures that only clusters consistently

activated across the majority of images are labeled as foreground, enhancing robustness against noise and dataset variability.

4. *Foreground descriptor clustering*: After this, we focus solely on the foreground descriptors and repeat the clustering process, resulting in segments for the foreground regions.
5. *Third clustering (optional)*: There can be a third clustering process to further refine the segmented regions. This is referred to as the "third stages" case.

6.2.1 Ablation Study on Group Specificity

In order to assess how group specificity affects performance, part co-segmentation was tested on progressively more specific image groups. The grouping strategies were chosen to test the hypothesis that more specific data grouping could reduce descriptor variability and improve segmentation accuracy, particularly in the challenging context of colonoscopy images.

We first apply this method to the global testing group as a control experiment, then progressively divide the data into more specific groups. For instance, by using K-Means clustering, we create groups of images with similar descriptors. Alternatively, we can group the data into classes according to human-perceived similarities, such as polyp size and number. Finally, we can form pairs of very similar images, utilizing the previously explored similar image retrieval method, achieving maximum specificity.

Cluster division. The number of clusters was determined using both the elbow method [85] and the silhouette method [86] in agreement. The clusters obtained using the training data are shown in Figure 6.6, alongside the center of each sample. From this comparison, we observe that the center of the origin influences the arrangement of images in the feature space, suggesting that while the descriptors capture robust features, they also retain some center-specific information.

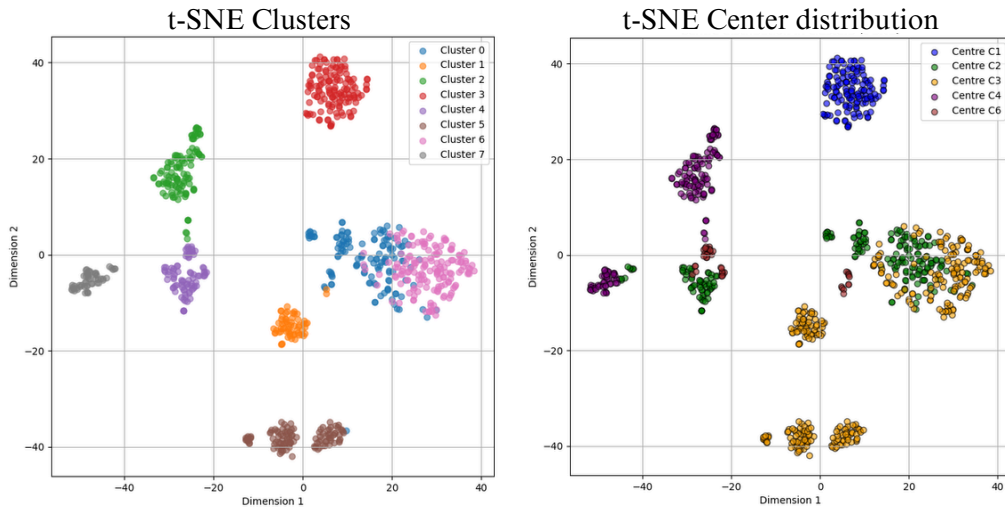


Figure 6.6: Train set clusters t-SNE representation (left) and origin center of each sample (right).

Samples reserved for ablation evaluation are then assigned to the clusters obtained from the training phase, as shown in Figure 6.7.

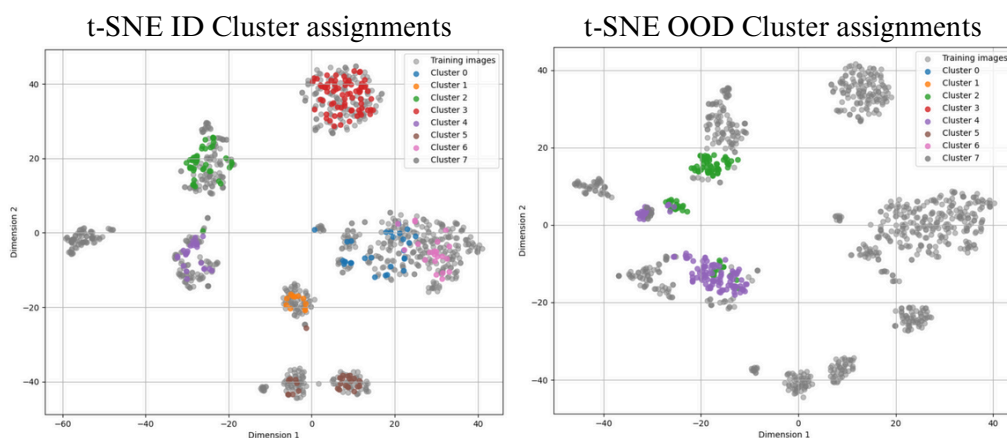


Figure 6.7: t-SNE representation of the inference clusters for the ablation study.

Class division. The classes were defined based on visual characteristics of the polyps: no polyp, multiple polyps, single small polyp, single medium polyp, and single large polyp. To define these classes, the ground truth masks were resized to 224x224 pixels, and their connected components were analyzed. If a ground truth mask was empty it was assigned to class 0. If it contained more than one connected component (excluding standalone pixels), it was assigned to class 1. If the mask had a single connected component with an area smaller than 690 pixels, it was assigned to class 2. If the polyp area was between 690 and 2167 pixels, it was assigned to class 3. Finally, if the area exceeded 2167 pixels, it was assigned to class 4. These thresholds were empirically defined.

Figure 6.8 illustrates examples of the defined classes, with red dots indicating the centroids of individual components. Some red dots appear isolated, not connected to any part, as they correspond to randomly selected pixels in the annotations.

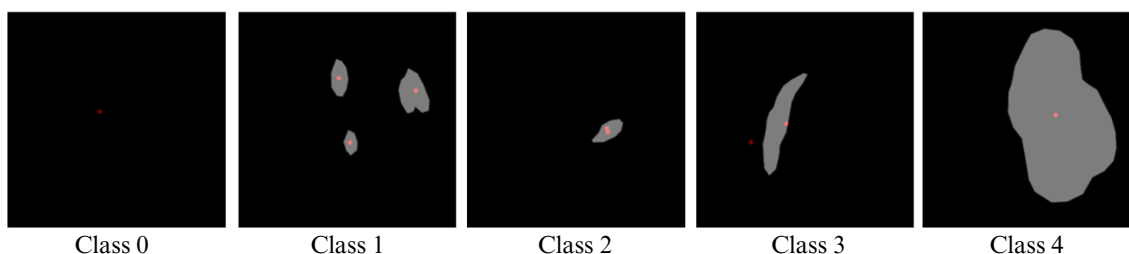


Figure 6.8: Examples of samples assigned to each considered class.

Sister images division. In the "sister images" division, we part co-segment images with high structural similarity, retrieved through the k -NN search method discussed in the previous section.

This was done for a single sister configuration, where only one nearest neighbor is retrieved, as well as for a configuration that retrieves a variable number of references, shown in Figure 6.9.

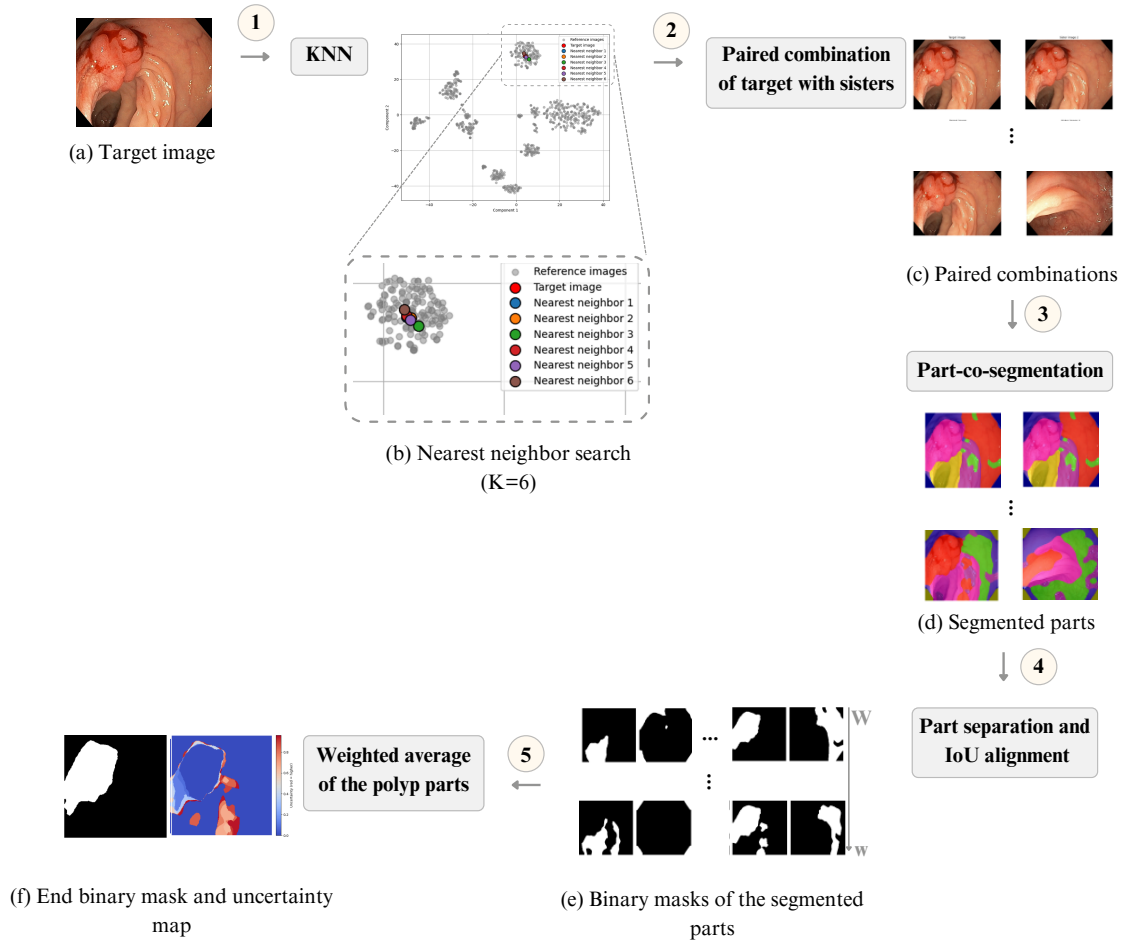


Figure 6.9: Multiple Sisters Pipeline. The target image (a) is assigned k nearest neighbors via NN search (b). $k+1$ paired combinations are then formed (c), and part co-segmentation is performed on each pair (d). The resulting parts are separated and aligned (e) and a weighted binary mask along with an uncertainty map are generated (f).

This approach has the advantage of providing multiple part co-segmentations for the same target, allowing us to highlight consistently segmented parts across various examples. This increases confidence in the segmentations. Additionally, by incorporating multiple "sister" opinions, it generates more stable results and produces an uncertainty map representing the voting distribution across the combinations.

The "multiple sisters" part co-segmentation approach goes as follows:

1. *Nearest-neighbor search and pairs definition:* The target descriptors are extracted, and K sister images are retrieved from the reference pool.

2. *Forming paired combinations with target:* Then, all possible pairs between the target and reference images are formed. Here the target image is treated as "sister image 0", with the highest possible similarity. With this, $K+1$ part co-segmentations are performed.
3. *Part co-segmentation of each combination:* Then, each paired combination is part-co-segmented independently.
4. *Part separation and alignment:* The target image's segmented parts are separated into individual binary masks. The indices between pairs are then aligned based on Intersection over Union (IoU), meaning that binary masks with the same index correspond to segmentations of the same semantic part.
5. *Polyp part combination between pairs and uncertainty map estimation:* The aligned polyp segmentation masks are combined as a weighted average, where closer neighbors to the target image contribute more heavily to the final mask, and vice versa. The target image itself is considered its top nearest neighbor, thus giving the end mask the highest weight in this voting. Neighbor number 1 is next, and so on. This process allows the generation of both the final segmentation mask and an associated uncertainty map.

The combined segmentation mask (M) is determined based on a weighted average of the binary polyp masks from the sister-target combinations (Equation 6.1).

$$M = \sum_{i=0}^{K-1} w_i \cdot \text{mask}_i \quad (6.1)$$

where w_i is the weight given to the binary mask produced by the combination of the target with sister i .

These weights are calculated based on Equation 6.2:

$$w_i = \lambda^i, \quad \text{for } i = 0, 1, 2, \dots, K-1 \quad (6.2)$$

where λ is the chosen decay factor. In this case, we used $\lambda = 0.7$.

This combined mask M is binarized to provide the final output according to Equation 6.3:

$$\text{Bin}_M = \begin{cases} 1, & \text{if } M \geq \frac{\sum_{i=0}^{K-1} w_i}{2} \\ 0, & \text{otherwise} \end{cases} \quad (6.3)$$

The uncertainty map can be defined by Equation 6.4:

$$U = \left| \mathbf{1}_{m \times n} - \frac{M}{\sum_{i=0}^{K-1} w_i} \right| \quad (6.4)$$

where $\mathbf{1}_{m \times n}$ represents a matrix of ones with the same dimensions as M . In U , lower values represent lower uncertainty and vice-versa.

6.2.2 Ablation Study on Extractor Settings

Besides different data divisions, we also analyze the performance across different extractors. For simplicity, they will be referred to with codenames being divided into:

- **ImageNet ViTs8 variants:**

- **ViTs8:** ViTs8 architecture pretrained on ImageNet;
- **ViTs8 + SSL100:** ViTs8 architecture pretrained on ImageNet and finetuned with DINO training on PolypGen data for a duration of 100 epochs;
- **ViTs8 + SSL200:** ViTs8 architecture pretrained on ImageNet and finetuned with DINO training on PolypGen data for a duration of 200 epochs;
- **ViTs8 + SSL100 + SL300:** The ViTs8 + SSL100 was used as a starting point for supervised training with a CNN decoder for polyp segmentation for 300 epochs;
- **ViTs8 + SSL200 + SL300:** The ViTs8 + SSL200 was used as a starting point for supervised training with a CNN decoder for polyp segmentation for 300 epochs;

- **GastroNet ViTs16 variants:**

- **ViTs16:** ViTs16 architecture pretrained on GastroNet;
- **ViTs16 + SL300:** ViTs16 architecture pretrained on GastroNet was used as starting point for supervised polyp segmentation training with a CNN decoder.

Implementation Details We used a fixed segment of PolypGen train/test images for consistency, splitting the data by patient to avoid any data leakage. Base extractor fine-tuning was performed on NVIDIA RTX8000 using the training images with a maximum of 200 epochs, a batch size of 4, learning rate of 10^{-5} and the Adam optimizer [82]. We used global crops scaled between 0.4 and 1.0 of the image size, and 8 local crops scaled between 0.05 and 0.4. The source code for DINO was retrieved from DINO repository. We then performed all part co-segmentation experiments on NVIDIA RTX2080Ti consistently resizing the input images to 224x224.

6.2.3 Quantitative Results

We perform an ablation across data divisions (global, clusters, classes, sisters, multiple sisters), extractor configurations (ViTs8 and ViTs16 variants), and clustering stages (2 or 3 stages). In order to see exact performance values in table format and results per division, please refer to Appendix A.

In the presented graphs (Figures 6.10 to 6.13), we depict supervised extractors in blue dot-dashed lines and self-supervised extractors in pink dotted lines. We incorporate supervised extractors as a baseline for comparison, aiming to primarily evaluate the performance of self-supervised extractors.

Looking at Figure 6.10, we observe that higher specificity in data grouping and increased PolypGen fine-tuning improve ID performance, with the multiple sisters approach yielding the best results. This pattern does not hold for the supervised variants, which is likely due to the consistent threshold used across experiments in the foreground saliency voting step. Since saliency for supervised extractors is heavily focused on polyps, this process results in the segmentation of the polyp into multiple parts, thus lowering performance. By comparing the blue and pink lines, we can see that applying the multiple sisters strategy allows the self-supervised extractors to achieve performance closer to that of the supervised models.

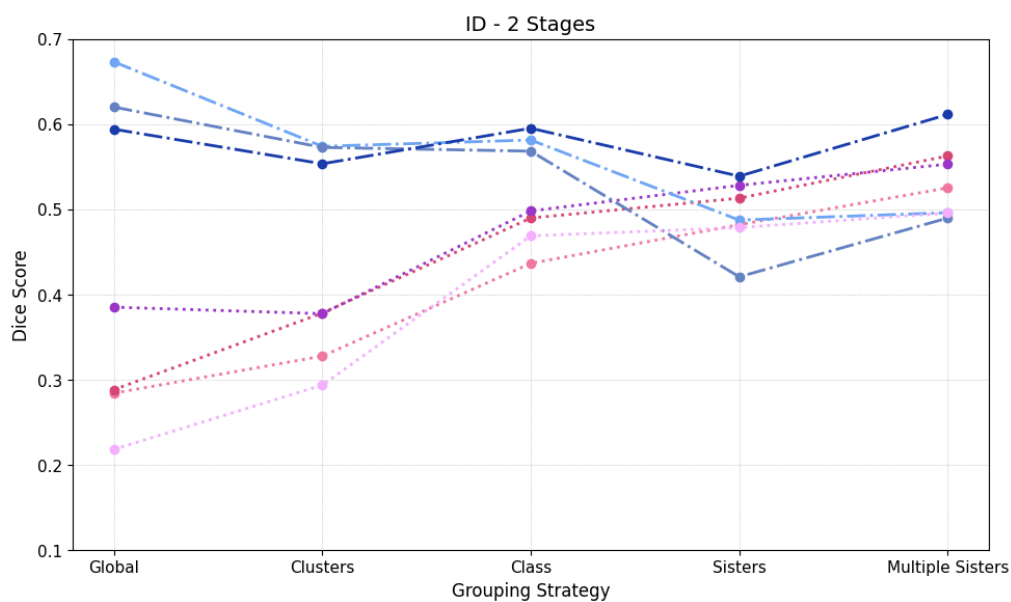


Figure 6.10: Quantitative In-Distribution (ID) results using 2 stages of clustering. The Y axis represents the dice score (Equation 4.3), and the X axis represents the group divisions, ranging from less specific (global) to most specific (sister variants). Each colored line represents an extractor configuration. Extractors with self-supervised pretraining are colored in pink dotted lines, whereas those with supervised training are depicted in blue dash-dot lines for better interpretation.

In Figure 6.11, the same analysis is presented with an additional clustering stage. Here, we observe that using just two stages yields better and more stable results, making this third stage unnecessary.

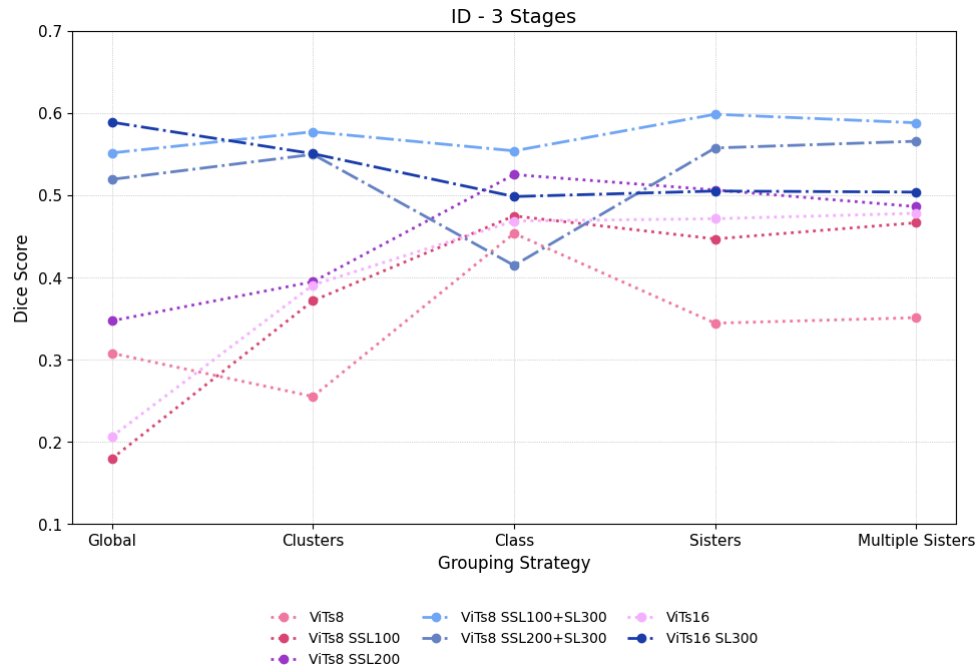


Figure 6.11: Quantitative In-Distribution (ID) results using 3 stages of clustering.

Looking at the OOD performance in Figures 6.12 and 6.13, we observe a similar improvement pattern with higher fine-tuning and group specificity, although the enhancement is less pronounced compared to the ID results.

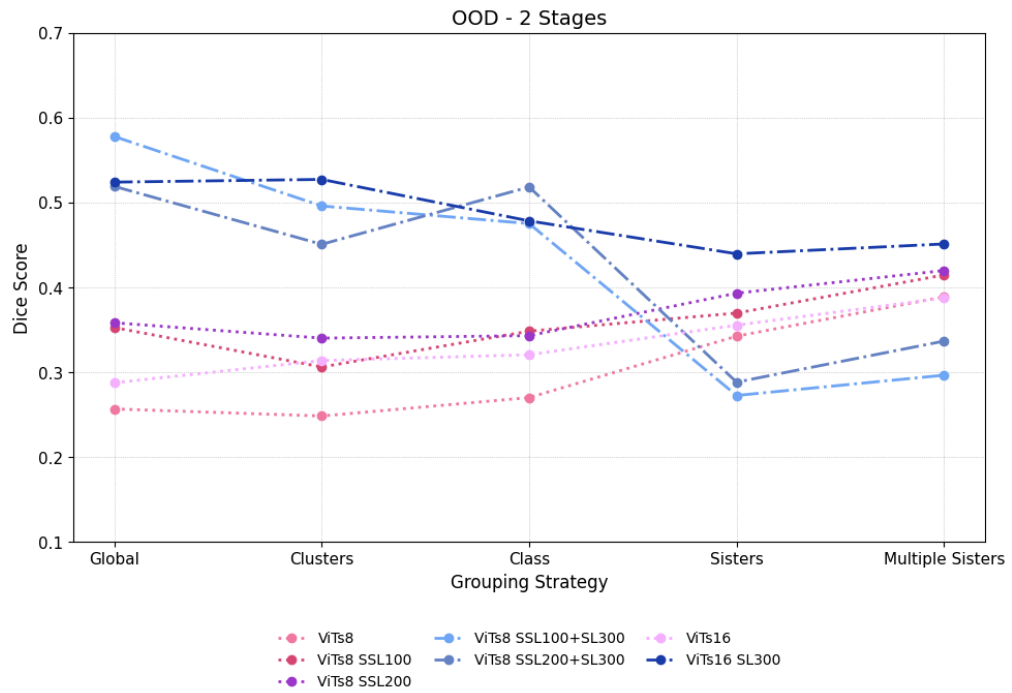


Figure 6.12: Quantitative Out-Of-Distribution (OOD) results using 2 stages of clustering.

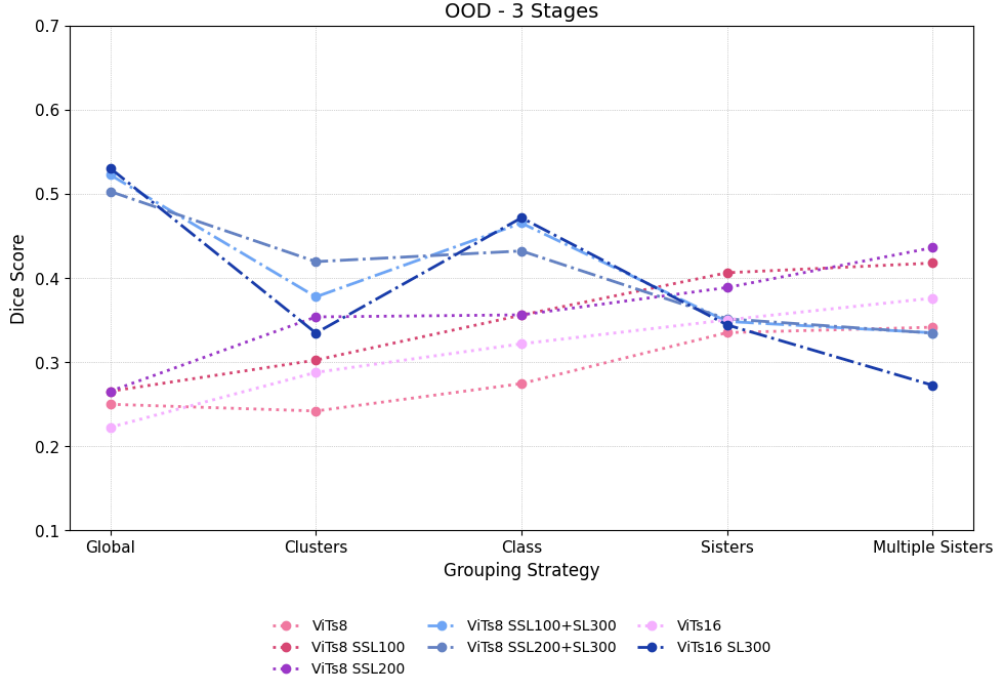


Figure 6.13: Quantitative Out-Of-Distribution (OOD) results using 3 stages of clustering.

We specifically assess the relative improvement in percentage provided by the multiple sisters approach (with $K=6$) when compared to the sister pairs (Table 6.1). The inclusion of multiple sisters shows a notable and significant improvement, especially with two-stage clustering. Although we kept the three-stage result for comparison, this extra clustering step introduces high instability, making it unnecessary in the part co-segmentation process and thus excluded from further analysis.

Table 6.1: Relative improvement (in %) added from the multiple sisters approach compared to the initial sisters grouping.

Extractor	Pretrained on	Finetuned on	Two Stages		Three Stages	
			Dice (ID)	Dice (OOD)	Dice (ID)	Dice (OOD)
DINO ViTs8	ImageNet	-	8.961	13.547	1.946	1.910
DINO ViTs8	ImageNet	PolypGen SSL (100 epochs)	9.665	12.192	4.410	2.831
DINO ViTs8	ImageNet	PolypGen SSL (200 epochs)	4.713	6.790	-4.009	12.198
DINO ViTs8	ImageNet	PolypGen SSL (100) + SL (300)	1.785	8.767	-1.738	-3.876
DINO ViTs8	ImageNet	PolypGen SSL (200) + SL (300)	16.393	19.270	1.471	4.838
DINO ViTs16	GastroNet	-	3.572	9.001	1.400	7.398
DINO ViTs16	GastroNet	PolypGen SL (300 epochs)	13.451	2.639	-0.257	-20.860

6.2.3.1 Qualitative Results

Here, we present qualitative results for the two main ablation lines: data division and extractor variants. Figure 6.14 shows a successful example, while Figure 6.15 illustrates a failure example.

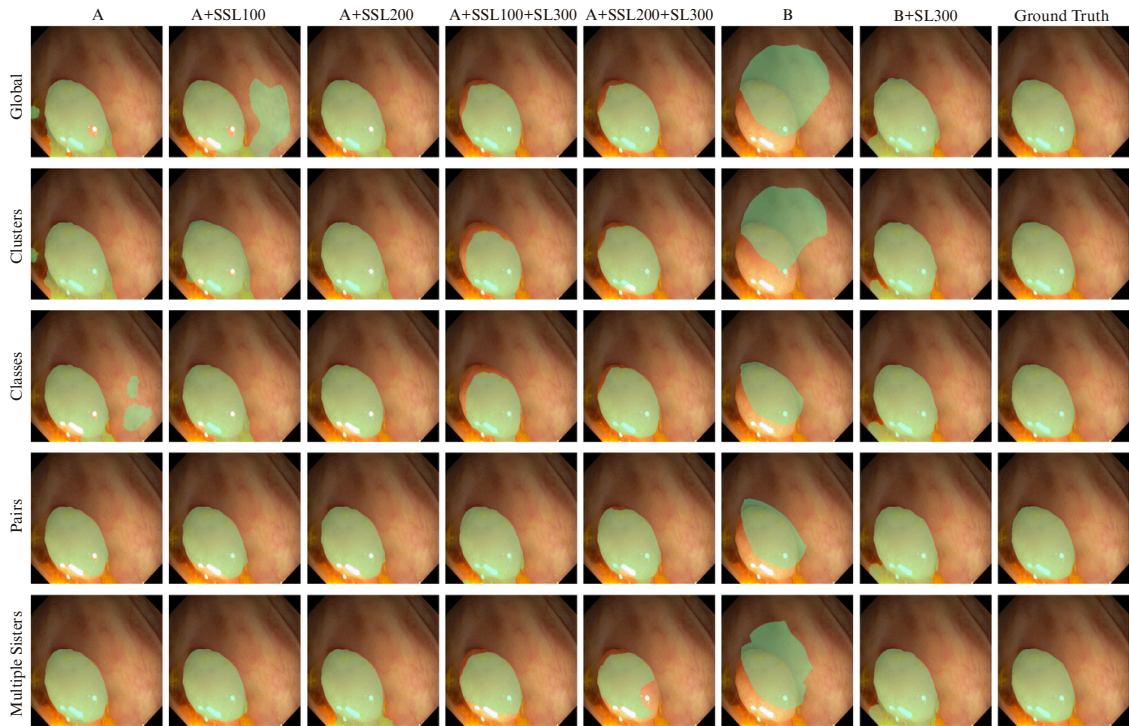


Figure 6.14: Qualitative results across extractors and data divisions. Columns represent extractors and rows division types. A = ImageNet pretrained ViT .

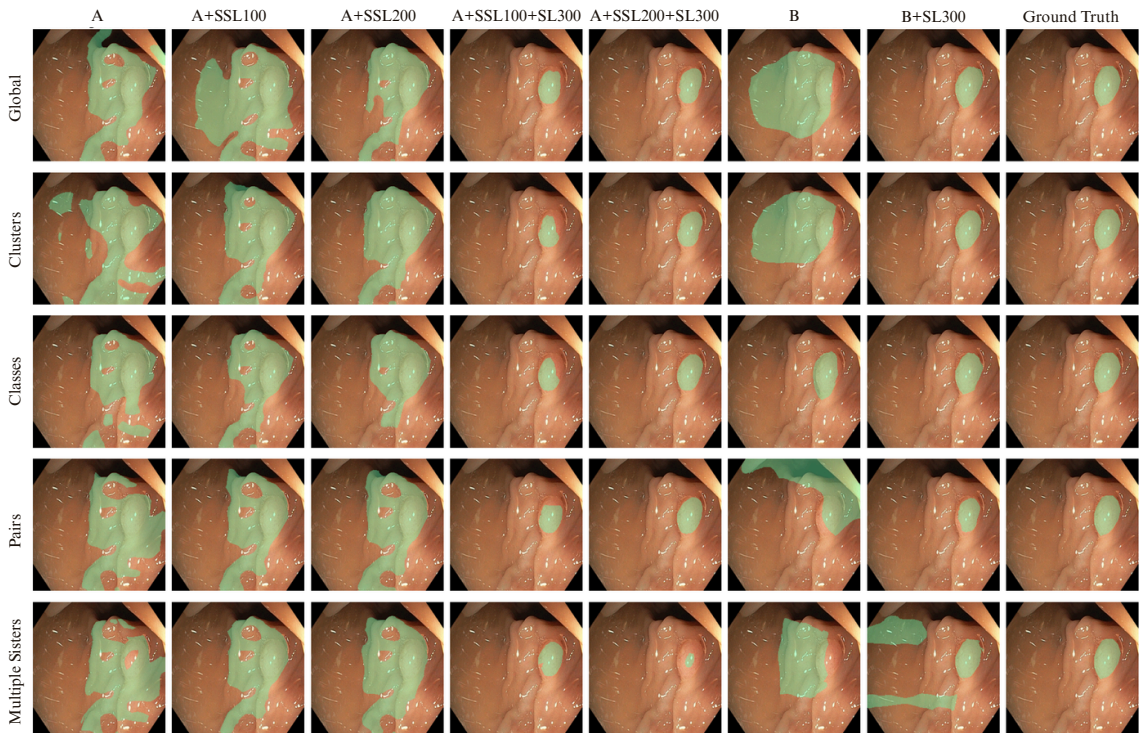


Figure 6.15: Qualitative failure/underperformance examples. Columns represent extractors and rows division types. A = ImageNet-pretrained ViT , B = GastroNet-pretrained ViT .

From these examples, we can extract similar relative conclusions as in the quantitative analysis. The best overall configuration is multiple sisters approach with the ViTs8 + SSL200 extractor.

We also provide a visual example of the sisters approach in Figure 6.16, including intermediate steps, namely the first clustering results and the final segmented parts before polyp part selection. Panel (b) shows the initial clustering, with corresponding colors in target and reference representing same cluster pixels. In this example, saliency voting did not result in the exclusion of background descriptors, thus having the final four segmented parts in (c) as refinements of the initial clustering. Lastly, panel (d) displays the selected polyp mask.

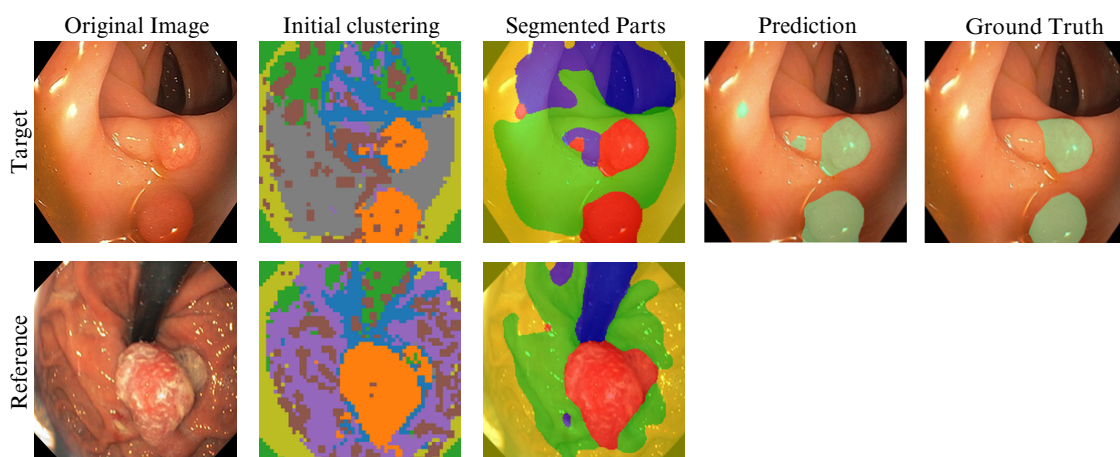


Figure 6.16: Example for a single pair. The second column shows the first clustering performed on the dense descriptors of both images, where correspondent colors mark descriptors of the same cluster. The third column depicts the final clustered parts, where, similarly, same color indicates same semantic part. Finally, the last columns show the segmented polyp mask and ground truth for the target image.

In Figure 6.17 we can also see two qualitative results of the multiple sisters approach, highlighting the addition it provides to the previous ones.

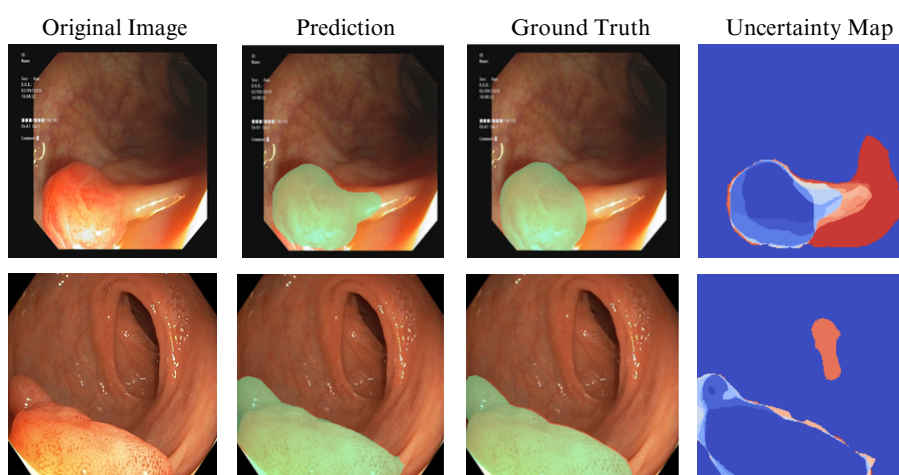


Figure 6.17: Examples of predictions and uncertainty maps for the 'multiple sister pairs' approach.

In sum, increasing **DINO** fine-tuning on the ImageNet extractor leads to improved polyp segmentation, achieving its best results when applied in a multiple-sisters setting. Furthermore, visual results of sister variants prove to be quite complete and interpretable, translating the algorithm’s decision on intermediate steps and including uncertainty. With these conclusions, we move on to the 5-fold cross-validation of this approach against a fully supervised baseline.

6.2.4 Evaluation of the Multiple Sisters Approach Against a Fully Supervised Baseline

We evaluate the multiple sisters approach against fully supervised U-Net using 5-fold cross-validation, both with and without an automatic polyp part selection method. When evaluating without automatic polyp part selection, we are measuring the ‘upper bound’ performance of this method. In this case, the polyp segmentation capacity is assessed without the added complexity of selecting the correct polyp part.

Introducing an automatic polyp part selection method inevitably leads to some performance degradation, representing the ‘real bound’ of the approach. This is because the method must identify the specific polyp part from among all segmented structures. As such, the ‘upper bound’ represents the raw segmentation capacity of the method, unimpeded by the challenges of part selection.

The polyp part selection method used in this study is illustrated in Figure 6.18.

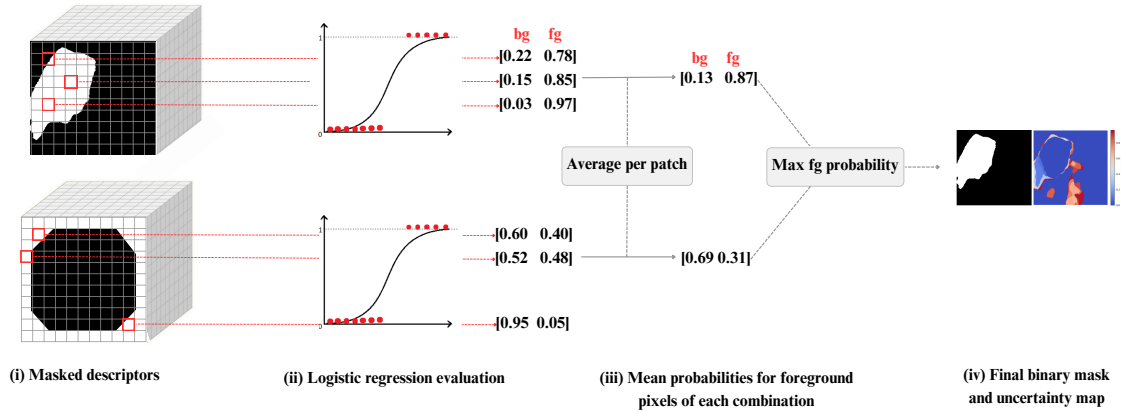


Figure 6.18: Logistic-based Polyp Part Selection. For each semantic part, the descriptors of each patch are evaluated by a logistic regression classification model that outputs the probability of it being a polyp patch. In the end, the predictions for all-part patches are combined, forming a global probability per part.

The patch descriptors from each segmented part are evaluated independently by a logistic regression model. This model is trained using patches sampled from one of the training folds. Each patch serves as an independent observation, with its descriptors as features. The logistic classifier assigns a probability to each patch, indicating the probability that it belongs to a polyp. The individual patch probabilities are then aggregated and averaged to produce a final probability

prediction for each segmented part. The part with the highest foreground probability is selected as the final result. Given that this logistic regression classifier requires labeled patches for training, the addition of this step introduces a degree of supervision to the method. However, since the classifier is trained at the patch level, it requires only a relatively small number of images for training.

Implementation details The data from centers 1-5 is split into 5 folds, ensuring that instances from the same patient are grouped together in order to prevent data leakage. In each iteration, 4/5 of the folds are used for training, 1/5 for validation. Center 6 samples were reserved exclusively for testing to provide an external validation set that ensures the model’s performance is evaluated on data that was not used during training or validation. The selection of this center for testing was arbitrary. Images from one of the training folds are used for logistic classifier training, avoiding any data leakage. A ground truth patch is considered a polyp patch (class 1) if more than 30% of its pixels are labeled 1, and vice-versa. Additionally, we sample the randomized patches from the training images in order to have an equal representation of the background (0) and polyp (1) classes. All experiments were performed on NVIDIA RTX2080Ti, ensuring hardware consistency.

6.2.4.1 Results

In Table 6.2, we compare the self-supervised multiple sisters approach with and without logistic part selection to a fully supervised U-Net. The proposed method shows raw performance close to the supervised baseline. However, when automatic polyp part selection is introduced, the upper bound drops by approximately 20%. This performance drop is expected, as part selection may occasionally fail, leading to incorrect part outputs and reduced performance. Despite this, the method’s performance remains competitive with the fully supervised approach across 5-fold cross-validation, highlighting its potential.

Table 6.2: DSC results from the 5-fold cross-validation comparing the multiple sisters approach (K=6) with the fully supervised baseline. Best performance is presented in **bold**.

Model	Average Dice _{val} $\pm$ std _{val}	Average Dice _{test} $\pm$ std _{test}
U-Net	0.6727 $\pm$ 0.0232	0.6744 $\pm$ 0.0146
Multiple Sisters upper bound	0.5491 $\pm$ 0.0262	0.5598 $\pm$ 0.0126
Multiple Sisters real bound	0.4290 $\pm$ 0.0101	0.4620 $\pm$ 0.0090

Figure 6.19 presents qualitative examples comparing the U-Net segmentation results with the multiple sisters approach. Additionally, we include intermediate results for two target-sister combinations in Figure 6.20. These visual results reinforce our quantitative findings, demonstrating that the multiple sisters approach, at its best, approximates the fully supervised baseline. It also generates uncertainty maps that highlight variability across combinations. Areas with higher variability indicate greater uncertainty, suggesting regions that require closer clinical attention and

further evaluation. Overall, we found that the best segmentations for this approach tend to show high inter-sister agreement, resulting in cleaner uncertainty maps.

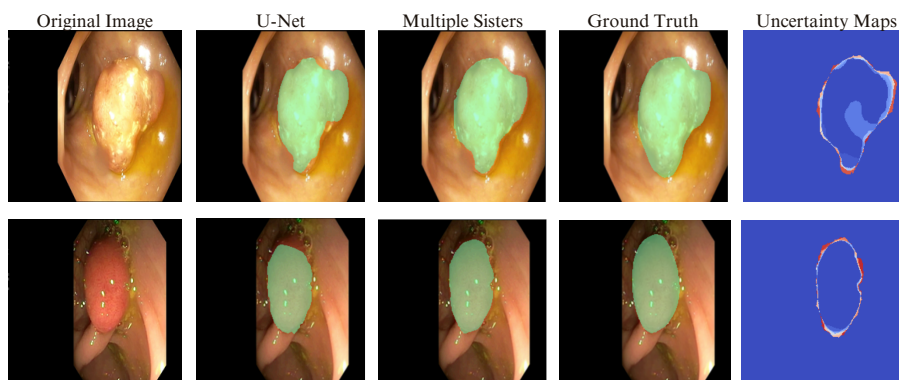


Figure 6.19: Qualitative test set examples for the multiple sisters approach vs U-Net, including corresponding uncertainty maps. Cooler colors translate lower uncertainty, whereas hotter color indicate higher uncertainty.

This approach also allows visualization of intermediate results (Figure 6.20), which help build the final prediction. By examining the retrieved similar images and the co-segmented parts aligned with the target, we see that the reference polyp matches the same semantic region as the target, boosting confidence in the segmentation. Moreover, the approach is intuitive and easy to understand, making it more accessible and demystifying the complexities of convolutional networks.

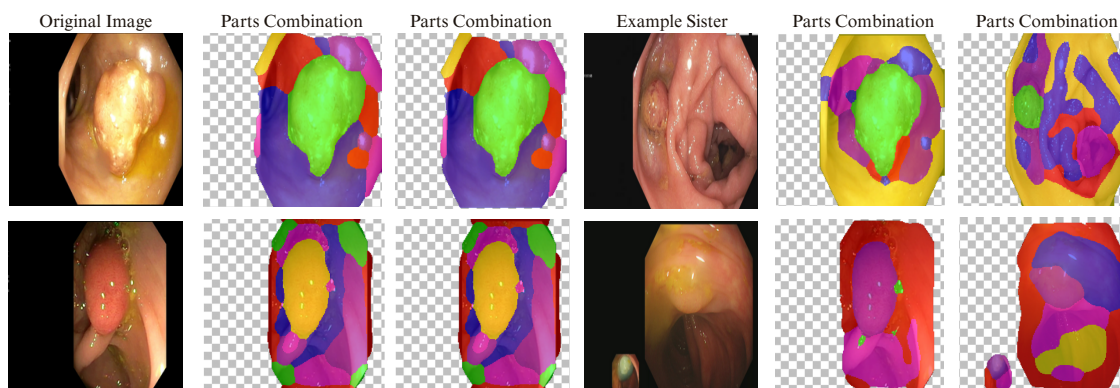


Figure 6.20: Representation of two sister combinations for each example. First combination being the target and itself, and second with one of the retrieved sisters.

6.2.5 Discussion

In this section, we performed extensive ablation studies on the part co-segmentation pipeline, namely across extractor pretraining conditions, grouping strategies and clustering stages. This analysis aimed to contribute by assessing whether simple modifications to the original pipeline could highlight its potential for eventual downstream use in the polyp segmentation task. We propose an end framework, using the original part co-segmentation as its base.

As expected, fine-tuning the extractor on colonoscopy data using the **DINO** framework leads to improved segmentation results. Additionally, a more specific grouping of the data further enhances performance, with the best results achieved through the multiple sisters pipeline. Beyond quantitative performance, the multiple sisters approach provides interpretable results by identifying common segmented parts across the target image and multiple reference examples. This demonstrates an understanding of semantic parts, effectively "validating" the polyp segmentation by highlighting other shared segments. Moreover, it generates uncertainty maps based on the different weighted votings, which can serve as a reference for less reliable segmentation areas or global outputs that may require further clinical evaluation. Besides the high interpretability this method offers, it does not require further training, i.e., it could directly be used on other colonoscopy datasets keeping the same reference group used for evaluation. Although a simple and lightweight method, it achieves a test performance comparable with fully supervised UNet's.

Although promising, this method has some limitations that should be considered. Several tunable parameters significantly influence the final performance, such as the foreground saliency threshold, the voting percentage, and the number of common segmented parts. This is particularly bothersome because it requires adapting these parameters experimentally for our own dataset. Furthermore, we found that some of the parameter choices from the original paper's implementation did not benefit our specific task or were insufficient for this more challenging setting. Although the authors highly recommended using three stages for smaller data groups and suggested that larger groups yield the best performance, our observations showed the opposite—performance was better with two stages, especially in smaller groups.

For future work, this method should be evaluated more thoroughly against established self-supervised and supervised baselines. Additionally, we would develop an adaptive approach for selecting optimal parameters, a similar concept to nnU-Net's fingerprint extraction strategy, to improve adaptability and robustness across datasets, eliminating the need for manual parameter tuning.

In sum, our contribution for the current section lies in offering a novel approach that combines a simple segmentation method with interpretability while also underlining its current limitations and future directions. The promising results from the part co-segmentation approach, particularly the multiple sisters pipeline, demonstrated the potential of leveraging self-supervised **ViT** descriptors for polyp segmentation. However, methods developed without direct supervision are inherently limited. In order to further improve segmentation accuracy, we turn to a supervised segmentation framework introduced in the next section, building on the strengths of the explored self-supervised representations.

6.3 KeyViTSeg: Supervised Polyp Segmentation

The DINO-ViT ‘key’ descriptors were further explored to improve polyp segmentation performance when enhanced within a supervised framework.

Figure 6.21 illustrates the proposed KeyViTSeg architecture, which consists of three main components: a **DINO** pretrained **ViT** extractor, a projection layer, and a **CNN**-based decoder. Rather than extracting the tokens from the deepest **ViT** layer, we focus on the key facets.

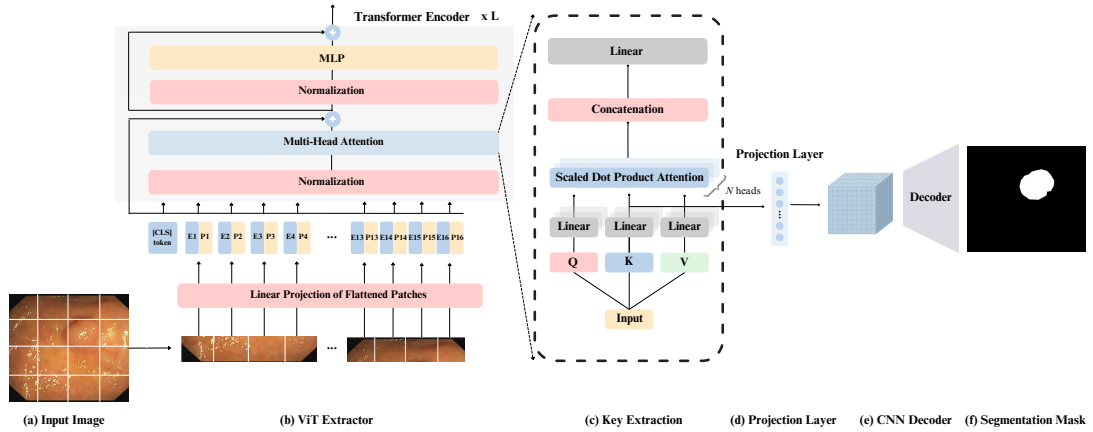


Figure 6.21: KeyViTSeg pipeline. The input image (a) is passed through a ViT encoder (b). Key features are extracted from the multi-head attention block of the last layer (11) of the ViT extractor (c). A projection layer reshapes the ViT features to match the spatial dimensions expected by the CNN decoder (d). The reshaped features are input to the decoder (e), generating a segmentation mask (f).

The method goes as follows:

1. The input image is divided into patches, which are then flattened.
2. Using a fully connected layer, the flattened patches are projected into embeddings. Since transformers do not have a built-in sense of spatial information, positional encodings are added to the patch embeddings. They hold the information regarding the position of the patch in the image.
3. After normalization, these embeddings go through a **MHSA** block, where attention is computed. We take the key representations from the multi-head attention module of the last layer (11) of the **ViTs8** extractor.
4. These descriptors do not have a compatible shape with the **CNN** decoder, so in this step we introduce a projection layer.
5. The projected descriptors are lastly processed by the **CNN** decoder and a segmentation mask is obtained.

6.3.1 Ablation Study on the Extractor Configurations

We conducted an ablation study on the extractor’s configurations in order to choose the best pretraining strategy for end evaluation. Table 6.3 presents these results. The extractor column lists the configurations tested, with the pretrained model specified for each. The fine-tuned column indicates whether the extractor underwent further training. In the case of self-supervised extractors, i.e., those that did not undergo supervised fine-tuning, the decoder was trained while keeping the extractor frozen.

The ViTs8 extractor shows a marked improvement in performance as the number of SSL epochs increases, reaching its best performance with the SSL200 + SL300 setting. In contrast, the ViTs16 extractor pretrained on GastroNet has slightly lower performance overall but still produces competitive results. With these results in mind, the choice of extractor configuration was ViTs8 + SSL200 + SL300.

Table 6.3: Ablation results for varying extractor settings. The best performance is represented in **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Dice	Recall	Accuracy
DINO ViTs8	ImageNet	PolypGen SSL (50)	0.6459	0.6218	0.9621
DINO ViTs8	ImageNet	PolypGen SSL (100)	0.6531	0.6085	0.9612
DINO ViTs8	ImageNet	PolypGen SSL (200)	0.6499	0.6248	0.9621
DINO ViTs8	ImageNet	PolypGen SL (300)	<u>0.7945</u>	0.6939	<u>0.9769</u>
DINO ViTs8	ImageNet	PolypGen SSL (100) + SL (300)	0.7883	0.7198	0.9752
DINO ViTs8	ImageNet	PolypGen SSL (200) + SL (300)	0.7964	<u>0.7016</u>	0.9782
DINO ViTs8	ImageNet	-	0.7180	0.6714	0.9704
DINO ViTs16	GastroNet	-	0.7180	0.6238	0.9673
DINO ViTs16	GastroNet	PolypGen SL (300)	0.7184	0.6360	0.9707

6.3.2 Generalization Tests

In this section, we focus on evaluating the generalizability of the proposed model by comparing it to several well-established baselines. We selected U-Net [3], nnU-Net [45], UM-Net [4], and its base architecture as comparison models due to their relevance in the field of image segmentation. U-Net is a foundational model in the image segmentation domain, widely regarded as a benchmark for various segmentation tasks, and its decoder block architecture matches the one used in our proposed model. nnU-Net, known for its impressive adaptability and performance across a wide range of image segmentation tasks, is a SOTA model, making it also a natural choice for comparison. UM-Net, introduced in 2024, is currently the SOTA model in the field of polyp segmentation.

While the comparative performance across these baselines offers valuable insights into the strengths of our proposed model, it is important to consider the decoder architecture as a potential confounding factor. The decoder used in our model consists of four standard U-Net decoder

blocks, which are similar to those in UM-Net. However, due to UM-Net’s reliance on refined skip connections from the encoder, which KeyViTSeg does not have, a direct and meaningful integration of their exact blocks into our architecture was not possible. To ensure transparency on these differences, we conducted a comparative analysis between KeyViTSeg’s decoder and UM-Net’s decoder, shown in Table 6.4. Therefore, while improvements in KeyViTSeg’s performance should not be attributed to these minor architectural variations in the decoder, we acknowledge that this structural difference could have some influence on the presented results.

Table 6.4: Comparison between KeyViTSeg and UM-Net Decoder Architectures

Features	KeyViTSeg Decoder	UM-Net Decoder
Core Blocks	Bilinear upsampling $\rightarrow$ double convolution. The first convolution reduces the number of input channels by half, followed by a second convolution that maps to the output channels. Both convolution layers use batch normalization and ReLU activation.	Double convolution $\rightarrow$ bilinear upsampling. The first convolution maps the input from the number of input channels to a quarter of the input channels, followed by a second convolution that maps to the output channels. Both convolution layers use batch normalization and ReLU activation.
Number of Core Blocks	4	4
Skip Connections	$\times$	$\checkmark$

We evaluate KeyViTSeg against the chosen baselines across two generalization tests: the **Domain Generalization (DG)** and the **Extreme Single Domain Generalization (ESDG)** tests [87]. These tests are designed to simulate real-world deployment scenarios, where models are required to perform well on unseen data from different domains or environments.

DG Test. This test follows the leave-one-domain-out protocol. In this setup, the model is trained using data from multiple centers, but the performance is assessed by excluding one center at a time during the training phase, testing the model’s ability to generalize to that unseen domain. This is a challenging test since it evaluates how well the model can handle data from entirely new domains that were not part of its training process.

ESDG Test. The ESDG test is an even more stringent evaluation, simulating a real-world scenario where the model is trained on a single domain (in this case, one center’s data) and then tested on the remaining centers. This test assesses the ability of the model to generalize to other domains without having seen them during training, in a much more data-limited setting.

Implementation details. These experiments were all performed on NVIDIA RTX8000 for hardware consistency using 12 different folds. For all cases, data was split by patient avoiding any data leakage. Furthermore, for each fold, the ViT extractor of the proposed model was finetuned with DINO in the corresponding training set, ensuring no data leakage. The models were trained for

300 epochs using a learning rate of 10^{-5} and the Adam Optimizer [82], keeping a constant batch size of 4. Data augmentation techniques include horizontal and vertical flipping, cropping and color jittering. UM-Net also incorporates color transfer, which was proposed by the authors. All models had their input images resized to 224x224, except for nnU-Net, which uses its own pre-processing steps. For penalization, we used the standard dice loss (Equation 4.4). The inference checkpoint was chosen based on the maximum validation DSC (Equation 4.3) throughout epochs.

The results are presented in Table 6.5, having marked in bold the best performing case. The results from both the DG and ESDG tests indicate that the proposed model consistently delivers superior generalization performance compared to SOTA methods. Notably, even when trained on a single domain, the model still maintains high performance across diverse unseen centers. This is particularly important for practical applications in medical imaging, where domain-specific variations are common, and models must be adaptable to new, unseen data without requiring retraining.

U-Net exhibits the largest performance drop from the DG to the ESDG test. This model, while effective in standard conditions, struggled significantly when trained on a single domain and tested on others, which highlights its limited generalization capacity and high sensitivity to domain shifts. nnU-Net, contrary to expectations, performed only slightly better than U-Net in most scenarios. UM-Net demonstrated the best generalization capacity among the evaluated baselines. It showed a more stable performance across both the DG and ESDG tests compared to U-Net and nnU-Net. However, despite its superior performance relative to the other two models, UM-Net still lagged behind the proposed model, especially in the ESDG tests. Additionally, it does not show a marked improvement when compared to its base architecture. So, while this model remains competitive, it still does not match the robustness and cross-domain adaptability achieved through KeyViTSeg.

These findings are significant, demonstrating KeyViTSeg’s robustness to domain shifts, a crucial characteristic for deploying AI models in real-world medical environments where data often come from heterogeneous sources. Furthermore, its high performance even in single-domain training underlines its reliable applicability in real-world cases where training data is limited. The consistently high performance of the proposed model across challenging settings further highlights that the DINO key descriptors capture transversal and clinically significant patterns in the data, being quite robust to the domain-specific variability.

Table 6.5: Results for the Domain Generalization Test (DG test) and the Extreme Single-Domain Generalization Test (ESDG test), comparing the proposed KeyViTSeg model and the baselines. The subscripts presented on the model names indicate the fold of the test. Evaluation of $Model_c$ under a DG test indicates that the model was trained on all centers excluding c for evaluation. Similarly, $Model_c$ for an ESDG test indicates that the data from c was used for model training, leaving all other domains for testing. Best performance is depicted in **bold**.

Model _{fold}	Dice _{DG}	Recall _{DG}	Accuracy _{DG}	Dice _{ESDG}	Recall _{ESDG}	Accuracy _{ESDG}
U-Net ₁	0.7142	0.7488	0.9598	0.4237	0.5019	0.9191
nnU-Net ₁	0.7342	0.7795	0.9590	0.5563	0.6920	0.9082
UM-Net _{base1}	0.8032	0.8242	0.9721	0.6304	0.6854	0.9541
UM-Net ₁	0.8044	0.8408	0.9631	0.6441	0.6450	0.9600
Proposed ₁	0.8387	0.8553	0.9754	0.7109	0.7239	0.9682
U-Net ₂	0.6201	0.6464	0.9483	0.3941	0.4008	0.9299
nnU-Net ₂	0.7068	0.6975	0.9585	0.5324	0.5789	0.9423
UM-Net _{base2}	0.7573	0.7373	0.9728	0.5889	0.6494	0.9496
UM-Net ₂	0.7600	0.7375	0.9728	0.5786	0.6750	0.9350
Proposed ₂	0.8187	0.7392	0.9777	0.6829	0.7296	0.9535
U-Net ₃	0.7334	0.7412	0.9673	0.4888	0.5024	0.9389
nnU-Net ₃	0.8085	0.8104	0.9733	0.6012	0.6966	0.9397
UM-Net _{base3}	0.8426	0.8418	0.9810	0.6446	0.6952	0.9579
UM-Net ₃	0.8515	0.8549	0.9800	0.6451	0.6811	0.9564
Proposed ₃	0.8787	0.8722	0.9831	0.7183	0.7246	0.9672
U-Net ₄	0.4138	0.3313	0.9489	0.3080	0.6428	0.7872
nnU-Net ₄	0.3747	0.4406	0.9419	0.3774	0.4171	0.9134
UM-Net _{base4}	0.4999	0.4479	0.9422	0.5837	0.6388	0.9512
UM-Net ₄	0.5783	0.4125	0.9643	0.5533	0.5406	0.9495
Proposed ₄	0.5875	0.4583	0.9649	0.6825	0.7265	0.9610
U-Net ₅	0.4938	0.5163	0.9648	0.3486	0.5502	0.8698
nnU-Net ₅	0.4805	0.6096	0.9254	0.5583	0.5795	0.9422
UM-Net _{base5}	0.5426	0.5500	0.9708	0.6456	0.6573	0.9534
UM-Net ₅	0.5337	0.5435	0.9667	0.6408	0.6304	0.9533
Proposed ₅	0.6128	0.5890	0.9728	0.7241	0.7133	0.9669
U-Net ₆	0.6698	0.6149	0.9613	0.3260	0.3307	0.9236
nnU-Net ₆	0.7043	0.7029	0.9619	0.4892	0.5134	0.9435
UM-Net _{base6}	0.7356	0.7633	0.9702	0.5689	0.6378	0.9452
UM-Net ₆	0.7582	0.7181	0.9807	0.5841	0.6518	0.9497
Proposed ₆	0.8260	0.7794	0.9880	0.6660	0.7102	0.9610

Figure 6.22 presents the distribution of **DSCs** across all evaluated models. This visualization allows for a clearer comparison of model performance by illustrating the range, median, and variability of **DSCs**.

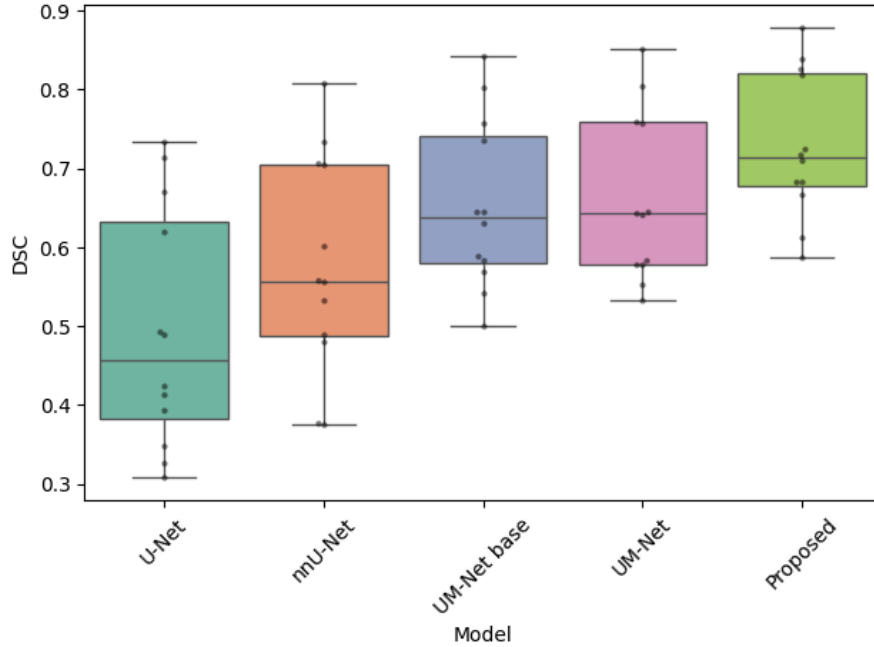


Figure 6.22: Boxplot showing Dice Score Coefficient (DSC) distribution across models.

U-Net exhibits the widest box, indicating greater variability in DSC values across folds compared to the other models. Both U-Net and nnU-Net show long whiskers, reflecting high variability in performance. The outliers for nnU-Net resemble those of U-Net, suggesting that certain cases performed considerably worse than others. UM-Net and its baseline architecture show improved performance compared to U-Net and nnU-Net; however, the performance difference between UM-Net and its baseline is minimal. The proposed model demonstrates the smallest variability, as evidenced by the narrow box and short whiskers, reflecting its more consistent performance across different folds. Furthermore, we achieve the highest median **DSC**, indicating superior overall performance to the other models. Notably, the boxes are of unequal size, highlighting the differences in performance between the **DG** and **ESDG** cases.

6.3.3 Statistical Evaluation

We aim to fully complement and support the above evaluation by assessing the statistical significance of the obtained **DSC** results using the Friedman test [88]. **DSC** was chosen as it is considered the most representative metric for evaluating segmentation performance.

The Friedman test serves as a non-parametric alternative to the one-way repeated measures ANOVA. It is used for the comparison between different subjects (in this case, folds) under different conditions (models). It works by ranking the data within each block (or row), rather than using

the raw values, making it less sensitive to outliers and distributions that don't meet the normality assumption. The null hypothesis H_0 for this test assumes that the medians between groups are not significantly different. In this case, it would mean that there is not a significant difference between the performances of the evaluated models. If the null hypothesis is rejected, at least one model is statistically different from the others [88].

In Table 6.6, we present the results for the statistical analysis. For a significance of 0.05, H_0 is rejected when the obtained p -value is inferior to that significance. The results point to a p -value of 5×10^{-9} , a much lower value than the significance threshold, translating to the rejection of the null hypothesis. In other words, there are significant differences between the performances of the different models. Kendall's W is a measure of agreement between the ranks of the models across folds. It ranges from 0 to 1, 0 meaning no agreement and 1 perfect agreement. We obtained a value of 0.9278, thus suggesting strong agreement in ranking of the evaluated models across folds. Similarly, Q measures the differences between groups being tested. A higher value indicates a larger difference.

Table 6.6: Friedman test results

Source	Kendall's W	Degrees of freedom	Q	p -value
Friedman	0.9278	4	44.53	0.000000004971

Next, we conduct a *post-hoc* analysis for pairwise comparisons using the Wilcoxon signed-rank test [89]. These tests allows us to determine the best performing model and the significance between paired results. Results for the pairwise comparison are present in Table 6.7. W in this table indicates the Wilcoxon signed-rank test statistic. This indicates the cumulative sum of the ranks for the differences observed between paired values. A small W -value (close to 0) indicates a significant difference between the two groups, with the majority of differences in the same direction, while a large W -value suggests a weaker or non-significant difference, with more evenly distributed differences.

Table 6.7: Wilcoxon signed-rank test results for different model pairs

Model A	Model B	W	p-value
Proposed	U-Net	0.0	0.000488
Proposed	UM-Net	0.0	0.000488
Proposed	UM-Net base	0.0	0.000488
Proposed	nnU-Net	0.0	0.000488
U-Net	UM-Net	0.0	0.000488
U-Net	UM-Net base	0.0	0.000488
U-Net	nnU-Net	5.0	0.004883
UM-Net	UM-Net base	27.5	0.423828
UM-Net	nnU-Net	0.0	0.000488
UM-Net base	nnU-Net	0.0	0.000488

All pairs show to be significantly different, with p – values lower than the significance threshold (0.05), with the exception of UM-Net and its base architecture. This reinforces the reliability and statistical significance of our proposed model’s performance improvements over existing baselines, making it a competitive choice for polyp segmentation.

Based on this analysis, we can conclude that the proposed model demonstrates statistically significant superior performance compared to the other approaches.

6.3.4 Computational Cost Analysis

In this subsection, we conduct a brief comparative analysis of the model sizes and the time taken for a single inference. Results for this assessment are presented in Table 6.8.

The classical U-Net exhibits the lowest number of parameters and inference time. The parameters to present for nnU-Net were calculated based on an architecture specifically adapted to this dataset, included in their architectural plans. While it has the highest number of parameters and overall complexity, it still achieves the best inference time after the classical U-Net. UM-Net has the highest inference time but manages to maintain the second lowest number of parameters. The proposed model falls between these two, offering improved performance without introducing additional complexity.

Table 6.8: Evaluation of model size and inference time. Best result is depicted in **Bold**.

Model	Params (M)	Inference Time (ms)
U-Net	17.27	2.6
nnU-Net	59.18	6.6
UM-Net base	21.58	5.9
UM-Net	22.79	15.3
Proposed	31.47	10.6

6.3.5 Qualitative Results

Figure 6.23 presents visual segmentation results for **ESDG** test experiments, being the most critical. These results reinforce the conclusions already extracted from the quantitative analysis.

U-Net tends to oversegment, often including adjacent areas in the final segmentation and struggling with accurate boundary delineation. nnU-Net performs better but still faces challenges, particularly with medical instruments and, like U-Net, it tends to oversegment the lesion. Additionally, both models struggle with chromoendoscopy images, as seen in first and last examples, showing complete segmentation failure. UM-Net and its base architecture yield results closer to the ground truth but still encounter difficulties, especially with the presence of chromoendoscopy and image annotations. In contrast, the proposed model demonstrates resilience to non-anatomical objects, chromoendoscopy, and smaller polyp sizes. It also exhibits superior sensitivity to boundary cues, consistently providing segmentation results that closely match the ground truth.

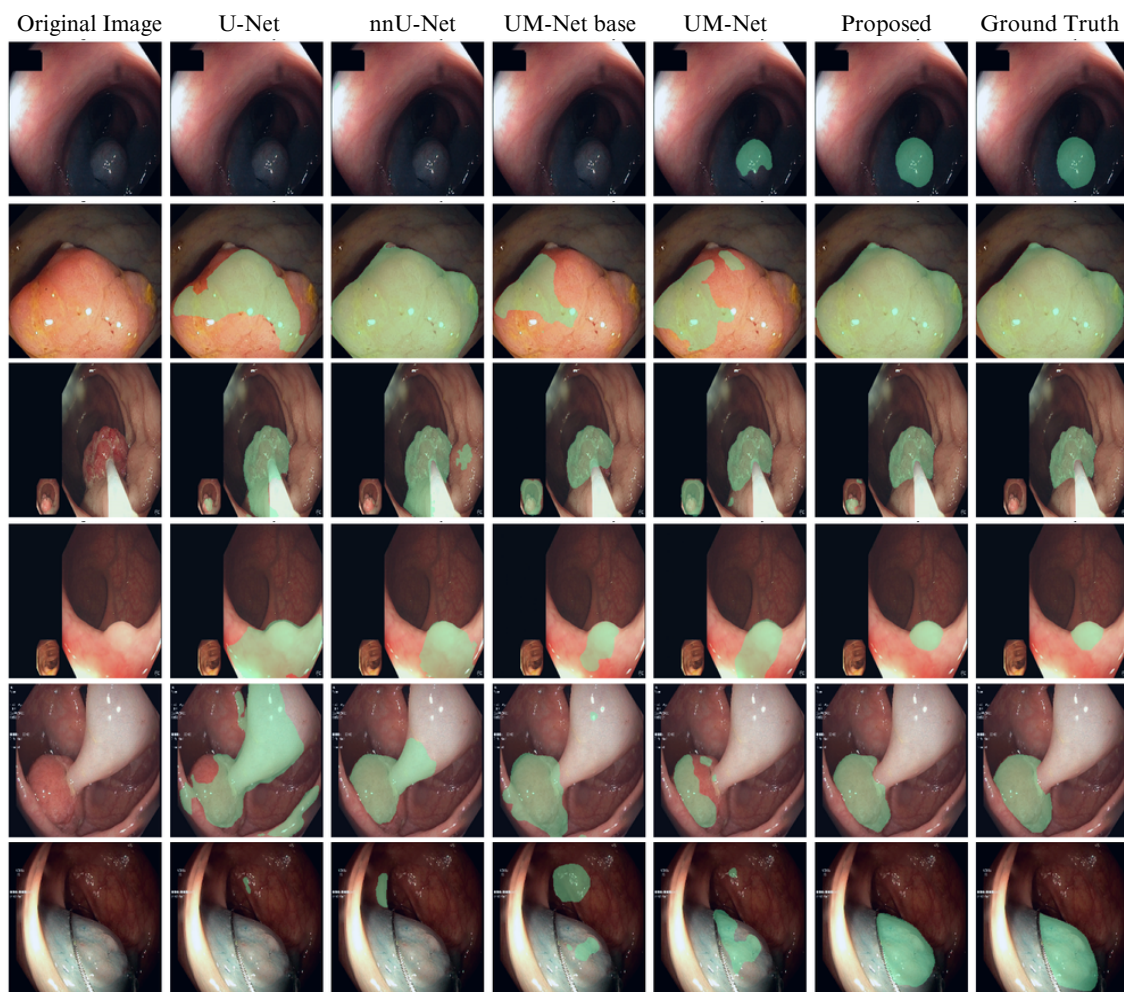


Figure 6.23: Qualitative results for the Extreme Single Domain Generalization (ESDG) tests.

In Figure 6.24, we present failure cases from the proposed model, highlighting instances where it struggles to accurately identify polyps. Interestingly, the behaviours in these failure cases are similar between the proposed model and UM-Net, particularly when the model misidentifies other structures as polyps. Commonly difficult cases include highly camouflaged polyps, as represented in the first example, and elongated polyps that closely resemble ordinary colon folds, as seen in the second example. Another significant challenge is the inclusion of medical instruments, especially when combined with chromoendoscopy, as the whiter color of the instrument in the midst of the chromo dye makes it resemble the polyp, as shown in the final example. The misidentified segmentation in the last example also appears to derive from a slightly lighter and protuberant area in the tissue with blue dye. Furthermore, while both UM-Net and the proposed model misidentify non-polyp structures as polyps in these examples, U-Net and nnU-Net output empty masks. In the first example, nnU-Net presents an exception by correctly identifying the polyp, along with other structures, despite the challenges presented by camouflaged polyps.

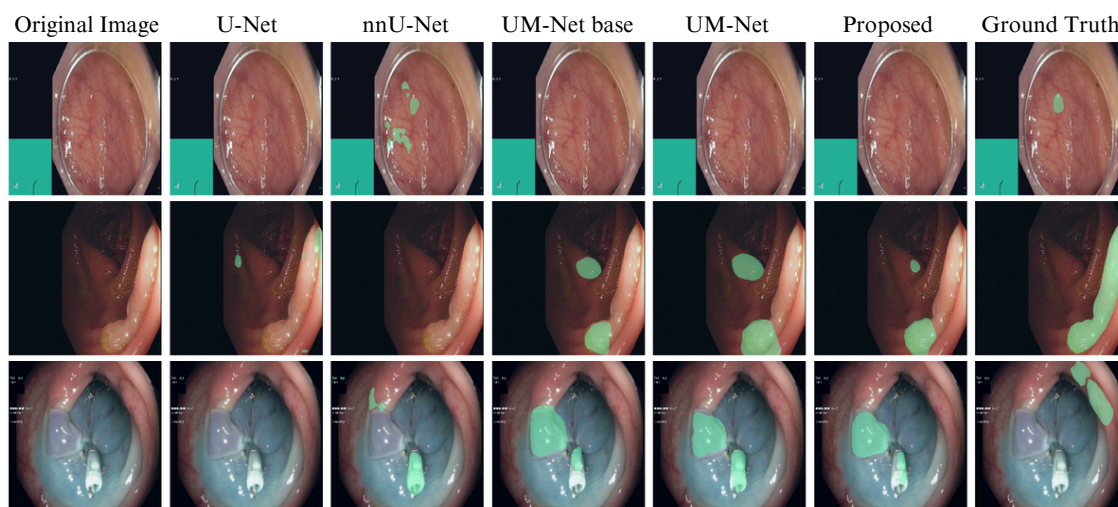


Figure 6.24: Examples of failed segmentations.

Looking both at quantitative and qualitative results, we can conclude the proposed model consistently outperforms baselines, showing high sensitivity to lesion cues in the input even when smaller or in the presence of more challenging conditions. Furthermore, its ability to consistently deliver high-quality segmentations, even when trained on single-center data, highlights its capacity to capture meaningful clinical patterns while disregarding domain-specific information that should not influence its predictions.

6.4 Conclusions

In this Chapter, we explore **DINO-ViT** "key" facets as the answer to reduce the impact of the domain shifts in polyp segmentation performance. We started by performing semantic tasks with these features, thus evaluating the relevance of the image content captured in these encodings. This included finding semantic correspondences by matching pairs of descriptors from two images, where we observed matches between similar structures such as polyp boundaries and medical instruments. We also used the similarity between these descriptors to retrieve similar image and polyp pairs, achieving highly accurate matches. By analyzing key progression across **ViT** layers, we observed similar descriptor representations for inter-example polyps and a highly detailed representation of semantic structures, such as blood vessels. This initial analysis emphasizes the representational power of these features, making them ideal for addressing the medical bias issue.

Due to the observations in the first section, before moving to a supervised approach, we explored a self-supervised polyp segmentation pipeline based on descriptor clustering. By simply clustering descriptors of a group of images, we can obtain interpretable segmented common parts, including polyps. We evaluated this performance with higher fine-tuning of the extractor in colonoscopy data and across progressively more specific image groups. The proposed multiple sisters approach achieved the best performance in the ablation studies, highlighting its potential and justifying its further comparison against a fully supervised and well-established segmentation

pipeline. The results of 5-fold cross-validation between the multiple sisters approach and the fully supervised U-Net show that, while the proposed method does not match the performance of the supervised baseline, it demonstrates promising potential. The inclusion of automatic part selection resulted in a predictable drop in performance, highlighting the need for further refinement. Additionally, the method relies on a considerable number of predefined parameters, which currently require manual tuning or adaptation to achieve optimal performance for each specific case. In our experiments, we used the original parameter values; however, for better results, these parameters should be automatically adjusted according to the characteristics of each case, thus reducing the need for manual intervention and enhancing the method's flexibility and efficiency.

Finally, we propose a simple encoder-decoder architecture using the explored **DINO ViT** key features and evaluate it extensively against selected baseline models: U-Net, no-new U-Net, UM-Net and its baseline architecture. By employing these features, our model achieves improved **SOTA** results for polyp segmentation in **OOD** settings.

Future work should include further analysis on other colonoscopy datasets, and focus on applying this architecture to video segmentation, the emerging trend in polyp segmentation. This would ideally enable real-time polyp segmentation during colonoscopy exams. For this, several key considerations need to be addressed. First, the model must make predictions faster than the real-time frame rate, which is typically 30 FPS in colonoscopy exams [90, 91]. This means the model's inference time should be under 33ms (1s/30). Our inference time is currently 10.6ms, which is sufficient. Additionally, we would have to move from treating each sample independently to including time-related information. For example, the model's predictions could be regularized by looking at previous frames' predictions and promoting consistency.

Furthermore, we advocate for exploring the use of these descriptors in other domains beyond colonoscopy, as their robustness could streamline approaches, enabling their application across various fields without the need for complex preprocessing or extensive dataset adaptation. However, we do recognize that the colonoscopy domain is much closer to ImageNet's than Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) data. Nevertheless, we believe that with further fine-tuning, these descriptors could be extended to medical imaging domains, potentially leading to significant advancements in the segmentation of diverse medical datasets.

To conclude, the analyses presented in this chapter culminate in an efficient polyp segmentation approach that demonstrates strong generalizability to unseen data. This approach not only opens a promising line of research for video segmentation but also holds great potential for expanding to other imaging techniques and cancer types, further advancing the field of medical image analysis.

Chapter 7

Conclusions

This dissertation explored techniques to improve the generalizability of polyp segmentation models, particularly by enhancing their robustness to data variability and domain shifts.

In the first stages of this work, we explored specifically the chromoendoscopy bias and ways to mitigate it. This bias arises from the use of contrast-enhanced colonoscopy images, which introduce color variations that the model might overly rely on, creating a shortcut for polyp detection. As a result, the model’s reliance on these non-structural features reduces its ability to generalize across different data distributions. We started by demonstrating that the progressive inclusion of chromoendoscopy samples in model training leads to worse out-of-distribution performance and, therefore, generalization. Given the observed relevance of this bias, we explored ways to mitigate it. This included traditional pre-processing techniques and color invariance augmentation methods. Traditional pre-processing methods, such as color normalization and histogram equalization, did not consistently mitigate this bias. Their effectiveness varied across different model architectures, which led us to explore more advanced color invariance methods. Color invariance methods are augmentation strategies that aim to reduce model focus on color information when making predictions. With this, the segmentations would be more focused on structural details of the input rather than color cues, therefore reducing the impact of chromoendoscopy. The explored techniques improved performance; however, as model generalization remained limited by other unaddressed biases, we shifted to a broader approach, aiming to tackle these biases as well.

The main contributions of this thesis are detailed in Chapter 6, where we explored **DINO ViT** key features to improve generalization and feature robustness. The experiments showed that these descriptors are effective in enhancing polyp segmentation performance. These descriptors better capture relevant anatomical features for the current task while showing resilience to domain-specific biases. By leveraging these self-supervised features, we proposed both a self-supervised and a supervised framework. The first showed promising results due to its simplicity, but its performance still lags behind fully supervised U-Net. Additionally, the method requires experimental adaptation of parameters, which is not ideal. Future work will focus on automating this parameter adaptation to improve segmentation outcomes and increase the approach’s flexibility, as well as validating it against other state-of-the-art self-supervised and supervised approaches. Then,

by refining these descriptors within a simple supervised framework, proposing KeyViTSeg, we achieved improved **SOTA** generalization results. For future work, we plan to extend the validation of this approach to other colonoscopy datasets, further demonstrating its generalization capabilities. Additionally, we intend to apply the proposed approach to video segmentation, enabling real-time polyp detection during colonoscopy procedures. It would also be valuable to test this architecture on other medical segmentation tasks to determine whether the enhanced feature quality can be generalized to more complex imaging techniques such as CT or MRI.

The work presented in Chapter 6 will lead to a paper submission for the Winter Conference on Applications of Computer Vision (WACV) 2026.

References

- [1] S. Ali, D. Jha, N. Ghatwary et al., “A multi-centre polyp detection and segmentation dataset for generalisability assessment,” Scientific Data, vol. 10, p. 75, 2023.
- [2] P. Davis, “Colorectal polyps as precursors to colorectal malignancy,” RGA, 2022, accessed: 2025-06-14. [Online]. Available: <https://www.rgare.com/knowledge-center/articles/colorectal-polyps-as-precursors-to-colorectal-malignancy>
- [3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” CoRR, vol. abs/1505.04597, 2015. [Online]. Available: <http://arxiv.org/abs/1505.04597>
- [4] X. Du, X. Xu, J. Chen, X. Zhang, L. Li, H. Liu, and S. Li, “Um-net: Rethinking icgnet for polyp segmentation with uncertainty modeling,” Medical Image Analysis, vol. 99, p. 103347, 2025.
- [5] M. Bi, M. Wang, Z. Li, and D. Hong, “Vision transformer with contrastive learning for remote sensing image scene classification,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. PP, no. 99, pp. 1–13, 2022. [Online]. Available: <https://doi.org/10.1109/JSTARS.2022.3230835>
- [6] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He, “A survey of visual transformers,” IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 6, pp. 7478–7498, 2024.
- [7] V. Rani, S. T. Nabi, M. Kumar, A. Mittal, and K. Kumar, “Self-supervised learning: A succinct review,” Archives of Computational Methods in Engineering, vol. 30, no. 4, pp. 2761–2775, 2023, published on 2023-05-01. [Online]. Available: <https://doi.org/10.1007/s11831-023-09884-2>
- [8] S. Albelwi, “Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging,” Entropy, vol. 24, no. 4, 2022. [Online]. Available: <https://www.mdpi.com/1099-4300/24/4/551>
- [9] F. Ros and R. Riad, Deep clustering techniques based on CNN. Cham: Springer Nature Switzerland, 2024, pp. 159–201. [Online]. Available: https://doi.org/10.1007/978-3-031-48743-9_10
- [10] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 9650–9660.

- [11] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *CoRR*, vol. abs/1802.02611, 2018. [Online]. Available: <http://arxiv.org/abs/1802.02611>
- [12] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, “Deep vit features as dense visual descriptors,” *arXiv preprint arXiv:2112.05814*, vol. 2, no. 3, p. 4, 2021.
- [13] M. Gupta and A. Mishra, “A systematic review of deep learning based image segmentation to detect polyp,” *Artificial Intelligence Review*, vol. 57, no. 7, 2024. [Online]. Available: <https://doi.org/10.1007/s10462-023-10621-1>
- [14] Z. Wu, F. Lv, C. Chen, A. Hao, and S. Li, “Colorectal polyp segmentation in the deep learning era: A comprehensive survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 1–1, 2024, arXiv preprint.
- [15] C. Clinic. (2022, Jun.) Why colon polyp size, shape and growth pattern matters. Accessed: 2024-09-20. [Online]. Available: <https://health.clevelandclinic.org/colon-polyp-cancer-risk/>
- [16] M. Illimoottil and D. Ginat, “Recent advances in deep learning and medical imaging for head and neck cancer treatment: Mri, ct, and pet scans,” *Cancers*, vol. 15, no. 13, p. 3267, 2023.
- [17] L. Stelter, V. Corbetta, R. Beets-Tan, and W. Silva, “Assessing the impact of federated learning and differential privacy on multi-centre polyp segmentation,” in *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2024, pp. 1–4.
- [18] L. Zhang, X. Wang, D. Yang, T. Sanford, S. Harmon, B. Turkbey, B. J. Wood, H. Roth, A. Myronenko, D. Xu, and Z. Xu, “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.
- [19] L. Luo, X. Huang, M. Wang, Z. Wan, W. Ma, and H. Chen, “Mitigating medical dataset bias by learning adaptive agreement from a biased council,” *Medical Image Analysis*, vol. 105, p. 103629, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1361841525001768>
- [20] F. H. Al-Kawas, “Chromoendoscopy,” *Gastroenterology & Hepatology*, vol. 2, no. 2, pp. 101–103, 2006. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5335624/>
- [21] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.
- [22] A. Tendle and M. R. Hasan, “A study of the generalizability of self-supervised representations,” *Machine Learning with Applications*, vol. 6, p. 100124, 2021.
- [23] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [24] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2022.

- [25] S. Li, Y. Ren, Y. Yu, Q. Jiang, X. He, and H. Li, "A survey of deep learning algorithms for colorectal polyp segmentation," *Neurocomputing*, p. 128767, 2024.
- [26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [27] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds. Cham: Springer International Publishing, 2018, pp. 3–11.
- [28] S. Safarov and T. K. Whangbo, "A-denseunet: Adaptive densely connected unet for polyp segmentation in colonoscopy images with atrous convolution," *Sensors*, vol. 21, no. 4, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/4/1441>
- [29] X. Zhao, L. Zhang, and H. Lu, "Automatic polyp segmentation via multi-scale subtraction network," in *Proceedings of the 25th International Conference on Medical Image Computing and Computer-Assisted Intervention*, L. Wang, Q. Dou, P. T. Fletcher, and et al., Eds. Cham: Springer, 2021, pp. 120–130.
- [30] T. Nguyen-Mau, D. Dang-Nguyen, and et al., "Pefnet: Positional embedding feature for polyp segmentation," in *MultiMedia Modeling. MMM 2023. Lecture Notes in Computer Science*. Springer, Cham, 2023, vol. 13834. [Online]. Available: https://doi.org/10.1007/978-3-031-27818-1_20
- [31] J. Mei, T. Zhou, K. Huang, and et al., "A survey on deep learning for polyp segmentation: techniques, challenges and future trends," *Vis. Intell.*, vol. 3, no. 1, 2025. [Online]. Available: <https://doi.org/10.1007/s44267-024-00071-w>
- [32] B. Murugesan, K. Sarveswaran, S. M. Shankaranarayana, K. Ram, J. Joseph, and M. Sivaprakasam, "Psi-net: Shape and boundary aware joint multi-task deep network for medical image segmentation," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 7223–7226.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [34] D. Jha, P. H. Smedsrud, M. A. Riegler, D. Johansen, T. de Lange, P. Halvorsen, and H. D. Johansen, "Resunet++: An advanced architecture for medical image segmentation," 2019. [Online]. Available: <https://arxiv.org/abs/1911.07067>
- [35] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.
- [36] R. Zhang, G. Li, Z. Li, S. Cui, D. Qian, and Y. Yu, "Adaptive context selection for polyp segmentation," in *Proceedings of the 23rd International Conference on Medical Image Computing and Computer-Assisted Intervention*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, and et al., Eds. Cham: Springer, 2020, pp. 253–262.

- [37] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, “Pranet: Parallel reverse attention network for polyp segmentation,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2020, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds. Cham: Springer International Publishing, 2020, pp. 263–273.
- [38] S. Jain, R. Atale, A. Gupta, U. Mishra, A. Seal, A. Ojha, J. Jaworek-Korjakowska, and O. Krejcar, “Coinnet: A convolution-involution network with a novel statistical attention for automatic polyp segmentation,” IEEE Transactions on Medical Imaging, vol. 42, no. 12, pp. 3987–4000, 2023.
- [39] T. D. Huy, H. Cao Huyen, C. D. T. Nguyen, S. T. M. Duong, T. Bui, and S. Q. H. Truong, “Adversarial contrastive fourier domain adaptation for polyp segmentation,” in 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI), 2022, pp. 1–5.
- [40] C. Yang, X. Guo, Z. Chen, and Y. Yuan, “Source free domain adaptation for medical image segmentation with fourier style mining,” Medical Image Analysis, vol. 79, p. 102457, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1361841522001049>
- [41] Y. Shen, Y. Lu, X. Jia, F. Bai, and M. Q.-H. Meng, “Task-relevant feature replenishment for cross-centre polyp segmentation,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2022, L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, Eds. Cham: Springer Nature Switzerland, 2022, pp. 599–608.
- [42] A. Wang, M. Xu, Y. Zhang, M. Islam, and H. Ren, “S²me: Spatial-spectral mutual teaching and ensemble learning for scribble-supervised polyp segmentation,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2023, H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, and R. Taylor, Eds. Cham: Springer Nature Switzerland, 2023, pp. 35–45.
- [43] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” CoRR, vol. abs/1706.05587, 2017. [Online]. Available: <http://arxiv.org/abs/1706.05587>
- [44] J. Zhong, W. Wang, H. Wu, Z. Wen, and J. Qin, “Polypseg: An efficient context-aware network for polyp segmentation from colonoscopy videos,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2020, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds. Cham: Springer International Publishing, 2020, pp. 285–294.
- [45] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Köhler, T. Norajitra, S. J. Wirkert, and K. H. Maier-Hein, “nnu-net: Self-adapting framework for u-net-based medical image segmentation,” CoRR, vol. abs/1809.10486, 2018. [Online]. Available: <http://arxiv.org/abs/1809.10486>
- [46] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Álvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” CoRR, vol. abs/2105.15203, 2021. [Online]. Available: <https://arxiv.org/abs/2105.15203>
- [47] A. Wang, M. Wu, H. Qi, H. Shi, J. Chen, Y. Chen, and X. Luo, “Pyramid transformer driven multibranch fusion for polyp segmentation in colonoscopic video images,” in 2023 IEEE International Conference on Image Processing (ICIP), 2023, pp. 2350–2354.

- [48] T. Ling, C. Wu, H. Yu, T. Cai, D. Wang, Y. Zhou, M. Chen, and K. Ding, “Probabilistic modeling ensemble vision transformer improves complex polyp segmentation,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2023, H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, and R. Taylor, Eds. Cham: Springer Nature Switzerland, 2023, pp. 572–581.
- [49] B. Xiao, J. Hu, W. Li, C.-M. Pun, and X. Bi, “Ctnet: Contrastive transformer network for polyp segmentation,” IEEE Transactions on Cybernetics, vol. 54, no. 9, pp. 5040–5053, 2024.
- [50] Y. Wang, Z. Deng, Q. Lou, S. Hu, K. Choi, and S. Wang, “Cooperation learning enhanced colonic polyp segmentation based on transformer-cnn fusion,” arXiv preprint, 2023.
- [51] S. Li, X. Sui, J. Fu, H. Fu, X. Luo, Y. Feng, X. Xu, Y. Liu, D. S. W. Ting, and R. S. M. Goh, “Few-shot domain adaptation with polymorphic transformers,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2021, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds. Cham: Springer International Publishing, 2021, pp. 330–340.
- [52] D. Jha, N. K. Tomar, V. Sharma, and U. Bagci, “Transnetr: Transformer-based residual network for polyp segmentation with multi-center out-of-distribution testing,” in Medical Imaging with Deep Learning, ser. Proceedings of Machine Learning Research, I. Oguz, J. Noble, X. Li, M. Styner, C. Baumgartner, M. Rusu, T. Heinmann, D. Kontos, B. Landman, and B. Dawant, Eds., vol. 227. PMLR, 10–12 Jul 2024, pp. 1372–1384. [Online]. Available: <https://proceedings.mlr.press/v227/jha24a.html>
- [53] J. Wang, Q. Huang, F. Tang, J. Meng, J. Su, and S. Song, “Stepwise feature fusion: Local guides global,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2022, ser. Lecture Notes in Computer Science, L. Wang, Q. Dou, P. Fletcher, S. Speidel, and S. Li, Eds. Springer, Cham, 2022, vol. 13433. [Online]. Available: https://doi.org/10.1007/978-3-031-16437-8_11
- [54] N. Duc, N. Oanh, N. Thuy, T. Triet, and V. Dinh, “Colonformer: an efficient transformer based method for colon polyp segmentation,” IEEE Access, vol. 10, pp. 80 575–80 586, 2022.
- [55] Y. Zhang, H. Liu, and Q. Hu, “Transfuse: fusing transformers and cnns for medical image segmentation,” in Proceedings of the 24th International Conference on Medical Image Computing and Computer-Assisted Intervention, M. de Bruijne, P. C. Cattin, S. Cotin et al., Eds. Cham: Springer, 2021, pp. 14–24.
- [56] B. Dong, W. Wang, D. Fan, J. Li, H. Fu, and L. Shao, “Polyp-pvt: polyp segmentation with pyramid vision transformers,” arXiv preprint, 2021, arXiv:2108.06932. [Online]. Available: <https://arxiv.org/abs/2108.06932>
- [57] M. Nguyen, T. Bui, Q. van Nguyen, T. Nguyen, and T. van Pham, “Lapformer: a light and accurate polyp segmentation transformer,” arXiv preprint arXiv:2210.04393, 2022.
- [58] L. Cai, M. Wu, L. Chen, W. Bai, M. Yang, S. Lyu et al., “Using guided self-attention with local information for polyp segmentation,” in Proceedings of the 25th International Conference on Medical Image Computing and Computer-Assisted Intervention, L. Wang, Q. Dou, P. T. Fletcher et al., Eds. Springer, 2022, pp. 629–638.

- [59] J. Yao and R. Summers, “Chapter 20 - polyp segmentation on ct colonography,” in Medical Image Recognition, Segmentation and Parsing, ser. The Elsevier and MICCAI Society Book Series, S. K. Zhou, Ed. Academic Press, 2016, pp. 451–484. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128025819000202>
- [60] L. Shan, X. Li, and W. Wang, “Decouple the high-frequency and low-frequency information of images for semantic segmentation,” in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 1805–1809.
- [61] R. Cong, M. Sun, S. Zhang, X. Zhou, W. Zhang, and Y. Zhao, “Frequency perception network for camouflaged object detection,” in Proceedings of the 31st ACM international conference on multimedia, 2023, pp. 1179–1189.
- [62] P. Schutte, V. Corbetta, R. Beets-Tan, and W. Silva, “Fedgs: Federated gradient scaling for heterogeneous medical image segmentation,” arXiv preprint arXiv:2401.11734, 2024, accepted for IEEE Transactions on Pattern Analysis and Machine Intelligence. [Online]. Available: <https://arxiv.org/abs/2401.11734>
- [63] J. Su, D. V. Vargas, and K. Sakurai, “One pixel attack for fooling deep neural networks,” IEEE Transactions on Evolutionary Computation (TEC), vol. 23, no. 5, pp. 828–841, 2019.
- [64] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, “Dense contrastive learning for self-supervised visual pre-training,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 3024–3033.
- [65] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” CoRR, vol. abs/1803.07728, 2018. [Online]. Available: <http://arxiv.org/abs/1803.07728>
- [66] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, “Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 16 684–16 693.
- [67] A. Kalapos and B. Gyires-Tóth, “Self-supervised pretraining for 2d medical image segmentation,” in Computer Vision – ECCV 2022 Workshops, L. Karlinsky, T. Michaeli, and K. Nishino, Eds. Cham: Springer Nature Switzerland, 2023, pp. 472–484.
- [68] A. Ziegler and Y. M. Asano, “Self-supervised learning of object parts for semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 14 502–14 511.
- [69] Z. Feng, C. Xu, and D. Tao, “Self-supervised representation learning by rotation feature decoupling,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
- [70] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), December 2015.
- [71] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 69–84.

- [72] F. Wei, Y. Gao, Z. Wu, H. Hu, and S. Lin, “Aligning pretraining for detection via object-level contrastive learning,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 22 682–22 694. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/bf5cd8b2509011b9502a72296edc14a0-Paper.pdf
- [73] X. Chen and K. He, “Exploring simple siamese representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 750–15 758.
- [74] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, “A survey on self-supervised learning: Algorithms, applications, and future trends,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 9052–9071, 2024.
- [75] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [76] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent - a new approach to self-supervised learning,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 271–21 284. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf
- [77] X. Wang, R. Zhang, C. Shen, and T. Kong, “Densecl: A simple framework for self-supervised dense visual pre-training,” *Visual Informatics*, vol. 7, no. 1, pp. 30–40, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2468502X22000936>
- [78] Y. M. Asano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” *CoRR*, vol. abs/1911.05371, 2019. [Online]. Available: <http://arxiv.org/abs/1911.05371>
- [79] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, “Self-supervised representation learning: Introduction, advances, and challenges,” *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 42–62, 2022.
- [80] R. Singh, K. H. Chiam, F. Leiria, L. Z. C. T. Pu, K. C. Choi, and M. Militz, “Chromoendoscopy: role in modern endoscopic imaging,” *Translational gastroenterology and hepatology*, vol. 5, p. 39, 2020. [Online]. Available: <https://doi.org/10.21037/tgh.2019.12.06>
- [81] P. Trivedi and B. Braden, “Indications, stains and techniques in chromoendoscopy,” *QJM: An International Journal of Medicine*, vol. 106, no. 2, pp. 117–131, 10 2012. [Online]. Available: <https://doi.org/10.1093/qjmed/hcs186>
- [82] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [83] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *CoRR*, vol. abs/1703.10593, 2017. [Online]. Available: <http://arxiv.org/abs/1703.10593>

- [84] Y. Gong, J. Li, L. Chen, and M. Jiang, “Exploring color invariance through image-level ensemble learning,” 2024. [Online]. Available: <https://arxiv.org/abs/2401.10512>
- [85] D. J. Ketchen and C. L. Shook, “The application of cluster analysis in strategic management research: an analysis and critique,” *Strategic management journal*, vol. 17, no. 6, pp. 441–458, 1996.
- [86] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- [87] H. Che, Y. Cheng, H. Jin, and H. Chen, “Towards generalizable diabetic retinopathy grading in unseen domains,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, and R. Taylor, Eds. Cham: Springer Nature Switzerland, 2023, pp. 430–440.
- [88] D. G. Pereira, A. Afonso, and F. M. M. and, “Overview of friedman’s test and post-hoc analysis,” *Communications in Statistics - Simulation and Computation*, vol. 44, no. 10, pp. 2636–2653, 2015. [Online]. Available: <https://doi.org/10.1080/03610918.2014.931971>
- [89] R. F. Woolson, “Wilcoxon signed-rank test,” *Encyclopedia of biostatistics*, vol. 8, 2005.
- [90] B. S. Kim, M. Cho, G. E. Chung, J. Lee, H. Y. Kang, D. Yoon, W. S. Cho, J. C. Lee, J. H. Bae, H.-J. Kong *et al.*, “Density clustering-based automatic anatomical section recognition in colonoscopy video using deep learning,” *Scientific Reports*, vol. 14, no. 1, p. 872, 2024.
- [91] M. Cho, J. H. Kim, K. S. Hong, J. S. Kim, H.-J. Kong, and S. Kim, “Identification of cecum time-location in a colonoscopy video by deep learning analysis of colonoscope movement,” *PeerJ*, vol. 7, p. e7256, 2019.

Appendix A

Segmentation with Self-Supervised ViT Representations Results

In this Appendix, we show some global and per-division quantitative and qualitative results for the part-cosegmentation experiments from Chapter 6, Section 6.2, particularly for the cluster and classes groups. These are the results left out of the main body of the dissertation in order to reduce the high volume while still providing in-depth, complete results.

A.1 Part-co-segmentation results

A.1.1 Global

Table A.1: Part Co-segmentation quantitative results globally (in-distribution and out-of-distribution). In the "Finetuned on" column SSL stands for Self-Supervised Learning, more particularly DINO training and is followed by the number of training epochs; SL means Supervised-Learning and is equally followed by the number of epochs. The best performance is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Two Stages		Three Stages	
			Dice (ID)	Dice (OOD)	Dice (ID)	Dice (OOD)
ViTs8	ImageNet	-	0.2843	0.2567	0.3078	0.2498
ViTs8	ImageNet	PolypGen (100 epochs)	0.2884	0.3527	0.1795	0.2653
ViTs8	ImageNet	PolypGen (200 epochs)	0.3853	0.3583	0.3472	0.2653
ViTs8	ImageNet	PolypGen SSL (100) + SL (300)	0.6733	0.5779	<u>0.5515</u>	<u>0.5225</u>
ViTs8	ImageNet	PolypGen SSL (200) + SL (300)	<u>0.6202</u>	0.5191	0.5193	0.5028
ViTs16	GastroNet	-	0.2183	0.2876	0.2064	0.2225
ViTs16	GastroNet	PolypGen SL (300 epochs)	0.5942	<u>0.5241</u>	0.5888	0.5304

A.1.2 Clusters

A.1.2.1 Quantitative results

Table A.2: Part Co-Segmentation global quantitative results for cluster division (in-distribution and out-of-distribution). The best performance is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Two Stages		Three Stages	
			Dice (ID)	Dice (OOD)	Dice (ID)	Dice (OOD)
ViTs8	ImageNet	-	0.3278	0.2484	0.2551	0.2418
ViTs8	ImageNet	PolypGen (100 epochs)	0.3783	0.3062	0.3716	0.3023
ViTs8	ImageNet	PolypGen (200 epochs)	0.3776	0.3402	0.3947	0.3537
ViTs8	ImageNet	PolypGen SSL (100) + SL (300)	0.5741	<u>0.4960</u>	0.5772	<u>0.3775</u>
ViTs8	ImageNet	PolypGen SSL (200) + SL (300)	<u>0.5728</u>	0.4508	<u>0.5497</u>	0.4194
ViTs16	GastroNet	-	0.2938	0.3138	0.3908	0.2877
ViTs16	GastroNet	PolypGen SL (300 epochs)	0.5536	0.5273	0.5505	0.3345

Table A.3: Performance results per cluster in-distribution (ID) with two stages of clustering for the different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents the type of supervision training (Self-Supervised Learning) and 100 the number of epochs it was trained for. The best performance per cluster is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Global
ViTs8	ImageNet	-	0.3229	0.5651	0.0987	0.4236	0.1655	0.5572	0.1770	0.4667	0.3278
ViTs8	ImageNet	SSL100	0.4856	0.7174	0.1418	0.3789	0.3607	0.5814	0.2758	0.4667	0.3783
ViTs8	ImageNet	SSL200	0.4303	<u>0.7693</u>	<u>0.1470</u>	0.4663	0.2034	0.5218	0.2531	0.4667	0.3776
ViTs8	ImageNet	SSL100+SL300	0.6450	0.7009	0.2979	0.6357	<u>0.6305</u>	0.7471	<u>0.6013</u>	0.4667	0.5741
ViTs8	ImageNet	SSL200+SL300	<u>0.7114</u>	0.8991	0.0487	<u>0.6729</u>	0.6999	<u>0.7992</u>	0.6705	0.4667	<u>0.5728</u>
ViTs16	GastroNet	-	0.4635	0.1271	0.0709	0.2815	0.2191	0.6406	0.2048	0.4667	0.2938
ViTs16	GastroNet	SL300	0.7516	0.7592	0.0603	0.7597	0.5387	0.8235	0.5576	<u>0.0816</u>	0.5536

Table A.4: Performance results per cluster in-distribution (ID) with three stages of clustering for different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents self-supervised learning trained for 100 epochs. The best performance per cluster is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Global
ViTs8	ImageNet	-	0.4269	0.5766	0.1234	0.1977	0.1784	0.4923	0.2143	<u>0.4667</u>	0.2551
ViTs8	ImageNet	SSL100	0.4945	0.7332	0.1420	0.3955	0.2424	0.5494	0.2929	<u>0.4667</u>	0.3716
ViTs8	ImageNet	SSL200	0.4823	0.5650	<u>0.1499</u>	0.4934	0.1930	0.5586	0.3464	0.4758	0.3947
ViTs8	ImageNet	SSL100+SL300	0.5848	0.6676	0.2979	<u>0.6910</u>	<u>0.6289</u>	<u>0.6765</u>	0.6358	<u>0.4667</u>	0.5772
ViTs8	ImageNet	SSL200+SL300	<u>0.6332</u>	0.8535	0.2979	0.6092	0.7069	0.5613	0.4941	<u>0.4667</u>	0.5497
ViTs16	GastroNet	-	0.4264	0.1291	0.0813	0.6084	0.2085	0.6199	0.2642	<u>0.4667</u>	0.3908
ViTs16	GastroNet	SL300	0.7756	<u>0.7422</u>	0.0596	0.7615	0.5322	0.8246	<u>0.5229</u>	0.0468	<u>0.5505</u>

Table A.5: Performance results per cluster out-of-distribution (OOD) with two stages of clustering for different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents self-supervised learning trained for 100 epochs. The best performance per column is **bold**, and the second-best is underlined. In the out-of-distribution cluster assignment, some clusters do not have assigned samples. For these, the missing values are indicated by “-”.

Extractor	Pretrained on	Finetuned on	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Global
ViTs8	ImageNet	-	-	-	0.2556	-	0.2497	-	-	0.0972	0.2484
ViTs8	ImageNet	SSL100	-	-	0.3388	-	0.2897	-	-	0.1523	0.3062
ViTs8	ImageNet	SSL200	-	-	0.3923	-	0.3147	-	-	0.0706	0.3402
ViTs8	ImageNet	SSL100+SL300	-	-	<u>0.5836</u>	-	<u>0.4339</u>	-	-	0.5035	<u>0.4960</u>
ViTs8	ImageNet	SSL200+SL300	-	-	0.4911	-	0.4214	-	-	<u>0.4717</u>	0.4508
ViTs16	GastroNet	-	-	-	0.4958	-	0.1925	-	-	0.1438	0.3138
ViTs16	GastroNet	SL300	-	-	0.5923	-	0.5009	-	-	0.0647	0.5273

Table A.6: Performance results per cluster out-of-distribution (OOD) with three stages of clustering for different pretrained extractors and fine-tuning settings. "Finetuned on" column values were shortened due to the higher volume of columns for the clustering results. In SSL100, SSL represents self-supervised learning trained for 100 epochs. The best performance per column is **bold**, and the second-best is underlined. Missing values are indicated by “-”.

Extractor	Pretrained on	Finetuned on	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Global
ViTs8	ImageNet	-	-	-	0.2266	-	0.2600	-	-	0.0635	0.2418
ViTs8	ImageNet	SSL100	-	-	0.3656	-	0.2640	-	-	0.1526	0.3023
ViTs8	ImageNet	SSL200	-	-	0.3973	-	0.3340	-	-	0.0890	0.3537
ViTs8	ImageNet	SSL100+SL300	-	-	0.4923	-	0.2953	-	-	0.4058	<u>0.3775</u>
ViTs8	ImageNet	SSL200+SL300	-	-	<u>0.4554</u>	-	<u>0.3983</u>	-	-	<u>0.3159</u>	0.4194
ViTs16	GastroNet	-	-	-	0.4326	-	0.1912	-	-	0.1504	0.2877
ViTs16	GastroNet	SL300	-	-	0.1411	-	0.4829	-	-	0.0518	0.3345

A.1.2.2 Qualitative results

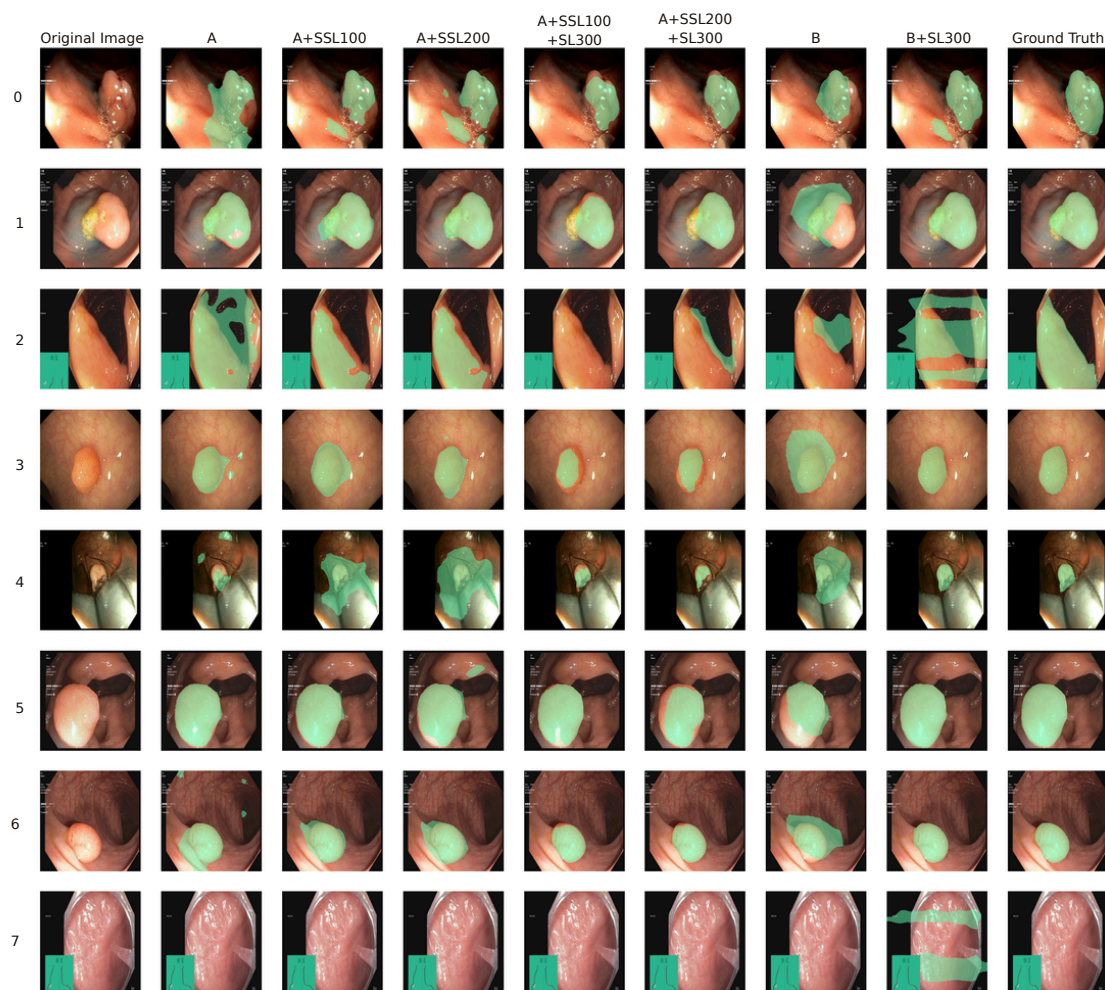


Figure A.1: Qualitative results for each cluster (rows) across different feature extractors (columns).

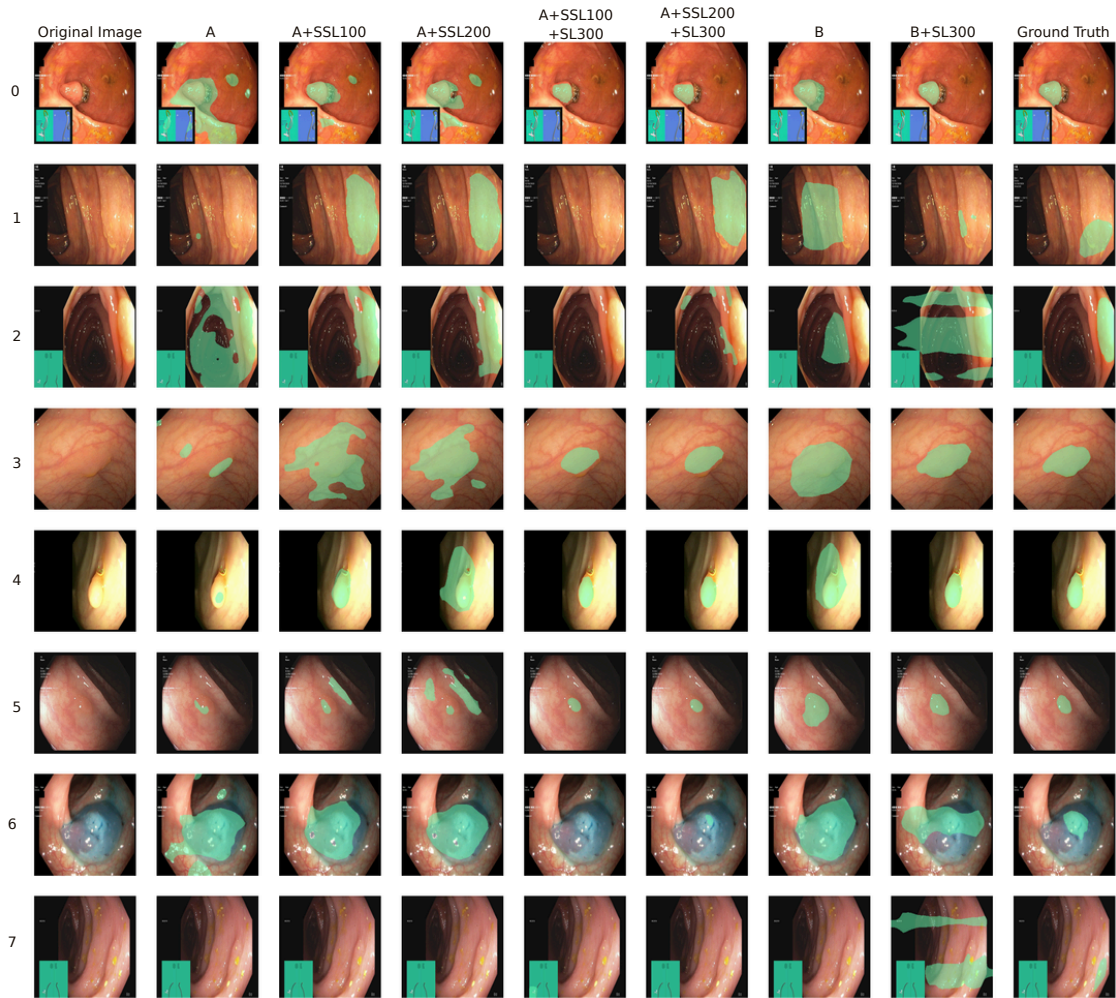


Figure A.2: Qualitative failure results for each cluster (rows) across different feature extractors (columns).

A.1.3 Classes

A.1.3.1 Quantitative results

Table A.7: Part Co-Segmentation quantitative results for class division (in-distribution and out-of-distribution). The best performance is marked in **bold**, and the second-best in underlined.

Extractor	Pretrained on	Finetuned on	Two Stages		Three Stages	
			Dice (ID)	Dice (OOD)	Dice (ID)	Dice (OOD)
ViTs8	ImageNet	-	0.4367	0.2701	0.4536	0.2744
ViTs8	ImageNet	PolypGen SSL (100 epochs)	0.4899	0.3484	0.4748	0.3558
ViTs8	ImageNet	PolypGen SSL (200 epochs)	0.4982	0.3431	<u>0.5251</u>	0.3561
ViTs8	ImageNet	PolypGen SSL (100) + SL (300)	<u>0.5815</u>	0.4752	0.5540	<u>0.4652</u>
ViTs8	ImageNet	PolypGen SSL (200) + SL (300)	0.5684	0.5184	0.4146	0.4321
ViTs16	GastroNet	-	0.4690	0.3206	0.4686	0.3217
ViTs16	GastroNet	PolypGen SL (300 epochs)	0.5953	<u>0.4784</u>	0.4984	0.4715

Table A.8: Performance results per class in-distribution (ID) with two stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Class 0	Class 1	Class 2	Class 3	Class 4	Global
ViTs8	ImageNet	-	<u>0.9667</u>	0.2549	0.0641	0.1523	0.5857	0.4367
ViTs8	ImageNet	PolypGen SSL (100 epochs)	1.0000	0.2891	0.1167	0.2146	0.6339	0.4899
ViTs8	ImageNet	PolypGen SSL (200 epochs)	1.0000	0.2464	0.0759	0.2015	0.6746	0.4982
ViTs8	ImageNet	PolypGen SSL (100 epochs) + SL (300 epochs)	0.1667	0.5770	<u>0.4392</u>	<u>0.6595</u>	0.6802	<u>0.5815</u>
ViTs8	ImageNet	PolypGen SSL (200 epochs) + SL (300 epochs)	0.0000	<u>0.5418</u>	0.5209	0.6195	<u>0.7044</u>	0.5684
ViTs16	GastroNet	-	1.0000	0.2896	0.0595	0.1435	0.6451	0.4690
ViTs16	GastroNet	PolypGen SL (300 epochs)	0.0000	0.4790	0.3705	0.6891	0.7641	0.5953

Table A.9: Performance results per class in-distribution (ID) with three stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Class 0	Class 1	Class 2	Class 3	Class 4	Global
ViTs8	ImageNet	-	<u>0.9667</u>	0.2969	0.0516	0.1527	0.6153	0.4536
ViTs8	ImageNet	PolypGen SSL (100 epochs)	0.9000	0.2777	0.0521	0.2353	0.6312	0.4748
ViTs8	ImageNet	PolypGen SSL (200 epochs)	1.0000	0.3240	0.0833	0.2288	<u>0.7084</u>	<u>0.5251</u>
ViTs8	ImageNet	PolypGen SSL (100 epochs) + SL (300 epochs)	0.1667	0.5454	0.4980	<u>0.6015</u>	0.6436	0.5540
ViTs8	ImageNet	PolypGen SSL (200 epochs) + SL (300 epochs)	0.0000	0.4211	0.3518	0.4117	0.5430	0.4146
ViTs16	GastroNet	-	1.0000	0.3309	0.0524	0.1355	0.6469	0.4686
ViTs16	GastroNet	PolypGen SL (300 epochs)	0.0000	<u>0.5069</u>	<u>0.3571</u>	0.6992	0.7652	0.4984

Table A.10: Performance results per class out-of-distribution (OOD) with two stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Class 0	Class 1	Class 2	Class 3	Class 4	Global
ViTs8	ImageNet	-	1.0000	0.2408	0.0518	0.1952	0.4424	0.2701
ViTs8	ImageNet	PolypGen SSL (100 epochs)	1.0000	0.2840	0.0676	0.1982	0.6197	0.3484
ViTs8	ImageNet	PolypGen SSL (200 epochs)	<u>0.5000</u>	0.1955	0.0621	0.2399	0.6334	0.3431
ViTs8	ImageNet	PolypGen SSL (100 epochs) + SL (300 epochs)	1.0000	<u>0.3673</u>	0.3336	0.6940	0.4660	0.4752
ViTs8	ImageNet	PolypGen SSL (200 epochs) + SL (300 epochs)	0.0000	0.3900	<u>0.3276</u>	<u>0.6144</u>	<u>0.6461</u>	0.5184
ViTs16	GastroNet	-	1.0000	0.1789	0.0471	0.1272	0.6430	0.3206
ViTs16	GastroNet	PolypGen SL (300 epochs)	0.0000	0.3341	0.0400	0.6026	0.7486	<u>0.4784</u>

Table A.11: Performance results per class out-of-distribution (OOD) with three stages of clustering for the different pretrained extractors and fine-tuning settings. The best performance per class is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Global
ViTs8	ImageNet	-	1.0000	0.2465	0.0593	0.2167	0.4342	0.2744
ViTs8	ImageNet	PolypGen SSL (100 epochs)	1.0000	0.2586	0.0687	0.2039	0.6454	0.3558
ViTs8	ImageNet	PolypGen SSL (200 epochs)	1.0000	0.2135	0.0660	0.2040	<u>0.6660</u>	0.3561
ViTs8	ImageNet	PolypGen SSL (100 epochs) + SL (300 epochs)	1.0000	0.4854	0.2859	0.4550	0.5606	<u>0.4652</u>
ViTs8	ImageNet	PolypGen SSL (200 epochs) + SL (300 epochs)	0.0000	<u>0.4054</u>	<u>0.2766</u>	<u>0.4746</u>	0.5257	0.4321
ViTs16	GastroNet	-	1.0000	0.1852	0.0445	0.1316	0.6423	0.3217
ViTs16	GastroNet	PolypGen SL (300 epochs)	0.0000	0.3336	0.0157	0.6009	0.7430	0.4715

A.1.3.2 Qualitative results

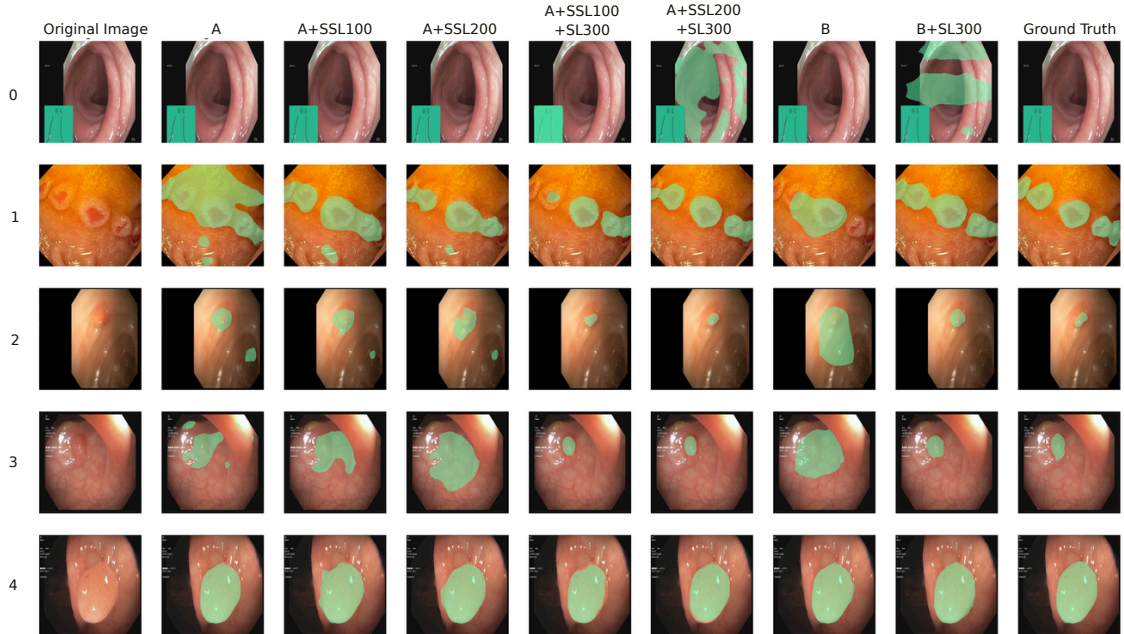


Figure A.3: Qualitative results for each class (rows) across different feature extractors (columns).

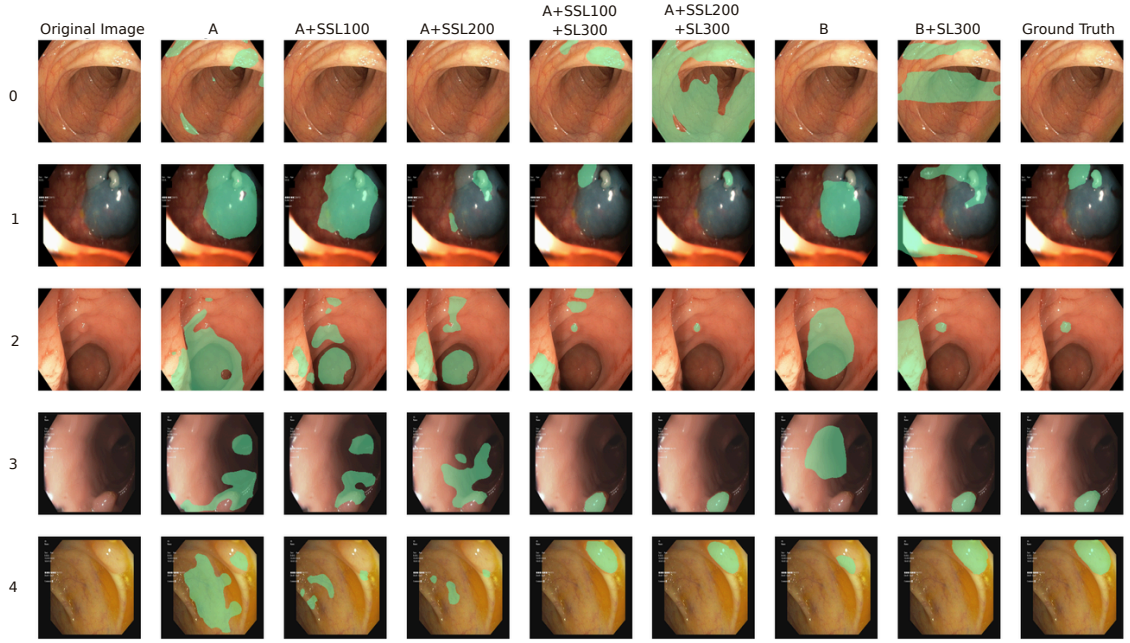


Figure A.4: Qualitative failure results for each class (rows) across different feature extractors (columns).

A.1.4 Sisters

Table A.12: Performance results for the paired sister images approach with $N = 1$. The best performance is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Two Stages		Three Stages	
			Dice (ID)	Dice (OOD)	Dice (ID)	Dice (OOD)
DINO ViTs8	ImageNet	-	0.4821	0.3425	0.3443	0.3350
DINO ViTs8	ImageNet	PolypGen SSL (100 epochs)	0.5132	0.3697	0.4468	0.4062
DINO ViTs8	ImageNet	PolypGen SSL (200 epochs)	<u>0.5283</u>	<u>0.3932</u>	0.5064	<u>0.3886</u>
DINO ViTs8	ImageNet	PolypGen SSL (100) + SL (300)	0.4875	0.2726	0.5985	0.3483
DINO ViTs8	ImageNet	PolypGen SSL (200) + SL (300)	0.4209	0.2883	<u>0.5575</u>	0.3514
DINO ViTs16	GastroNet	-	0.4787	0.3555	0.4714	0.3501
DINO ViTs16	GastroNet	PolypGen SL (300 epochs)	0.5390	0.4396	0.5051	0.3442

Table A.13: Performance results for the multiple sisters approach considering 6 sisters. The best performance is **bold**, and the second-best is underlined.

Extractor	Pretrained on	Finetuned on	Two Stages		Three Stages	
			Dice (ID)	Dice (OOD)	Dice (ID)	Dice (OOD)
DINO ViTs8	ImageNet	-	0.5253	0.3889	0.3510	0.3414
DINO ViTs8	ImageNet	PolypGen SSL (100 epochs)	<u>0.5628</u>	0.4147	0.4665	<u>0.4177</u>
DINO ViTs8	ImageNet	PolypGen SSL (200 epochs)	0.5532	<u>0.4199</u>	0.4861	0.4360
DINO ViTs8	ImageNet	PolypGen SSL (100) + SL (300)	0.4962	0.2965	0.5881	0.3348
DINO ViTs8	ImageNet	PolypGen SSL (200) + SL (300)	0.4899	0.3367	<u>0.5657</u>	0.3344
DINO ViTs16	GastroNet	-	0.4958	0.3875	0.4780	0.3760
DINO ViTs16	GastroNet	PolypGen SL (300 epochs)	0.6115	0.4512	0.5038	0.2724