UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

INFORMATION ENGINEERING DEPARTMENT

MASTER'S DEGREE IN COMPUTER ENGINEERING

An Empirical Study on Segmentation Methods with Deep Ensembles and Data Augmentation

Supervisor:
Prof. Loris Nanni

Student:
Daniela Cuza

ACCADEMIC YEAR: 2022/2023

Graduation Day: 07/09/2023

The value of an idea lies in the using of it.

-Thomas A. Edison

I have not failed, I've just found 10,000 ways that won't work.

-Thomas A. Edison

Abstract

In the past few years, there has been a growing focus on semantic segmentation, which involves assigning each pixel in an image to a specific label from a given set[69].The use of autoencoder architectures has been explored by numerous computer vision researchers in an attempt to develop models capable of learning both the semantics of an image and a low-level representation of it. When utilizing an autoencoder architecture, the input undergoes encoding to produce a low-dimensional representation. This representation is subsequently leveraged by a decoder to reconstruct the original data. The presented approach involves a combination of convolutional neural networks (CNNs) and transformers to form an ensemble, as detailed in this work. Ensemble methods rely on multiple models being trained and utilized for classification, with the ensemble combining the outputs of individual classifiers. By capitalizing on the varying strengths of each classifier, this approach enhances the overall performance of the system. Distinct loss functions are employed to ensure diversity among the individual networks. The ensemble method employs a combination of the DeepLabV3+, HarDNet, and PVT environments, with varying backbone networks. Additionally, a novel loss function is presented, which integrates the Dice and Structural Similarity Index. To assess the proposed ensemble, a comprehensive empirical evaluation is conducted on six real-world scenarios, namely polyp, skin segmentation, leukocyte segmentation, butterfly identification, microorganism identification, and radiology segmentation. The proposed model has achieved state-of-the-art performance on these scenarios.

Contents

1 Introduction **1**

2 Related Work **7**

2.1 Skin Segmentation 8

2.2 Determining the Optimal Sample Size for Segmentation in Medical Imaging . . 10

3 Methods **13**

3.1 Topologies 13

3.1.1 DeepLabV3+ 13

3.1.2 HardNet 14

3.1.3 PVT 14

3.2 Loss Function 15

3.2.1 Generalized Dice Loss 15

3.2.2 Tversky Loss 16

3.2.3 Focal Tversky Loss 16

3.2.4 Focal Generalized Dice Loss 17

3.2.5 Log-Cosh Type Losses 17

3.2.6 Neighbor Loss 17

3.2.7 SSIM Loss 18

3.2.8 Different Functions Combined Loss 18

3.2.9 Cross Entropy 19

3.2.10 Weighted intersection over union 19

3.2.11 Structure Loss 20

3.2.12 BoundExpStructure 21

3.2.13 Boundary Enhancement Loss 21

3.2.14 Contour-aware Loss 22

3.3 Data Augmentation 22

3.3.1 DA1 23

3.3.2 DA2 25

3.3.3 Contrast and Motion Blur 25

3.3.4 Shadows . . . 27
3.3.5 Color Mapping . . . 28
3.3.6 JPEG approach . . . 28
3.3.7 Alternating Vertical Shift . . . 29
3.3.8 Alternating Horizontal Shift . . . 29
3.3.9 Alternating Diagonal Shift . . . 30
3.3.10 Random Shift with Black or Wrap . . . 30
3.3.11 Random Shift Up Down with Black or Wrap . . . 30
3.3.12 Random Rectangles Mix and Blackout . . . 30
3.3.13 Random Rectangle Rotation . . . 30
3.3.14 Random Rectangle Flip . . . 30
3.3.15 Random Rectangle Brightness . . . 31

4 Results **33**
4.1 Metrics . . . 33
4.2 Datasets . . . 34
4.2.1 Skin segmentation (SKIN) . . . 34
4.2.2 Segmentation in Radiology: VinDr-RibCXR . . . 36
4.2.3 Polyp segmentation (POLYP) . . . 37
Kvasir . . . 38
COLON-DB . . . 38
CVC-T . . . 38
ETIS . . . 39
CVC-ClinicDB . . . 39
4.2.4 Leukocyte segmentation (LEUKO) . . . 40
4.2.5 Butterfly identification (BFLY) . . . 41
4.2.6 Microorganism identification (EMICRO) . . . 41
4.3 Experiments . . . 41
4.3.1 Skin Segmentation . . . 41
4.3.2 Radiology Segmentation . . . 43
4.3.3 Other Segmentation Applications . . . 47

5 Conclusion **55**

Bibliography **61**

Chapter 1

Introduction

Medical imaging is an essential component of modern healthcare, allowing clinicians to accurately diagnose, treat, and monitor diseases, with applications ranging from colorectal cancer detection to brain tumor segmentation and prostate cancer diagnosis. However, the effectiveness of these procedures relies heavily on the accuracy of image analysis, making the need for automated, highly accurate systems in medical image analysis ever more critical.

Colorectal cancer (CRC), the third most common and deadly cancer in the United States, is often linked to adenomatous polyps. Although colonoscopy is the standard method for detecting and removing these polyps, the accuracy of this process is hindered by several factors such as the physician's expertise and the characteristics and location of the polyp. As a result, research has indicated that colonoscopies can miss 6% to 28% of colorectal polyps [23], leading to interval CRCs, which make up 5% to 8% of all CRC cases. To address this issue, deep-learning-assisted diagnostic tools have been developed to identify polyps in colonoscopy videos, thus improving the quality of colonoscopy screenings [11][105][66][104].

In the realm of brain tumor segmentation, precision is paramount for diagnosis, treatment planning, and monitoring treatment response. The Brain Tumor Segmentation Challenge (BraTS) has encouraged the development and comparison of models for this purpose, with recent high-performing models employing deep neural networks and encoder-decoder architectures[48].

Prostate cancer, the second most common cancer among men globally, also stands to benefit from advancements in machine learning techniques. Current diagnostic methods such as prostate-specific antigen (PSA) blood tests and transrectal ultrasound (TRUS) biopsies present limited efficacy due to their low specificity and sensitivity. However, encoder-decoder CNNs show promise in enhancing prostate cancer diagnosis, staging, and treatment management by effectively segmenting prostate T2W MRI images[54].

One of the most significant challenges in developing these automated systems is the lack of large, accurately annotated datasets for training. This is particularly challenging in the medical field, where data collection and annotation can be costly and time-consuming. Furthermore,

without adequate data, models risk overfitting, meaning they become too specialized to their training data and underperform on new, unseen instances. To address this, many strategies have been developed. One of the most promising is data augmentation. In this thesis, simple and traditional data augmentation techniques such as flipping, rotating, cropping, and color transformations like the separation of RGB color components and the introduction of noise have been explored in detail. These methods can significantly improve the size and variety of datasets, which in turn can enhance the resilience of deep learning models.

While the focus of this thesis is on simpler methods, it is worth noting that there are more advanced techniques in the realm of data augmentation that present substantial potential. Examples include the utilization of Generative Adversarial Networks (GANs) which create synthetic instances from a dataset, maintaining attributes similar to the original data. Additionally, there are meta-learning approaches like Neural Augmentation and Feature Transform that push the boundaries of conventional augmentation. While these complex methods are beyond the scope of this work, they represent a vibrant and evolving field of study in deep learning. Harnessing the potential of advanced data manipulation and augmentation can lead to breakthroughs in various fields. This brings us to the exploration of innovative models that show promise outside the realm of medical imaging but could hold potential within it. The Segment Anything (SA) project [57], designed for natural image segmentation, is one such model that has demonstrated promising results. While its effectiveness for medical imaging is not fully confirmed, the SA model could potentially be utilized as a post-processing tool to refine the outputs of medical image segmentation models. The project's principle of a "data engine" that continually collects and refines data could also inspire similar strategies in the medical field. In this way, the innovative methodologies of the SA project may contribute to advancements in medical image segmentation research. However, the applicability of the SA model and similar models to medical imaging warrants further investigation to ensure their effectiveness and accuracy in this critical context.

To tackle the issues of limited and weak annotations common in medical image segmentation datasets, techniques such as gamification [102] have been introduced as a human-in-the-loop strategy. This innovative approach transforms the annotation task into a game, engaging users through the thrill of competition. Such human-in-the-loop strategies can indeed be pivotal, especially when exact annotations are hard to come by.

Yet, as annotations become sparser or less precise, the importance of algorithmic adaptations rises. In this vein, there's a burgeoning interest in weakly supervised learning. For instance, Graph Convolutional Networks (GCNs) [40] have been making strides, achieving state-of-the-art performance in Weakly Supervised Semantic Segmentation (WSS). Notably, the HyperGraph Convolutional Networks for Weakly Supervised Semantic Segmentation (HyperGCN-WSS) stands out, capturing intricate spatial and structural nuances from instances within the dataset.

Within the scope of this thesis, while our main approach has been rooted in supervised

techniques, the landscape of weakly supervised methods warrants discussion. In fact, these techniques aim to produce semantic segmentation masks for medical images with minimal guidance, often capitalizing on image-level annotations. A prevalent strategy in this domain involves the use of class activation maps (CAMs) for generating pixel-level masks. However, CAMs can sometimes miss the mark, either omitting vital parts of the object or inadvertently including irrelevant background. To address these imperfections, the BoundaryFit module [78] was introduced, bridging the gap between the preliminary CAM prediction and the subsequent mask refinement stage, enhancing object boundaries for a more accurate segmentation mask.

Further innovation in this space can be seen with the advent of the weakly supervised segmentation method, TransWS [116]. This method, grounded in Transformers and end-to-end learning, skillfully utilizes image-level labels for the classification branch, treating the CAM rendered by the classifier as pixel-level pseudo-labels for the segmentation branch. By merging the insights from both the segmentation and classification branches, TransWS achieves a more nuanced and precise segmentation outcome.

For a considerable period, identifying objects in images was exclusive to humans [67]. It took more than 14 years to match an untrained human's performance in the ImageNet competition. The complexity increases when the job involves not only recognizing the object in an image but also determining its borders. This is referred to as semantic segmentation, and it involves categorizing every pixel in an image in machine learning. The performance enhancements associated with implementing machine learning models have made this task applicable to numerous real-life situations [36][11]. For example, in clinical settings, it can aid in detecting polyps, and in skin and blood analysis, object identification can assist in visually identifying the existence of various ailments. Furthermore, this task is utilized in autonomous vehicles to recognize objects in the vehicle's vicinity, in the classification of environmental microorganisms, and numerous other applications. The conventional method is to develop a system consisting of two modules: an encoder and a decoder. The first module is trained to capture the semantic features of the input image and create a low-dimensional representation of it. The second module is then trained to reconstruct the original input image from this compressed feature vector. U-Net [74] was one of the earliest systems developed for semantic segmentation, and it utilized the aforementioned approach. Autoencoders [4][13][61] were also implemented to perform this task because they can acquire semantic low-level representations of an image using the encoder module and reconstruct the original input from the compressed representation. The excellent results achieved by autoencoders have led many researchers and practitioners in the field of computer vision to adopt them. However, the performance of autoencoders, along with other classification technologies, is heavily influenced by architectural configuration and other settings, commonly known as hyperparameters, which require tuning.

Hyperparameter tuning involves determining the optimal values for specific attributes of the model. This is a domain-specific task that necessitates knowledge of the field as well as

proficiency with the applied machine learning methods, resulting in significant effort and time consumption. In light of this, optimizing hyperparameters emerges as a pivotal step for achieving peak performance in machine learning models, especially within the realm of CNN architectures. Numerous techniques have been devised for this intricate task of hyperparameter tuning. Among these are Grid Search (GS), Random Search (RS), Bayesian Optimization (BO), Nelder Mead (NM), Simulated Annealing (SA), Particle Swarm Optimization, and Evolutionary Algorithms. Yet, with the marked surge in hyperparameters intrinsic to modern CNN architectures, this optimization venture is becoming ever more convoluted. Notably, some methods have yet to encompass every hyperparameter integral to CNN design. On the forefront of advancements in CNN hyperparameter optimization, we find techniques like Sequential Model Based Optimization (SMBO), Gaussian Process based Bayesian Optimization, and evolutionary approaches. In scenarios demanding high-dimensional hyperparameter optimization, tree-based models such as Tree structured Parzen Estimators (TPE) and Random Forests have showcased commendable outcomes. At its core, irrespective of the technique harnessed, hyperparameter optimization stands paramount for extracting the zenith of performance from a machine learning model.

The "no-free lunch" theorem in machine learning states that there cannot be a single model that performs optimally on all datasets. Given this, another approach involves employing sets of classifiers, often weak or shallow, and combining their predictions to form the system's output. These frameworks are known as ensemble methods. Ensemble methods involve training multiple classifiers on the same dataset in a manner that ensures each model generalizes differently in the training space. While ensembles can yield state-of-the-art outcomes in numerous domains, it is critical to ensure certain properties, such as enforcing diversity among the set of classifiers.

Integral to the performance of these classifiers, beyond their architecture, is the choice of their guiding metric: the loss function. The right loss or objective function can significantly elevate a model's performance. Literature broadly categorizes loss functions into four groups: Distribution-based, Region-based, Boundary-based, and Compounded. Distribution-based losses, as exemplified by Binary Cross Entropy, arise from the distribution of labels. Region-based ones aim at maximizing the overlap between predictions and ground truths. In contrast, Boundary-based losses focus on minimizing the distance between these two. Lastly, compounded functions combine characteristics of various losses.

Throughout the course of my research, I've delved into creating a novel loss function tailored for semantic segmentation, particularly in the medical imaging arena. A recurrent challenge in this space is the class imbalance, with the positive class often underrepresented. Our proposed loss function aspires to tackle this by incorporating strategies that counteract the effects of such imbalance.

Returning to ensemble methods, this thesis presents a new ensemble technique for semantic segmentation based on convolutional neural networks (CNNs) and transformers. Here, the diversity among the individual classifiers is enforced by utilizing distinct loss functions

and implementing various forms of data augmentation. Our approach merges DeepLabV3+, HarDNet-MSEG, and Pyramid Vision Transformers models. We evaluated our model on six distinct scenarios, including polyp detection, skin detection, leukocyte recognition, environmental microorganism detection, butterfly recognition, and radiology segmentation. After developing the proposed solution, we conducted a comprehensive empirical evaluation, which compared our approach to state-of-the-art solutions. Our assessment revealed promising results that were frequently superior to the best available methods.

A new architecture has emerged from the realm of natural language processing (NLP), where researchers explore ways to grasp the meaning of text and automate tasks such as summarization or translation. This novel model, known as Transformer, employs a self-attention mechanism, enabling the system to concentrate on specific parts of the input. Transformers have also found use in computer vision tasks and often achieve comparable or even superior performance to CNNs. However, as with other machine learning models, their primary limitation lies in the requirement for vast amounts of data to train a stable and high-performing system. Two recent medical domain approaches, TransFuse [117] and UACANet [55], employ different techniques. The former integrates CNN kernels and Transformers, while the latter blends U-Net and a parallel axial attention autoencoder. Regardless of the architecture, the objective is to acquire information at both local and global levels.

As noted earlier, semantic segmentation plays a crucial role in numerous contexts. For example, autonomous vehicles employ semantic segmentation to recognize objects in the vehicle's vicinity and make safe decisions accordingly. Deep learning techniques are also widely used in skin detection, ranging from face detection to hand gesture recognition. However, deep learning approaches have encountered certain challenges in this domain, such as background clutter that impedes the accurate detection of hand gestures in real-world settings. CNNs have also demonstrated their efficacy in this area, as demonstrated by the works of Roy et al. [104] and Arsalan et al. [99]. In the former, the authors propose using a skin detection-based CNN to improve the hand detector's output. In contrast, the latter work introduced a CNN with residual skip connections, OR-Skip-Net, which reduces the computational burden of the network while handling challenging skin segmentation tasks. This is reached by directly transferring data from the initial layer to the last layer of the network. CNNs are also used for automatic sign language translation [42]. In [49], a comparative analysis of multiple leading technologies on a variety of skin detection benchmarks is presented via a comprehensive empirical evaluation.

Leukocyte recognition and classification can be automated using deep learning, aiding medical practitioners in diagnosing blood-related diseases. The analysis can be performed through histogram-based techniques or iterative algorithms like GrabCut, which can segment white blood cells. This technique has been utilized in recent studies [61][33].

The aim of this study is to address semantic segmentation by introducing a new ensemble method that uses DeepLabV3+, HarDNet-MSEG, and Pyramid Vision Transformers backbones.

To promote diversity among individual classifiers, various loss functions and data augmentation approaches are adopted. The proposed approach is evaluated on six different scenarios, and the results are compared with existing frameworks. The empirical evaluation indicates that the proposed method produces results that are comparable or superior to state-of-the-art levels.

The remainder of this thesis is organized as follows. In Section 2, we present an in-depth overview of the prior studies and research carried out in the field of image segmentation, with a specific focus on skin segmentation and the optimal sample size for medical imaging segmentation. In Section 3, we elaborate on the techniques used in this research, such as the topologies, loss functions, and data augmentation methods. The results in Section 4 demonstrate that our best ensemble approach outperforms other methods. The separate sections for skin segmentation and radiology segmentation in this thesis are included to highlight the distinct characteristics and challenges in each area, and to allow for a more in-depth examination of each. Lastly, in Section 5, we present our conclusions and suggestions for future work.

Chapter 2

Related Work

The primary goal of semantic segmentation is to recognize objects within an image and delineate their boundaries [71]. Its significance extends to several practical applications, including medical diagnosis [12] and autonomous vehicles [37]. As previously stated, this method assigns a class label to each object at the pixel level in an image. For deep semantic segmentation, the Fully Convolutional Network (FCNs) is an early Deep CNN (DCNN) used to replace the last fully connected layer of a CNN architecture with a fully convolutional layer. By doing so, the network is capable of generating pixel-level predictions and solving the issue of semantic segmentation [62]. As the proposed method in this study is based on DCCN, the succeeding discussion will concentrate on DCCN semantic segmentation approaches. For image segmentation using other deep learning models, such as recurrent neural networks and attention and generative models, the reader is directed to [64]. The incorporation of an autoencoder unit in FCN enables the creation and training of deconvolutional networks. An autoencoder unit is comprised of an encoder network, usually a pre-trained CNN like VGG or ResNet, followed by a decoder network. The encoder's objective is to extract features that will generate a latent image, while the decoder is responsible for reconstructing the image. U-Net is a widely used autoencoder for semantic segmentation [84]. The autoencoder in U-Net reduces the image's dimensions while also enlarging the input feature size and resolution to enable segmentation. SegNet is another frequently used autoencoder for semantic segmentation [5]. VGG serves as the encoder network, and unlike other networks, the input of the decoder in SegNet is not the anticipated output of the encoder. Instead, the max pool indices of the corresponding encoder layer are fed to each decoder layer in SegNet. This architecture enables SegNet to consume less memory and perform better in segmentation tasks. [93]. Several other deep segmentation methods use the transformer [53], a deep learning approach initially created for text comprehension and summarization. Remarkably, the transformer's structure appears to imitate the human brain's vision process, making it simple to extend this segmentation method to computer vision. The transformer builds on autoencoder units but incorporates a self-attention mechanism that analyzes the input information in great detail while simultaneously processing the rest of the

information. According to [64], the training process involves two steps:

1. Training the model on a large dataset to set the weights in a way that enables the model to generalize better to a more extensive solution space.

2. Fine-tuning the model on a smaller dataset to improve its performance on the specific task at hand.

Because the complexity of the attention operator is quadratic, some reduction of the input size is necessary, which is accomplished by initially dividing the image into patches [34], a prevalent technique in computer vision. The image is then subjected to linear transformation and position embeddings, producing the input to the transformer encoder. An instance of utilizing this segmentation method in the medical field is TransFuse [118], which merged the ability of CNN kernels to capture local information with the transformer to represent information at a more advanced level. UACANet is an alternative method that accounts for both local and global information levels. It involves using U-Net and a parallel axial attention encoder and decoder [56]. Google has developed a successful line of evolving networks called DeepLab[20], which is widely used for semantic segmentation. DeepLab uses atrous convolution to increase the filter window size and maintain computational efficiency, by upsampling the output of the last convolution layer using a dilation rate. DeepLabV3 enhances DeepLab by: 1) using a combination of cascade and parallel units for convolutional dilation, and 2) incorporating batch normalization and 1x1 convolutions in Atrous Spatial Pyramid Pooling. DeepLabV3+ [22] further improves upon DeepLabV3 by introducing a decoder that employs point-wise convolutions to operate on the same channel but distinct locations, as well as depth-wise convolutions that process on the same location but distinct channels. HarDNet-MSEG [45], which was designed for polyp segmentation, is another example of a DCCN. HarDNet uses HarDNet68 [16], a CNN with an encoder-decoder architecture that has demonstrated success in various computer vision tasks, as its backbone. HarDNet's decoder architecture was inspired by the Cascaded Partial Decoder [111], which is recognized for its efficiency in precisely identifying prominent objects. Prior to feeding the encoder's output to the decoder, a Receptive Field Block [60] enhances the features by incorporating various receptive fields.

2.1 Skin Segmentation

The upcoming paragraphs will introduce the latest developments in skin segmentation. Notably, a deep learning architecture has been put forward to address the challenges associated with low-resolution grayscale images, particularly those captured using the SPAD array camera [75]. Moreover, the network is designed specifically for facial skin segmentation. The proposed colorization network from [6] is adapted slightly to suit the specific application, and then fine-tuning is applied. To tackle the challenge of facial skin segmentation, a dedicated dataset [75]

is presented, which is created by merging two pre-existing datasets, MUCT and Helen. The resulting dataset comprises 6000 grayscale facial images, each accompanied by a corresponding skin labeling mask. For an extensive examination of skin cancer detection, the reader can consult a recent survey on the use of Deep Learning Techniques for skin cancer detection [30].

Recent research has highlighted the continued significance of Convolutional Neural Networks (CNNs) in the realm of skin detection. Two recent papers, OR-Skip-Net [2] and a novel skin detection CNN model [86], provide examples of this. The former is a fully convolutional network with outer residual paths that extend from the encoder to the decoder. The latter model features three convolution layers, a down-sampling layer, a flatten layer, and fully connected layers. The Skinny network [98] builds upon the U-Net architecture and offers distinct benefits over certain architectural components, such as inception modules and dense blocks, by effectively incorporating both local and global pixel descriptions. A new approach for addressing skin detection issues is presented in [29], which employs a zero-sum game theory model. In this model, the classification problem is viewed as a competition between two players, namely skin and non-skin pixels. To apply this approach, the image is divided into small patches or regions, and each region is assessed and classified utilizing a group of classifiers. If all classifiers are in agreement, the patch is classified accordingly, but if there is a difference in prediction, the patch is deemed conflicting. The classifiers used include those based on color space thresholding and artificial neural networks. This technique has the potential to minimize the identification of non-skin regions as skin.

One of the earliest techniques proposed for detecting skin among many others is rule-based, including methods based on thresholding. The concept behind color space thresholding is to identify pixels of a specific color, which in this case is skin color. In [65], the skin color thresholds for two of the most widely used color models, HSV and YCbCr, are presented. The HSV color model is comprised of three elements: hue, saturation, and value, which represent the cylindrical coordinates of an RGB color model. On the other hand, YCbCr is composed of three components: luma, chrominance blue, and chrominance red, that can be trivially computed from RGB values.

In [100], a new dataset of abdominal skin images generated from Google images is introduced. The dataset comprises 1400 images that have been manually segmented to depict the abdomen of individuals from diverse ethnic groups. Specifically, the dataset includes 700 images depicting people with dark skin and another 700 images depicting people with light skin. Additionally, some of the images portray individuals with higher body mass indices, and others display individuals with tattoos.

2.2 Determining the Optimal Sample Size for Segmentation in Medical Imaging

As stated in Chapter 7 of [41], deep learning models are frequently utilized in complex fields, such as image processing, which is the primary focus of the thesis. These models are used to simulate the entire universe, as the true generation process in these fields essentially requires it. The authors suggest that in reality, managing complexity in deep learning is not a straightforward task of determining the appropriate size of the function space $\mathcal{F}$. This is because the correct size of the function space may be as vast as the entire universe. It is worth noting that there exists a significant gap between the learning problems and models studied by statistical learning theory or other theoretical results (such as classification using support vector machines), and the problems and models currently employed in deep learning (such as semantic segmentation of images using DeepLabV3+). In the empirical evaluation of deep learning models, the optimal classifier is obtained by utilizing a large function space with the right amount of regularization. This helps to minimize the generalization error without affecting the training error.

According to [41, Chapter 11], if the performance on the training data is unsatisfactory, collecting more data is not an effective solution. Instead, it is more beneficial to either use a more complex model or adjust the hyperparameters values to improve the result. Even if a larger model and a better learning algorithm are employed, the performance on the training data may still be poor. In such a scenario, it is advisable to assess the quality of the training data, specifically the presence of noise and accurate input, and gather new, clean data with a broader set of features.

Another scenario occurs when the performance on the training data is good, but the result on the test set is poor. In this situation, it is crucial to gather more data. However, in medical applications, obtaining more data can be a challenging task as it can be difficult to find medical experts, and there may also be concerns related to patient and hospital privacy. In such cases, alternative options include reducing the complexity of the model and increasing regularization.

The relationship between the size of the training set and the generalization error curve can be useful in determining the amount of data required to achieve a desired performance in a specific application.

The aforementioned guidelines are aimed at practitioners, who use machine learning as a mere tool.

According to Widrow's rule of thumb for multivariate analysis, it is recommended that the sample size, n, should be at least ten times larger than the number of parameters that need to be estimated ([108]). However, this guideline may not always provide the optimal sample size and may result in either an underestimation or overestimation, depending on the circumstances ([15].

The aim of sample-size determination methodologies (SSDMs) in machine learning, espe-

cially in medical imaging, is to determine the appropriate number of images needed to achieve a desired performance for an algorithm ([7]). Essentially, there are two types of SSDMs: model-based and curve-fitting. Curve-fitting SSDMs rely on empirical evaluation to determine the performance of an algorithm with respect to the size of the training sample. These approaches can further be divided into two categories: learning curve-fitting and linear curve-fitting. The goal of learning curve-fitting methods is to model the correlation between the size of the training set and the classification accuracy. Typically, the curve obtained through curve-fitting SSDMs follows an inverse power law function ([38]). Using the learning curve approach, [8] discovered that a sample size of 75-100 is sufficient to evaluate a classifier that is not perfect but performs well in biospectroscopy.
The required number of images per class to achieve an overall classification accuracy (OCA) of 82% was determined to be 10000 images by Rokem et al. (2017)[83]. Conversely, Cho et al. (2015) [25] calculated that 4092 images per class would result in an OCA of 99.5%. The differences in the results of these two studies can be attributed to the variations in the subsampling method, machine learning model, and application used.
The linear curve-fitting approach examines the relationship between the area under the receiver operating characteristic curve and the inverse of the training data size.

Shao et al. (2013) [90] introduced the SSNR-based protocol, a straightforward and effective method for determining the minimum training sample size in gene expression microarray technology. This protocol not only provides an estimate of the minimum sample size but also predicts the performance of classifiers with minimal prior information.

Narayana et al. (2020) [72] conducted a thorough investigation into the effect of training size on segmentation accuracy in multiple sclerosis. The authors employed a learning curve approach, plotting the accuracy against the size of the training set and using an inverse power law. The following equation, as introduced by [72], shows the relationship between accuracy (Y), training set size (X), and the parameters of the learning curve (a, b_1, b_2):

$$Y = (1 - a) - b_1 X^{b_2} \tag{2.1}$$

The results of the study indicate an improvement in performance as the training size increases. Specifically, with a training size of 50 or more, the results for lesion segmentation are exceptional. Conversely, a training size of 10 is sufficient to achieve good results for gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF).

According to Willemink et al. (2020)[109], having a large training set size alone is not enough to achieve optimal results. It is also important to properly curate, investigate, categorize, and apply the image data in a clinical context. Furthermore, Fang et al. (2021[35] show that the benefits of increasing the training size progressively become less significant.

In the study by Fang et al. (2021)[35], the U-Net network was utilized to examine fourteen

regions of interest, including the brainstem, spinal cord, eyes, lenses, optic nerves, temporal lobes, parotids, larynx, and body. The results indicated that for six of the organs, the optimal outcome was achieved using 800 training images, while a smaller training size of 600 or 400 was needed for the remaining organs.

In another study focused on the impact of training sample size, Wulms et al. (2022) [112] emphasized the significance of having a sufficient training size for accurate prediction of white matter hyperintensity volume. Their tests revealed that as the training size increases, the accuracy of the predictions improves. Several methods have been introduced in the literature for enhancing classification performance without expanding the sample size. Some of these methods include:

- ensemble techniques;
- data augmentation techniques;
- weakly supervised learning and, more recently, few-shot/zero-shot learning;
- domain adaptation techniques, particularly transfer learning for deep neural networks;
- regularization techniques.

Chapter 3

Methods

3.1 Topologies

3.1.1 DeepLabV3+

In this study, we delve into the DeepLabV3+ model. DeepLab is composed of a series of autoencoder models, as described by Chen et al. (2018)([19]), and has demonstrated remarkable success in multiple fields of application, as reported in Zheng et al. (2021)([121]). The following are some of the key features that contribute to DeepLab's success:

- Improved resolution is achieved through the use of dilated convolutions, reducing the impact of pooling and stride.
- The Atrous Spatial Pyramid Pooling technique allows for data to be gathered at multiple scales.
- The combination of CNNs and probabilistic graphic models results in accurate object boundary identification.

DeepLabV3 introduces two key improvements: the use of a 1×1 convolution in Atrous Spatial Pyramid Pooling and the inclusion of batch normalization, as well as a combination of cascaded and parallel convolutional dilation modules. In this work, we employ the DeepLabV3+ model([21]), an improvement on the series of models proposed by Google. The key innovation in this network is its decoder, which includes both depth-wise and point-wise convolutions. The depth-wise convolutions work in the same location but across distinct channels, while the point-wise convolutions operate on the same channel across distinct locations. In order to develop a diverse set of models within a framework, we can focus on different aspects of the network architecture. Each architecture model has its own distinct features.

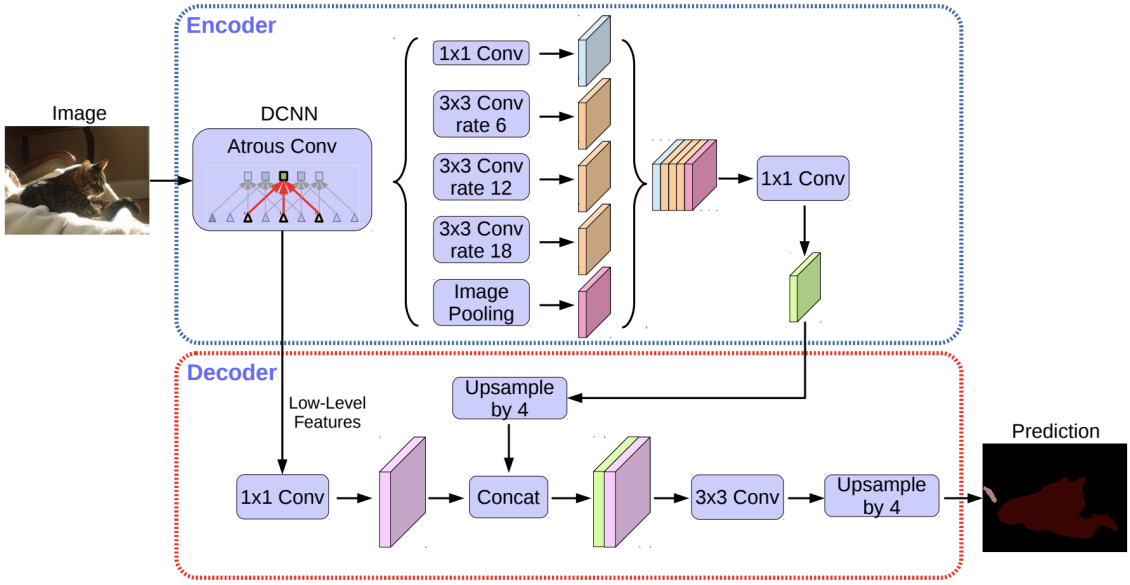

Figure 3.1: DeepLabv3+ [21] employs an encoder-decoder structure to enhance the capabilities of DeepLabv3. The encoder module uses atrous convolution at multiple scales to encode multi-scale contextual information, while the decoder module, which is simple yet effective, refines the segmentation results along the boundaries of objects.

3.1.2 HardNet

HarD-Net(Harmonic Densely Connected Net-work) , as described in the paper by Chao et al. [17] [1], is a model that takes inspiration from Densely Connected Networks. One of the advantages of HarD-Net is its efficient use of memory, achieved by reducing the number of connection layers in comparison to DenseNet, leading to lower concatenation costs. Moreover, the input-to-output channel ratio is balanced due to the increase in the channel width of the layers, resulting in an increase in its connections.

3.1.3 PVT

The Pyramid Vision Transformer (PVT) [31] [2] is a transformer network that does not use any convolutional layers. Its goal is to obtain a high-resolution representation starting from a detailed input. The model's computational cost is reduced through a progressive shrinkage in the form of a pyramid, along with the depth of the model. A spatial-reduction attention (SRA) layer has been added to further reduce the complexity of the system.

[1] https://github.com/james128333/HarDNet-MSEG - Last access on June 30th, 2022
[2] https://github.com/DengPingFan/Polyp-PVT - Last accessed on June 30th, 2022

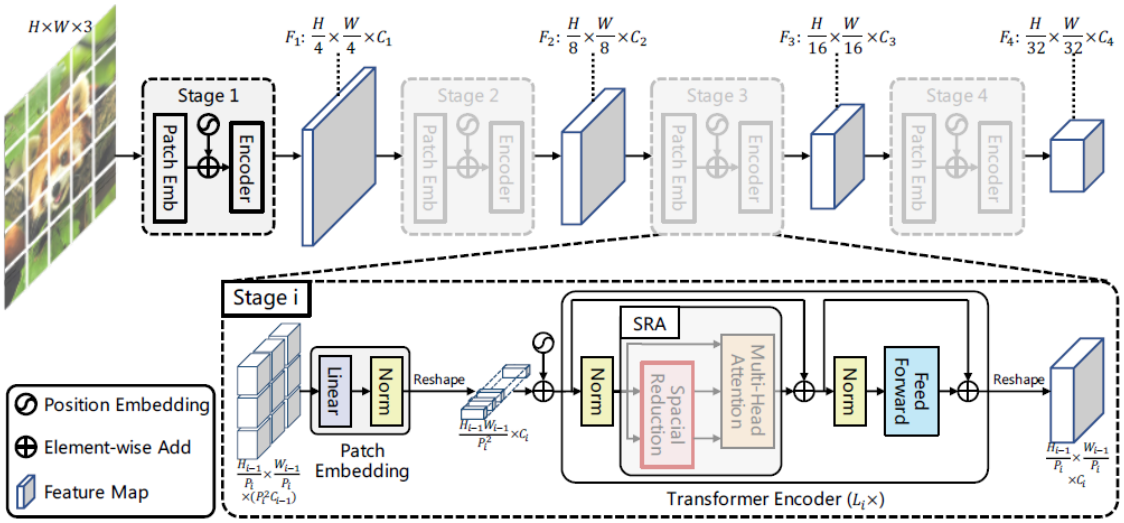

Figure 3.2: The Pyramid Vision Transformer (PVT) [31] has an architecture consisting of four stages, each containing a patch embedding layer and a Li-layer Transformer encoder. These stages follow a pyramid structure, with the output resolution progressively decreasing from high (4-stride) to low (32-stride).

3.2 Loss Function

3.2.1 Generalized Dice Loss

The Dice Loss is derived from the Sorensen-Dice coefficient, which is a commonly used metric to evaluate the performance of semantic segmentation models [69]. The Sorensen-Dice coefficient measures the similarity between two images on a scale from 0 to 1. To address the challenge of using Dice Loss for multiclass problems, the Generalized Dice Loss was introduced [95]. The Generalized Dice Loss formula compares the predicted values Y with the target values T. It is represented as:

$$L_{GD}(Y,T) = 1 - \frac{2 * \sum_{k=1}^{K} w_k * \sum_{m=1}^{M} Y_{km} T_{km}}{\sum_{k=1}^{K} w_k \sum_{m=1}^{M} (Y_{km}^2 + T_{km}^2)} \tag{3.1}$$

Here, K represents the number of classes, M represents the number of pixels. The formula incorporates a weighting factor w_k, which is used to emphasize on a specific region. The weight is inversely proportional to the label frequency for the given class k. The weight calculation is represented by the equation:

$$w_k = \frac{1}{\left(\sum_{m=1}^{M} T_{km}\right)^2} \tag{3.2}$$

3.2.2 Tversky Loss

One challenge that often arises in image segmentation is the imbalance in class distribution. To address this problem, the Tversky Loss was proposed [87]. This loss is based on the Tversky Index, which is an extension of the dice similarity coefficient. The Tversky Index employs two weighting factors, α and β, to balance the trade-off between false positives and false negatives. When both α and β are set to 0.5, the Tversky Index reduces to the Dice Similarity coefficient. The Tversky Index measures the similarity between the predicted values Y and the ground truth values T for a specific class k. The formula is expressed as:

$$TIk(Y,T) = \frac{\sum^{M} m = 1 Y_{pm} T_{pm}}{\sum_{m=1}^{M} Y_{pm} T_{pm} + \alpha \sum_{m=1}^{M} Y_{pm} T_{nm} + \beta \sum_{m=1}^{M} Y_{nm} T_{pm}} \tag{3.3}$$

Here, p refers to the positive class, n to the negative class, and M is the total number of pixels. The Tversky Loss formula is expressed as:

$$L_T(Y,T) = \sum_{k=1}^{K} \left(1 - TI_k\left(Y,T\right)\right) \tag{3.4}$$

where K is the number of classes. In this study, the weights are assigned as α= 0.3 and β= 0.7, which means more emphasis was given to false negatives.

3.2.3 Focal Tversky Loss

Cross-entropy (CE) is one of the widely used distribution-based loss functions. It works towards reducing the difference between two probability distributions, without any preference for larger or smaller regions. Numerous variations of the cross-entropy loss have been introduced in literature, including Binary Cross-Entropy and Focal loss [59]. Binary Cross-Entropy is a straightforward application of CE to binary classification problems. On the other hand, Focal loss aims to give more attention to challenging examples by down-weighting well-classified ones. This is achieved by incorporating a modulating factor $\gamma > 0$.Focal Loss (L_F) is particularly effective in scenarios where foreground and background classes are imbalanced. Similar loss functions that incorporate the γ factor to focus on hard examples are the Focal Tversky Loss [59] and Exponential Logarithmic Loss [110].

$$L_{FT}(Y,T) = L_T(Y,T)^{\frac{1}{\gamma}} \tag{3.5}$$

By using the Tversky Index, the Focal Tversky Loss manages to strike a favorable balance between precision and recall.

3.2.4 Focal Generalized Dice Loss

Inspired by the Focal Tversky Loss, we introduced the γ factor to Generalized Dice Loss to create Focal Generalized Dice Loss. This loss function emphasizes smaller regions of interest and down-weights commonly occurring examples. In our experiments, we set γ to $4/3$.

$$L_{FGD}(Y,T) = (L_{GD}(Y,T))^{\frac{1}{\gamma}} \tag{3.6}$$

3.2.5 Log-Cosh Type Losses

The Log-Cosh Dice Loss is a combination of the Dice Loss and the Log-Cosh function, which is commonly used in regression tasks to smoothen the curve. The Log-Cosh function, $\log(\cosh(x))$, approximates to $x^2/2$ for small values of x, and $|x| - \log(2)$ for large values of x. The Log-Cosh Generalized Dice Loss can be expressed as:

$$L_{lcGD}(Y,T) = \log(\cosh(L_{GD}(Y,T))) \tag{3.7}$$

In our experiments, we were motivated by Log-Cosh Dice Loss to make other loss function curves smoother. We introduce Log-Cosh Binary Cross Entropy Loss, Log-Cosh Tversky Loss, and Log-Cosh Focal Tversky Loss, which are all variations of Binary Cross-Entropy Loss, Tversky Loss, and Focal Tversky Loss. The only distinction is the inclusion of the Log-Cosh term. Specifically, Log-Cosh Focal Tversky Loss can be defined using the following formula:

$$L_{lcFT}(Y,T) = \log\left(\cosh\left(L_{FT}(Y,T)\right)\right) \tag{3.8}$$

3.2.6 Neighbor Loss

A novel loss function called Neighbor Loss [114] has been recently proposed. This loss function can be interpreted as a cross-entropy that incorporates weights for each pixel based on its eight neighboring pixels. The aim is to account for the spatial correlation among neighboring pixels. The weight assigned to a pixel is determined by the number of neighbors that have a prediction different from that of the center pixel. Similar to Focal Loss, Neighbor Loss attempts to handle difficult samples by incorporating a threshold t and a binary indicator function $1_\bullet$, which excludes easily classified pixels. Despite this approach, our experiments indicate that Neighbor Loss exhibits poor performance, and as a result, we do not include it in our proposed ensemble.

3.2.7 SSIM Loss

SSIM Loss [80] is an image quality estimation metric that originates from the Structural similarity (SSIM) index [106]. The formula for SSIM is given as:

$$SSim\,(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{3.9}$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} denote the local means, standard deviations, and cross-covariance for images x and y, and C_1, C_2 are regularization constants. The SSIM Loss between an image Y and its corresponding ground truth T is defined as:

$$L_S\,(Y,T) = 1 - SSim\,(Y,T) \tag{3.10}$$

We also introduce a modification, $L_{MS}(Y,T)$, which is defined similarly to L_S, but instead of using the SSIM index, it employs the Multiscale Structural Similarity (MS-SSIM) index.

3.2.8 Different Functions Combined Loss

When working with imbalanced data, like in the case of early cancer detection, there is a potential trade-off between high precision and low recall. To address this issue, Generalized Dice Loss employs a recurrent method that reduces the impact of class imbalance by introducing a weight w_k, which is the inverse of the label frequency. A drawback of Dice Loss is that it represents a harmonic mean of false positives and false negatives. To ensure that no lesion is overlooked, it is crucial to have the flexibility to balance false positives and false negatives, with a tendency towards weighting false negatives higher, as doctors do. To focus the model on challenging cases and harness the advantages of both Focal Generalized Dice Loss and Focal Tversky Loss, we merged them:

$$Comb_1(Y,T) = L_{FGD}(Y,T) + L_{FT}(Y,T) \tag{3.11}$$

An alternative approach to down-weighting simple examples is to blend Log-Cosh Dice Loss, Focal Generalized Dice Loss, and Log-Cosh Focal Tversky Loss. In this situation, we regulate the non-convex behavior of the curve by incorporating the Log-Cosh technique:

$$Comb_2\,(Y,T) = L_{lcGD}\,(Y,T) + L_{FGD}\,(Y,T) + L_{lcFT}\,(Y,T) \tag{3.12}$$

Lastly, we suggest a fusion of the SSIM Loss and the Generalized Dice Loss:

$$Comb_3(Y,T) = L_S(Y,T) + L_{GD}(Y,T) \tag{3.13}$$

3.2.9 Cross Entropy

The cross-entropy (CE) loss function offers a way to quantify the dissimilarity between two probability distributions. The objective is to reduce this dissimilarity, and as a result, it does not exhibit any bias towards small or large regions.

When working with imbalanced datasets, this can pose a problem. Therefore, the weighted cross-entropy loss was introduced to address this concern and has been shown to generate well-balanced classifiers in imbalanced scenarios [3].

The equation for weighted binary cross-entropy can be found in 3.14. Here, T represents the ground truth label image, and T_{ik} refers to the true value of pixel i, which can either be 0 or 1. If pixel i belongs to class k, T_{ik} is equal to 0; otherwise, it is 1.

P represents the prediction for the output image, and P_{ik} is the probability of the i-th pixel belonging to the k-th class, computed using the sigmoid activation function. In contrast, the softmax activation function is employed for P to obtain probabilities.

w_{ik} denotes the weight assigned to the i-th pixel of the image belonging to class k. To calculate these weights, we applied average pooling over the mask using a 31x31 kernel and a stride of 1, taking into account non-maximal activations as well.

$$L_{WBCE} = -\sum_{k=1}^{K}\sum_{i=1}^{N} w_{ik} T_{ik} \log\left(P_{ik}\right) \tag{3.14}$$

Here, K corresponds to the number of classes, while N refers to the number of pixels.

3.2.10 Weighted intersection over union

Intersection over Union (IoU) loss is another widely recognized loss function that was initially proposed in [81]. The original formulation is as follows:

$$IoU = \frac{|P \cap T|}{|P \cup T|} \tag{3.15}$$

Here, P and T represent the predicted and ground truth label images, respectively.

Regrettably, the symbols used for Intersection and Union are non-differentiable as P and T must be either 0 or 1. However, since this is not true for P, the original equation was approximated as follows:

$$IoU' = \frac{|P \cdot T|}{|P + T - P \cdot T|} \tag{3.16}$$

In this formula, $P \cdot T$ denotes the element-wise multiplication of T and P. The denominator subtracts the "intersection" between P and T to avoid counting the intersection twice.

After converting the set operators to arithmetic ones, the formula becomes differentiable, enabling us to compute the gradient.

However, IoU is a performance metric used to evaluate the quality of predictions, with a value of 1 denoting a flawless prediction. As a result, the loss function takes the following form:

$$L_{IoU} = 1 - IoU' \tag{3.17}$$

This loss function is used to quantify the error between the predicted and ground truth label images.

Nonetheless, we face the same challenge encountered earlier with CE since it can be challenging to determine the labels for object boundaries in general. As suggested in [26], we adopt the same approach as before and use weighted Intersection over Union ($wIoU$) instead of the standard IoU.

The formula for the weighted Intersection over Union loss is as follows:

$$L_{WIOU} = 1 - \frac{|wPT|}{|w\,(P+T) - wPT|} = 1 - \frac{\sum_{i=1}^{N} w_{ik} \sum_{k=1}^{K} T_{ik} Y_{ik} + 1}{\sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik} (T_{ik} + Y_{ik} - T_{ik} Y_{ik}) + 1} \tag{3.18}$$

In this equation, N corresponds to the number of pixels, and K represents the number of classes. The previously mentioned method is employed to calculate the weights w_{ik}. T_{ik} and Y_{ik} refer to the ground truth and predicted values, respectively, for pixel i belonging to class k. In order to avoid division by zero, we add 1 to both the numerator and denominator.

3.2.11 Structure Loss

Drawing on the insights from [46], we combine the weighted Intersection over Union and weighted binary-crossed entropy loss functions[67]. The resulting loss function is expressed as follows:

$$L^{'}STR = LwIoU + L_{wbce} \tag{3.19}$$

To enhance the diversity in the deep learning network, we have altered the original loss function as follows: rather than using avgpool over the mask, we apply it over the predictions.

We assign a weight of 2 to the binary-crossed entropy loss to give it greater significance.

As a result, the final loss function becomes:

$$L_{STR} = L_{wIoU} + 2L_{wbce} \tag{3.20}$$

3.2.12 BoundExpStructure

In order to enhance the model's ability to identify small structures within a highly imbalanced dataset, we have combined three loss functions: Structure Loss, Boundary Loss, and Exponential Logarithmic Loss.

The Structure Loss function is used to capture the global structural information of the image, while the Boundary Loss function is used to improve the detection of boundaries. The Exponential Logarithmic Loss function, on the other hand, is designed to handle the class imbalance in the dataset.

The final loss function, named BoundExpStructure, is expressed as:

$$L_{BoundExpS} = L_{Bound} + L_{Exp} + L_{Str} \tag{3.21}$$

By combining these loss functions, we are able to achieve better performance in identifying small structures and detecting boundaries, even in the presence of highly imbalanced data. This can be particularly important in medical imaging applications, such as the early detection of cancer, where small lesions and accurate boundary detection are critical for proper diagnosis and treatment.

3.2.13 Boundary Enhancement Loss

The Boundary Enhancement Loss, proposed in [113], is a loss function designed to focus explicitly on the boundary areas during training. This loss function has shown good performance without requiring any pre- or post-processing of the image or a specific network architecture.

The Laplacian filter $\mathcal{L}(\cdot)$ is a key component of the Boundary Enhancement Loss, as it generates strong responses around the boundaries and returns zero elsewhere. Specifically, when the Laplacian filter is applied to a mask S, it produces the following expression:

$$\mathcal{L}(\mathrm{x}, \mathrm{y}) = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \tag{3.22}$$

Using the Laplacian filter has the advantage of being relatively easy to achieve through a series of convolution operations. The approach involves computing the difference between the filtered output of the ground truth labels and the filtered output of the predictions.

The boundary enhancement loss is defined [113] as:

$$L_{BE} = ||\mathcal{L}(T) - \mathcal{L}(Y)||_2 = \left\| \frac{\partial^2 (T - Y)}{\partial x^2} + \frac{\partial^2 (T - Y)}{\partial y^2} \right\|_2 \tag{3.23}$$

The l_2 norm is denoted by $|| \bullet ||_2$ in equation 3.23. This operation can be easily performed as explained in the original paper by Yang et al. [113].

We will combine Dice Loss and Boundary Enhancement Loss, along with Structure Loss, in a weighted manner based on the approach in the paper. The resulting loss function can be expressed as:

$$L_{DiceBES} = \lambda_1 L_{Dice} + \lambda_2 L_{BE} + L_{Str} \tag{3.24}$$

The optimal results were obtained when we set λ_1= 1 and λ_2= 0.01

3.2.14 Contour-aware Loss

The Contour-aware Loss, initially introduced in [24], is a loss function that utilizes a weighted binary cross-entropy loss. The objective of these weights is to prioritize the borders of the image by assigning them greater significance.

The Contour-aware Loss utilizes a morphological gradient edge detector, which calculates the difference between the dilated and eroded label map. The resulting map is then smoothed with Gaussian blur for better results. This process generates a spatial weight map, which can be formulated as:

$$M^C = Gauss\left(K \bullet \left(dilate\left(T\right) - erode\left(T\right)\right)\right) + 1 \tag{3.25}$$

The operations $dilate(T)$ and $erode(T)$ represent dilation and erosion with a 5×5 kernel, respectively. The hyperparameter K is used to assign high values to contour pixels and was empirically set to 5. The matrix with a value of 1 in each position is denoted as $\mathbb{1}$.

The loss computation can be expressed using the following equation:

$$L_C = -\sum_{i=1}^{N} M_i^C * \left(T_i * \log\left(Y_i\right) + \left(1 - T_i\right) * \log\left(1 - Y_i\right)\right) \tag{3.26}$$

The final loss can be computed using the contour-aware loss L_C and the structure loss L_{Str}:

$$L_{CS} = L_C + L_{Str} \tag{3.27}$$

3.3 Data Augmentation

The goal of data augmentation techniques is to enhance performance by expanding the pool of training data without the need to gather new data. This is achieved by creating synthetic samples that either replicate the original ones with modifications or are automatically generated to possess the same statistical characteristics as the real samples, or both. Not only does data augmentation enhance classification accuracy, but it has also been found to enhance generalization and serves as a regularizer. The method used to generate additional samples is based on the classification field. A comprehensive review of data augmentation techniques for deep learning can be found in [91].

Essentially, methods for augmenting image data can be categorized into two main groups:

1. Image manipulations for data augmentation can be classified into several categories. Geometric transforms, such as rotation, flipping, warping, cropping, and others, can modify the spatial layout of the image. Filters like low-pass filters and noise injection can add noise to the image. Random erasing, as proposed by [123], is another technique that replaces random regions of pixels with a constant value or noise. Statistical methods, such as equalization and color casting, can transform the color space of the image. Additionally, per-pixel weighted mixes of other images can be employed to generate new images.

2. Deep learning techniques for image data augmentation include several approaches. Feature space augmentation involves using the lower-dimensional representations of images output by intermediate layers of convolutional neural networks to generate new data. Adversarial training is another approach that uses an auxiliary network to produce synthetic images that can mislead the main network. Generative modeling is another popular approach that employs a generative adversarial network (GAN) to generate synthetic images that are similar to real ones. While there are other options available, GANs are the most widely used due to the high quality of the images they produce. Neural style transfer is another technique that utilizes an auxiliary network to transfer the style of one image to another while preserving the original content. These deep learning approaches are powerful tools that can significantly improve the performance of image classification models.

Data augmentation techniques can be combined to further increase the number of available samples. For instance, synthetic images generated using deep learning approaches can undergo basic image manipulation to produce even more variations. However, as noted by [91], it is not always guaranteed that combining different techniques will improve performance.

In addition, data augmentation and ensemble techniques can also be combined to enhance model performance. For example, [70] propose an ensemble approach based on a kind of bagging, where multiple classifiers are trained on different training sets generated by combining fourteen data augmentation approaches. By using a variety of data augmentation techniques and training multiple classifiers on the resulting datasets, this approach can improve the accuracy and robustness of the model.

3.3.1 DA1

DA1 is a basic data augmentation which involves performing horizontal and vertical flips, as well as 90° rotations, on the input images.

Figure 3.3: Original image and label.

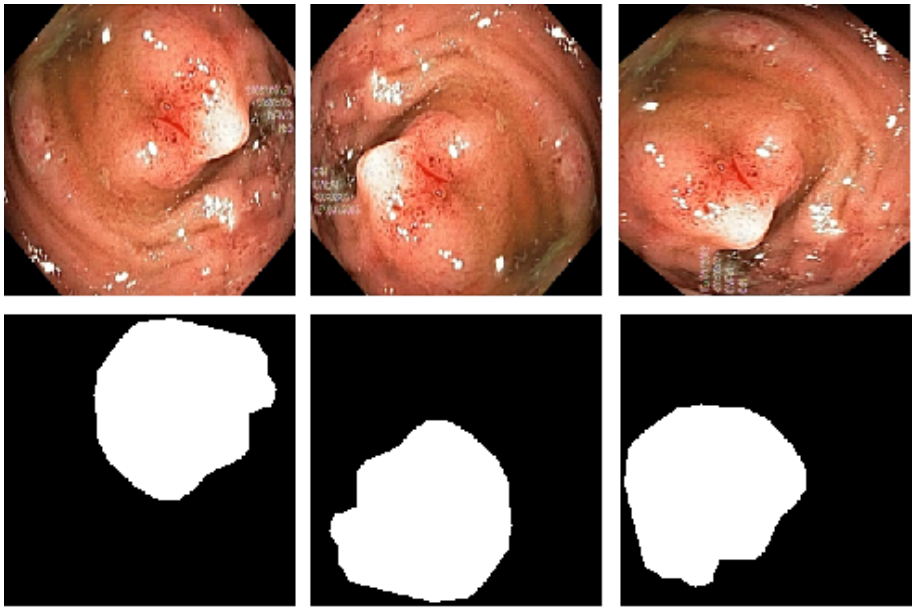

Figure 3.4: Application of DA1 to images.

Figure 3.5: Original image and label.

3.3.2 DA2

DA2 incorporates both shape-based and color-based transformations to produce eleven artificial images for each image in the dataset.

The transformations applied are:

1. Shift the image horizontally.
2. Shift the image vertically.
3. Rotate the image by a randomly selected angle between 0° and 180°.
4. Apply horizontal or vertical shear employing the function "randomAffine2d".
5. Flip the image horizontally or vertically.
6. Adjust the brightness levels by adding the same values to each RGB channel.
7. Adjust the brightness levels by adding different values to each RGB channel.
8. Add speckle noise by using the function "imnoise".
9. Apply the technique “Contrast and Motion Blur”, illustrated below.
10. Apply the technique “Shadows”, illustrated below.
11. Apply the technique “Color Mapping”, illustrated below.

3.3.3 Contrast and Motion Blur

The process of transforming the image involves two consecutive modifications. The first step involves adjusting the contrast of the original image, either by increasing or decreasing it. Subsequently, a filter that emulates camera movement is applied. A pair of contrast-modifying

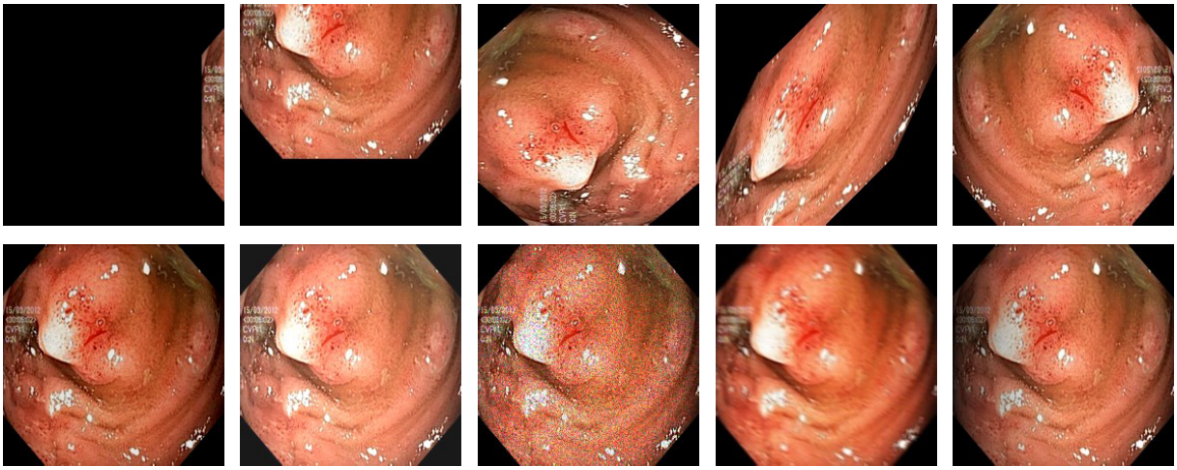

Figure 3.6: Application of DA2 to images.

Figure 3.7: Application of DA2 to labels.

functions have been incorporated, however, only one of the two is randomly selected and applied to the image. The first contrast modification function involves an equation represented as follows:

$$\frac{(x-\frac{1}{2})\sqrt{1-\frac{k}{4}}}{\sqrt{1-k(x-\frac{1}{2})^2}}+0.5, \text{ where } k \leq 4 \tag{3.28}$$

By manipulating the parameter k, the contrast of the image can be adjusted. Specifically, a decrease in contrast can be achieved when $0 < k \leq 4$, while an increase in contrast can be obtained if $k < 0$. In the case where $k = 0$, the image remains unchanged. The code selects a random value for the parameter k from a set of four predefined ranges. These ranges are as follows:

- U(2.8, 3.8) → This range leads to a hard decrease in contrast.
- U(1.5, 2.5) → This range leads to a soft decrease in contrast.
- U(-2, -1) → This range leads to a soft increase in contrast.
- U(-5, -3) → This range leads to a hard increase in contrast.

The second contrast function employed in the code is defined by :

$$y = \begin{cases} \frac{1}{2}(\frac{x}{0.5})\alpha, & 0 \leq x < \frac{1}{2} \\ 1-\frac{1}{2}(\frac{1-x}{0.5})\alpha, & \frac{1}{2} \leq x \leq 1 \end{cases} \tag{3.29}$$

In this function, the contrast is controlled by the parameter α. Specifically, an increase in contrast occurs when $\alpha > 1$, a decrease in contrast occurs when $0 < \alpha < 1$, and the image remains unchanged when $\alpha = 1$. The value of the parameter α is randomly selected from one of four possible ranges in the code. These ranges are:

- U(0.25, 0.5) → This range results in a hard decrease in contrast.
- U(0.6, 0.9) → This range results in a soft decrease in contrast.
- U(1.2, 1.7) → This range results in soft increase in contrast.
- U(1.8, 2.3) → This range results in hard increase in contrast.

3.3.4 Shadows

To create the final image, a shadow is added to either the left or the right side of the original image. This is accomplished by multiplying the intensities of each column of the image with

the following equation:

$$y = \begin{cases} \min\left\{0.2 + 0.8\sqrt{\frac{x}{0.5}}, 1\right\} & direction = 1 \\ \min\left\{0.2 + 0.8\sqrt{\frac{1-x}{0.5}}, 1\right\} & direction = 0 \end{cases}$$

Certain artificial images may consist entirely of background pixels. To eliminate such images, those with less than 10 foreground pixels are discarded.

3.3.5 Color Mapping

By altering the color map of an image, a new image can be generated. Specifically, it's feasible to match the colors of one image to those of another image. We created pairs of images by combining any image from the original training set with another image chosen randomly from the same set. We utilized the Stain Normalization toolbox, which offers this capability in three distinct versions:

- RGB Histogram Specification
- Reinhard
- Macenko

3.3.6 JPEG approach

The JPEG (Joint Photographic Experts Group) standard was defined in the years 1986-1992 by working groups set up by two organizations, the CCITT (Consultative Committee on International Telegraph and Telephone) and the ISO (International Organization for Standardization) with the following goals:

- Use the most advanced techniques then available.
- Allow the user to vary the compression ratio as desired.
- Create an algorithm independent of content, size, and image resolution.
- Keep computational complexity low.

More recent techniques, based on fractals and wavelets (e.g. JPEG 2000) have superior performance but require computational complexity considerably larger. For this JPEG is still the de facto standard in many fields of multimedia (from digital cameras to the Web). In this work, we exploit the idea of JPEG, to create our augumented images. In particular, starting from the original images, the subsequent steps were followed:

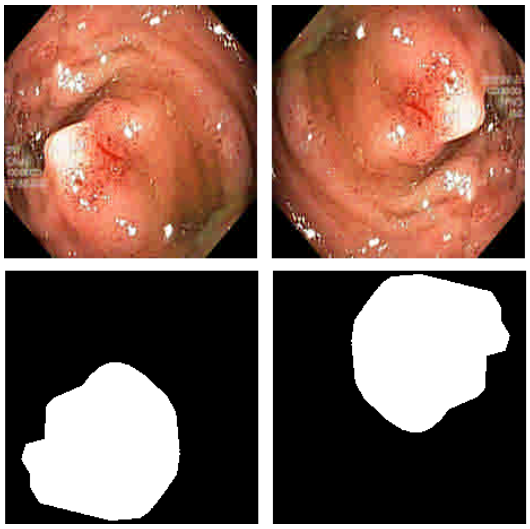

Figure 3.8: Application of the jpeg data augmentation technique.

1. Divide the image in 8 x 8 blocks.
2. Compute DCT separately, for each block.
3. Every element of each 8 x 8 block of DCT coefficients is divided by the corresponding coefficient of a quantization table Q. The tables of quantization are multiply by a scale factor α reflecting the degree of compression you want to achieve. In our code, we set $\alpha = 5$.
4. Compute the inverse of DCT separately, for each block.

Finally, given the compressed image, two operations were performed:

- horizontal flip;
- vertical flip.

3.3.7 Alternating Vertical Shift

The "Alternating Vertical Shift" method divides the image into alternating vertical strips and randomly shifts each strip up or down by a random amount.

3.3.8 Alternating Horizontal Shift

The "Alternating Horizontal Shift" technique takes a set of training images and labels and applies horizontal shifts to alternating horizontal strips in a symmetric manner. The strip height is chosen randomly within the specified minimum and maximum height range. For each strip, the function shifts the content to the left, while the following strip is shifted to the right. The process is repeated for all alternating strips in the image.

3.3.9 Alternating Diagonal Shift

The "Alternating Diagonal Shift" function takes a set of training images and labels, and applies diagonal shifts to alternating square regions in a symmetric manner. The size of the square regions is chosen randomly within the specified minimum and maximum size range. For each square region, the function shifts its content diagonally towards the top-left corner, while the content of the next square region (in both x and y directions) is shifted towards the bottom-right corner.

3.3.10 Random Shift with Black or Wrap

The "Random Shift with Black or Wrap" method, for each image, randomly shifts the image left or right by a random amount within the specified shift width. Then, it either fills the resulting empty space with a black strip or wraps the cut piece around to the other side.

3.3.11 Random Shift Up Down with Black or Wrap

The "Random Shift Up Down with Black or Wrap" function, for each image, randomly shifts the image up or down by a random amount within the specified shift width. Then, it either fills the resulting empty space with a black strip or wraps the cut piece around to the other side.

3.3.12 Random Rectangles Mix and Blackout

The "Random Rectangles Mix and Blackout" function takes a set of training images and labels, and applies random rectangle transformations to each image. The function either blacks out the rectangle or mixes it with another random rectangle from the same image. It creates a specified number of rectangles (default is 10) with random sizes and positions within the given minimum and maximum size range.

3.3.13 Random Rectangle Rotation

This technique selects a random number of small rectangles with random dimensions and positions, then rotates each rectangle by a random angle to simulate added noise.

3.3.14 Random Rectangle Flip

This method selects a random number of small rectangles with random dimensions and positions, then flips each rectangle randomly, vertically or horizontally to simulate added noise.

3.3.15 Random Rectangle Brightness

This function selects a random number of small rectangles with random dimensions and positions, then apply brightness to each rectangle by a random amount of brightness to simulate added noise.

Chapter 4

Results

We perform a thorough empirical evaluation to assess the performance of our ensemble system. This evaluation includes a comparison with multiple state-of-the-art models for a comprehensive analysis of our system. The empirical evaluation is conducted on six real-world scenarios including polyp segmentation, skin segmentation, leukocyte identification, butterfly identification, microorganism identification, and radiology segmentation.

4.1 Metrics

The performance of our system has been evaluated through two commonly used metrics: the Dice score and the Intersection over Union (IoU). The following formula shows how these metrics are calculated using true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

The **Dice score** (also known as the F1-score in binary classification tasks) is a measure of the performance of a semantic segmentation model that takes into account both precision and recall. It is calculated as a weighted average of these two metrics and can be defined mathematically as:

$$\text{F1-score} = \text{Dice} = \frac{2 \cdot |A \cap B|}{|A| + |B|} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}} \tag{4.1}$$

The Intersection over Union (IoU) metric measures the overlap between two masks by dividing the shared area by the total area of both masks combined. This is mathematically defined as:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{4.2}$$

The labels A and B refer to the predicted mask and the ground truth map, respectively. As the input size of the models used in our experiments differs from the original image size, we resize the images accordingly. However, to ensure consistency and enable comparison, we always resize the predicted masks back to their original dimensions.

Table 4.1: Datasets for Skin Segmentation. ECU dataset is split in 2000 images for training and 2000 for test set. For ECU, we considered the subset of images that were not used in the training phase.

Tag	Name	#Samples	Ref.	Available
CMQ	Compaq	4675	[51]	Ask Authors
HGR	Hand Gesture Recognition	1558	[52]	Yes
MCG	MCG-skin	1000	[47]	Ask Authors
PRT	Pratheepan	78	[96]	Yes
SFA	SFA	1118	[14]	Ask Authors
SCH	Schmugge dataset	845	[89]	Yes
VMD	Human activity recognition	285	[88]	Yes
ECU	ECU Face and Skin Detection	2000	[77]	N/A
UC	UChile DB-skin	103	[85]	Ask Authors
VT	VT-AAST	66	[1]	Ask Authors

4.2 Datasets

4.2.1 Skin segmentation (SKIN)

The skin detection task involves distinguishing the regions of an image that correspond to "skin" and "non-skin," which is essentially a binary classification problem[28]. In this thesis, we adopt the methodology proposed by Lumini et al. [63], which employs a small training set of 2000 images from the ECU dataset [77], as well as ten diverse datasets, listed in Table 4.1. Following the original testing protocol proposed by Lumini et al. [63], we calculate the Dice coefficient (i.e., F1-score) at the pixel level rather than the image level, and average the results over the entire dataset. In our experiments, we utilize resized images of dimensions 352×352 for all the datasets.

To facilitate research in the field of skin detection, several color image datasets with annotated ground truth are available. It is crucial to employ a standard and representative benchmark to perform a fair empirical evaluation of skin detection methods. In Table 4.1, we present a summary of some of the most widely used datasets, along with a brief description of each of them in this section.

Jones and Rehg's **Compaq** dataset (2002) is among the earliest and most widely used large-scale skin datasets. It comprises images acquired by crawling the Web, containing a total of 9731 images with skin pixels (though only 4675 images with segmented skin regions are included in the ground truth), as well as 8965 images with no skin pixels. The dataset has been widely utilized to test and compare various methods, but due to the lack of a standardized testing protocol, the comparisons made using this dataset may not always be impartial. Additionally, the ground truth for this dataset was obtained through an automated software tool, resulting in imprecise outcomes.

The **HGR** dataset, created by Kawulok, Kawulok, Nalepa, et al. in 2014, is a collection of images for gesture recognition. The dataset also includes ground truth binary masks that indicate the presence of skin. It consists of 1558 images representing Polish and American sign language gestures, captured with both controlled and uncontrolled backgrounds. The dataset is segregated into three subsets, namely HGR1, HGR2A, and HGR2B. During the testing phase, the size of the images in subsets HGR2A and HGR2B was downsized by a factor of 0.3.

The **MCG** skin database is a collection of 1000 images sourced from the internet to ensure the inclusion of challenging backgrounds, varying ambient lighting conditions, and a diverse range of human races. The ground-truth annotations were generated through manual labeling, although they may not be entirely accurate, as features such as eyes, eyebrows, and even bracelets may have been mistakenly labeled as skin.

Tan, Chan, Yogarajah, and Condell (2012) developed the **Pratheepan** dataset, which consists of a limited set of 78 images that were randomly downloaded from Google. The dataset is categorized into two groups: FacePhoto, which contains 32 single-subject images with plain backgrounds, and FamilyPhoto, which encompasses 46 images with elaborate backgrounds and multiple subjects.

The **SFA** dataset, developed by Casati, Moraes, and Rodrigues in 2013, is comprised of images from the FERET (876 images) and AR (242 images) face databases that have been manually labeled with moderate precision. The SFA dataset is organized into folders to segregate the 1118 original images (ORI), 1118 ground truth (GT) masks, 3354 skin samples (SKIN), and 5590 non-skin samples (NS), which vary in dimensions from 1 to 35×35. For the purposes of the study, ORI/GT was used to evaluate the model's performance.

The **Schmugge** skin dataset, developed by Schmugge, Jayaram, Shin, and Tsap in 2007, is a compilation of 845 images obtained from various face datasets, including the UOPB dataset, AR face dataset, and University of Chile database. The dataset is accurately labeled, with all images classified into one of three categories: skin, not-skin, or don't care.

The **Human activity recognition** dataset encompasses EDds2, LIRIS3, SSG4, UT5, and AMI6 datasets. These datasets exhibit a diverse range of scenarios, viewing distances, and resolutions, making skin detection a challenging task due to various factors such as illumination changes and poor visibility. The dataset consists of 285 images, and the corresponding ground truth was generated at the pixel level for approximately 50 images from each dataset. The evaluation set includes more than 870,000 skin pixels.

ECU (Phung, Bouzerdoum, & Chai, 2005) skin and face datasets which comprise approximately 4000 color images that are annotated with ground truth that is relatively accurate. The dataset is a significant challenge because it features a diverse range of skin types, lighting conditions, and background scenes.

The **UChile** (Ruiz-Del-Solar & Verschae, 2004) consists of 103 images captured in a variety of lighting situations and intricate backgrounds. While the dataset's ground truth annotations were produced with reasonable precision, some images may not have precisely delineated boundaries between skin and non-skin pixels.

The **VT-AAST** image database is a collection of color face images developed by researchers at Virginia Tech and the Arab Academy for Science, Technology, and Marine Transport. Its purpose is to help researchers evaluate how well automatic face detection algorithms and human skin segmentation techniques work. The dataset contains four parts: a set of 286 color photographs with over 1,000 faces captured in various settings, a set of the same images in a different file format, a set of image files that contain manually segmented human skin regions, and a set of the same skin regions in grayscale. Anyone can access the database online for free, as long as it's for noncommercial use. The dataset contains images of diverse indoor and outdoor scenes captured by consumer-grade digital cameras from different manufacturers. The original images are in JPEG format, while the compressed ones are in GIF format with 300x225 pixels per image. The images have a size range of 3 to 5.2 megapixels and depict a variety of backgrounds, facial expressions, poses, and orientations, as well as differing luminance conditions and structural features such as hair, beards, mustaches, and glasses. Moreover, the dataset comprises images of people of various genders and races.

The manual segmentation method employed in the dataset is used to isolate skin areas in each image and is expected to aid in the creation of automated skin detection systems. The dataset is valuable to face detection research since it provides a vast collection of images that can be used as a benchmark for the development and comparison of face detection and skin segmentation algorithms, addressing a previously unfulfilled need.

4.2.2 Segmentation in Radiology: VinDr-RibCXR

The VinDr-RibCXR dataset ([73]) is a small, publicly accessible collection of 245 anterior-posterior chest X-ray images and corresponding masks, created by expert annotators. It is designed for the segmentation and labeling of the anterior and posterior ribs (as shown in Figure 4.1). The raw DICOM format images were obtained from the VinDr-CXR dataset, and all patient information has been removed to maintain privacy. The labeling tool, VinDr Lab,

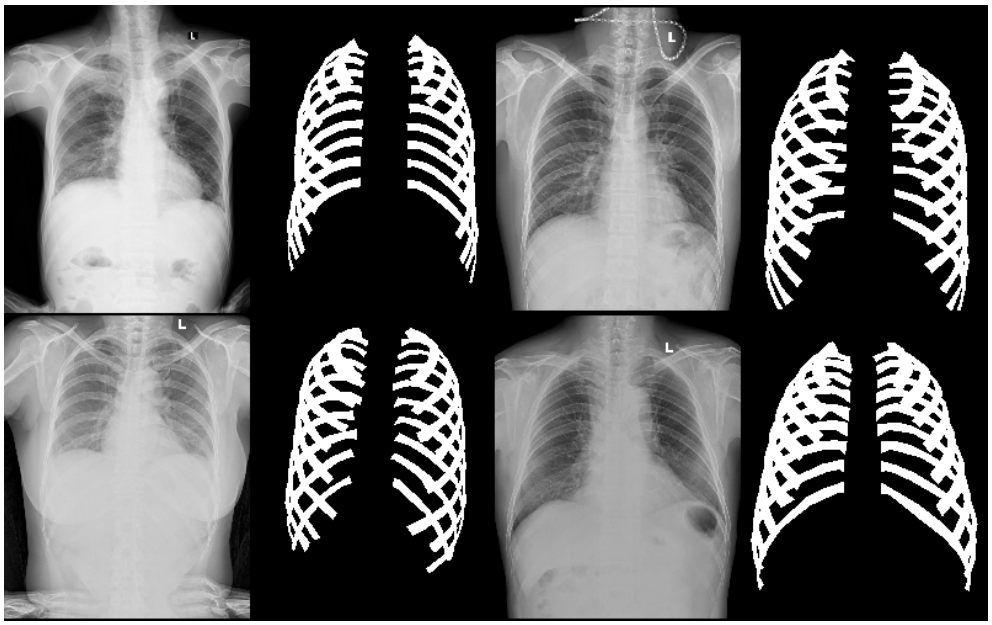

Figure 4.1: Examples of images from the VinDr-RibCXR dataset together with ground truth masks adopted in our experiments.

was used by experts to segment and annotate the 20 individual ribs (L1 to L10 for the left ribs and R1 to R10 for the right ribs) at the pixel level. The resulting masks were stored in a JSON file for future use in training instance segmentation models. VinDr-RibCXR is considered the first publicly available dataset with annotations for individual rib segmentation, covering both anterior and posterior ribs.

4.2.3 Polyp segmentation (POLYP)

Segmenting polyps from colonoscopy images is a difficult task that involves differentiating between the low contrast background of the colon and the polyp foreground pixels. Our study presents results based on a well-known benchmark [44] available on GitHub[1], which includes five datasets for polyp segmentation (Kvasir [50], ColonDB [9], CVC-T [101] and ETIS [92] and ClinicalDB [10])). The training set is made up of 1450 images, mostly from the largest dataset (Kvasir) with 900 images and ClinicalDB with 550 images. The remaining images are used for testing, with 100 from Kvasir, 380 from ColonDB, 60 from CVC-T, 196 from ETIS, and 62 from ClinicalDB, following common practices in the field. In these datasets, we use resized images with a size of 352×352.

[1] https://github.com/james128333/HarDNet-MSEG

Kvasir

The KVASIR dataset [50] is a collection of images taken during endoscopic procedures in the gastrointestinal tract. It was created as part of the medical multimedia challenge hosted by MediaEval. The images in the dataset were annotated and verified by medical doctors and include 8 different classes, including three anatomical landmarks, three pathological findings, and two other classes related to the polyp removal process. In total, the dataset contains 8,000 endoscopic images, with 1,000 images for each class. The data for the study was gathered at the Vestre Viken Health Trust (VV) in Norway using endoscopic equipment. VV is made up of four hospitals that provide healthcare to 470,000 individuals. One of these hospitals, the Baerum Hospital, has a large gastroenterology department and will contribute more data to the dataset in the future. The images were meticulously labeled by medical experts from both VV and the Cancer Registry of Norway (CRN). CRN is an independent institution under Oslo University Hospital Trust that focuses on researching and gaining new knowledge about cancer. It is also responsible for the national cancer screening programs with the goal of early detection and prevention of cancer death.

COLON-DB

The COLON-DB dataset 4.2 [9] was created to evaluate the effectiveness of techniques for segmenting and describing images. It features 15 random cases, each annotated by medical experts who identified all sequences that contained polyps. The experts then selected a random sample of 20 frames per sequence, with a size of 500×574 pixels, and cropped the central portion of the images to exclude any non-functional black borders. They made sure that each of the 20 frames showed a distinct viewpoint of the scene. The dataset only includes frames that contain a polyp in order to maximize the diversity of the images. It comprises 300 images that showcase a variety of polyp appearances. You can access the COLON-DB dataset by following this link: http://mv.cvc.uab.es/projects/colon-qa/cvccolondb.

CVC-T

The CVC-EndoSceneStill dataset (CVC-T)[101] is a benchmark for endoluminal scene object segmentation created by combining two datasets, CVC-ColonDB and CVC-ClinicDB, and contains 912 images from 44 video sequences taken from 36 patients. The CVC-ColonDB contains 300 images with polyp masks from 13 polyp video sequences, while the CVC-ClinicDB contains 612 images with polyp and background (mucosa and lumen) masks from 31 polyp video sequences. The annotations have been updated to include lumen, specular highlights, and a void class for black borders. The dataset is split into three sets: training (60%), validation (20%), and test (20%) with one patient not appearing in multiple sets. The training set includes 20 patients

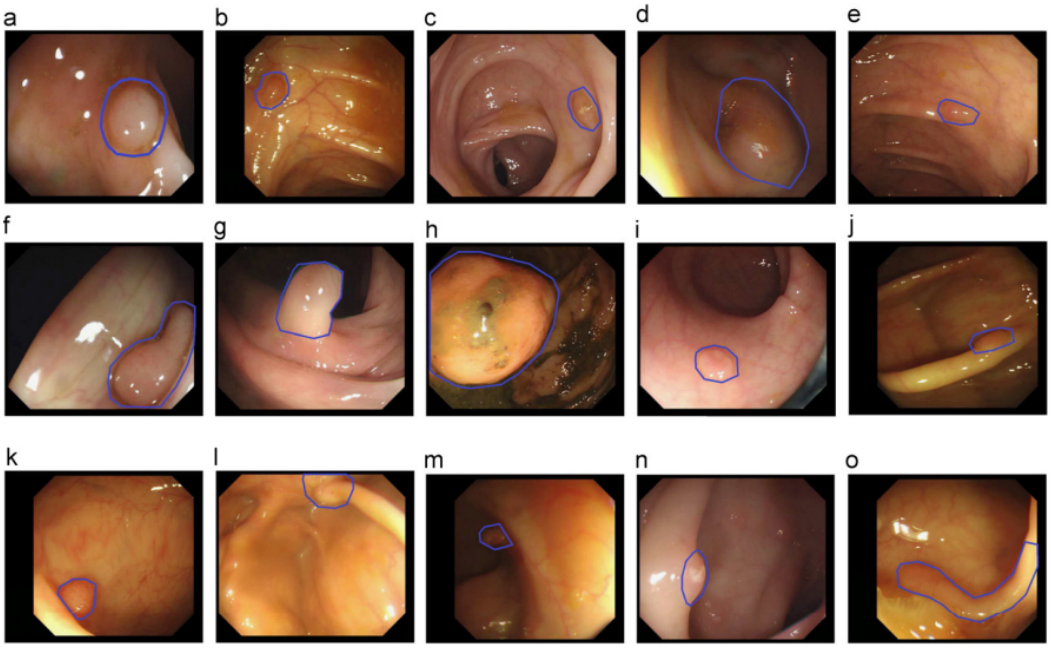

Figure 4.2: The figures (a) to (o) show examples of the various types of polyps present in each colonoscopy video of the COLON-DB database. The polyps are highlighted by blue contours, corresponding to video 1 to video 15 respectively. [9]

and 547 frames, the validation set includes 8 patients and 183 frames, and the test set includes 8 patients and 182 frames.

ETIS

The primary dataset [9] is used to construct the ETIS dataset [92], which involves segmenting each image into five regions. A gastroenterology specialist divides the image into five thumbnails, each representing a region of interest (ROI). The first thumbnail (a) depicts the polyp, while the other four thumbnails (b-e) depict non-polyp regions. The final dataset comprises 1,500 images, with 300 images displaying polyps and 1,200 images displaying non-polyps. To ensure accuracy and precision, each image is labeled by a specialist. This comprehensive dataset facilitates the development and training of machine learning models to detect and identify polyps in medical images, which is crucial for the early diagnosis and treatment of colorectal cancer.

CVC-ClinicDB

CVC-ClinicDB [10] database was developed in partnership with the Hospital Clinic of Barcelona, Spain. The database comprises 612 polyp images with a size of 576 x 768, which were extracted from 23 standard colonoscopy video studies using white light. To ensure diversity in polyp appearance, sequences containing a polyp were extracted from each study, and frames with poor visualization quality or high levels of patient preparation were excluded. The resulting database

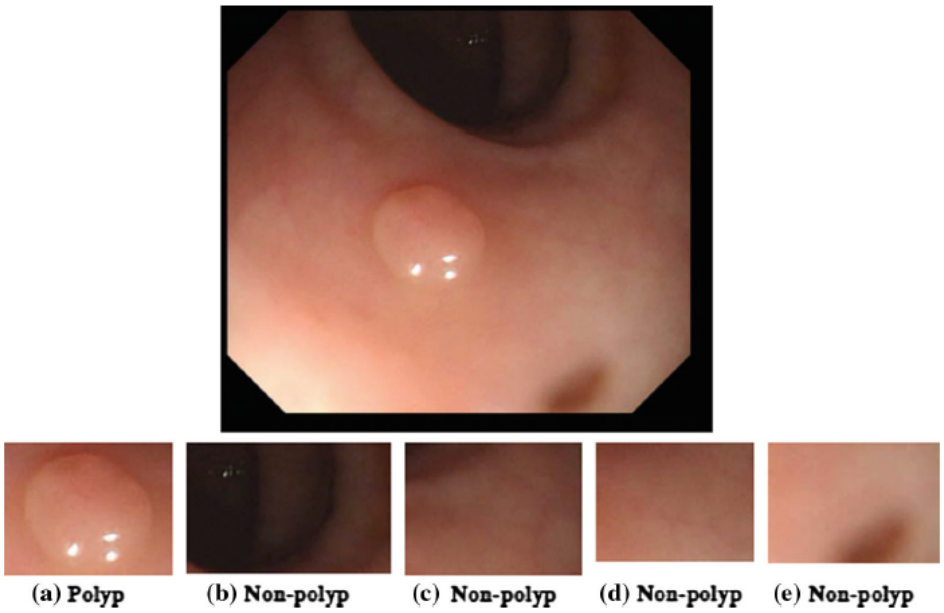

Figure 4.3: Here is an example of how the learning/testing ETIS database [92] is created from the primary data in [9].

contains 31 frame sequences, each with an average of 25 frames, and a total of 31 different polyps.

The authors manually created a ground truth for each frame by defining a mask on the region covered by the polyp. They also provided ground truth for specular highlights to assess the impact of image preprocessing on polyp localization results.

4.2.4 Leukocyte segmentation (LEUKO)

The task of leukocyte recognition involves the segmentation of white blood cells from the background, which is essential for the diagnosis of numerous diseases, including leukemia and infections. In our experiment, we utilized the LISC database [82], which is freely available and contains 250 hematological images that were extracted from the peripheral blood of eight healthy individuals. The database can be accessed at [2]. High-resolution images (720x576 pixels) were obtained and manually labeled to segment ten distinct types of leukocytes. In this study, our focus is on the segmentation performance and not classification. As recommended by the dataset authors, we employ a 10-fold cross-validation testing protocol where Dice results are computed at the image level, averaged for each fold, and then across all 10 folds. We use resized images with dimensions of 513x513 for this dataset.

[2] http://users.cecs.anu.edu.au/~ $hrezatofighi/Data/Leukocyte\%20Data.htm$

4.2.5 Butterfly identification (BFLY)

The butterfly identification task [103] in our study utilized the public dataset from [3], which has also been used in prior research.

The Butterfly Identification (BFLY) dataset includes ten distinct butterfly species selected for their unique features and characteristics. Unlike "top-level" categories such as people, cars, or bicycles, butterfly species have limited "global" characteristics that can be utilized to differentiate between them, such as part configurations. To gather data for the ten categories, natural text descriptions were acquired from the eNature online nature guide.

To ensure comparability with previous studies, we followed the testing protocol recommended by the dataset authors, which involves a four-fold cross-validation with 624 training images and 208 test images per fold. We resized the images in the dataset to 513×513.

4.2.6 Microorganism identification (EMICRO)

The microorganism identification task in our study utilized the EMicro dataset [119], which is publicly available at [4] (accessed on 20 April 2022).

EMicro is a subset of the Environmental Microorganism Image Dataset Sixth Version (EMDS-6), consisting of 1680 images, with 21 distinct classes of EM images, each consisting of 40 images. Thus, there are 840 unique images, and each of them has a corresponding ground truth (GT) image, bringing the total number of GT images to 840. Several individuals have significantly contributed to the production of the EMDS-6 dataset, which involved collecting images between 2012 and 2020. In particular, The EMDS-6 dataset features GT images were produced by Prof. Dr.-Ing Chen Li, M.E. Bolin Lu, M.E. Xuemin Zhu, and B.E. Huaqian Yuan of Northeastern University in China. The GT images adhere to defined labeling rules in which the foreground region containing the microorganism is designated white, while the background is marked black.

To ensure comparability with the original paper, we divided the dataset such that 37.5% of the images were allocated to the test set. We resized all images in the dataset to a uniform size of 513×513.

4.3 Experiments

4.3.1 Skin Segmentation

We aim to highlight the significance of deep learning techniques by comparing the results of our ensemble with those reported in a recent survey [63]. The results of various methods developed

[3] http://www.josiahwang.com/dataset/leedsbutterfly/

[4] https://figshare.com/articles/dataset/EMDS-6/17125025/1

Figure 4.4: The provided samples include images and corresponding masks for polyp segmentation, skin segmentation, leukocyte identification, butterfly recognition, and microorganism identification [67].

Table 4.2: Performance (Dice=F1-score) in the skin detection problem. Best performance in bold.

Method	PRT	MCG	UC	CMQ	SFA	HGR	SCH	VMD
GMM	0.581	0.688	0.615	0.600	0.789	0.658	0.595	0.130
Bayes	0.631	0.694	0.661	0.599	0.760	0.871	0.569	0.252
SPL	0.551	0.621	0.568	0.494	0.700	0.845	0.490	0.321
Cheddad	0.597	0.667	0.649	0.588	0.683	0.767	0.571	0.261
Chen	0.540	0.656	0.598	0.549	0.791	0.732	0.571	0.165
SA1	0.613	0.664	0.567	0.593	0.788	0.768	0.482	0.199
SA2	0.693	0.755	0.663	0.645	0.771	0.806	0.594	0.156
SA3	0.709	0.762	0.625	0.647	0.863	0.877	0.586	0.147
DYC	0.599	0.680	0.663	0.618	0.569	0.616	0.613	0.275
SegNet	0.730	0.813	0.802	0.737	0.889	0.869	0.708	0.328
U-Net	0.787	0.779	0.713	0.686	0.848	0.836	0.671	0.332
DeepLab	0.875	0.879	0.899	0.817	0.939	0.954	0.774	0.628
Vote1	0.717	0.754	0.670	0.666	0.737	0.849	0.625	0.269
Vote2	0.811	0.816	0.81	0.772	0.854	0.949	0.700	0.481
Vote3	0.812	0.841	0.829	0.773	0.902	0.950	0.714	0.423
Vote4	0.879	0.878	0.897	0.819	0.944	0.967	0.776	0.620
Hardnet	0.913	0.880	0.900	0.809	0.951	0.967	0.792	0.717
PVT	0.920	0.888	0.925	0.851	0.951	0.966	0.792	0.709
Ensemble	**0.927**	**0.894**	**0.932**	**0.868**	**0.954**	**0.971**	**0.797**	**0.767**

for skin detection are displayed in Table 4.2. The significant improvement in performance from SegNet to Ensemble is largely due to the adoption of deep learning and attention-based techniques. As shown in Table 4.2, the Methods Vote*x* represent a combination of handcrafted methods and deep learning approaches reported in [63]. The training for both Hardnet and PVT was done using the Adam optimization algorithm and with the inclusion of DA1 data

augmentation.

We tested various combinations of HardNet and PVT for the ensemble, incorporating different data augmentations and loss functions. All of these combinations delivered better results compared to previous models. We only present the highest performing ensemble method in this work. The ensemble method involves combining the individual predictions of two HardNet models trained with SGD, two HardNet models trained with the Adam optimizer, and two PVTs. Each pair of segmentators has one model trained with DA1 data augmentation and the other with DA2 data augmentation, and the predictions are combined using the sum rule. The ensemble method exhibits improved performance compared to the individual classifiers employed as baselines. The results demonstrate that the ensemble method outperforms the single individual classifiers employed as baselines. This supports the idea that incorporating diverse individual classifiers, which generalize differently in the training space, can result in an enhancement of the final performance of the ensemble, as noted in [27]. This improvement can be attained by varying the type of data augmentation or the types of individual classifiers utilized.

4.3.2 Radiology Segmentation

This section showcases new experiments on semantic segmentation of ribs in chest radiographs, which are examined as a case study. The training and testing samples used in the experiments are sourced from VinDr-RibCXR dataset ([73]). The experiments have a threefold objective: (1) investigating the learning potential of some state-of-the-art models for semantic segmentation, as well as evaluating the effectiveness of (2) ensembles and (3) data augmentation when dealing with small datasets like the one used in this study.

For semantic segmentation, we utilize DeepLabV3+ network. In this case study, we investigate ResNet101 ([43]), a popular CNN that utilizes the input block to obtain a residual function. Specifically, we employ the pre-trained ResNet101 model on the VOC segmentation dataset with the proposed parameters (to avoid overfitting, the parameters remain identical for all datasets under examination):

1. initial learning rate = 0.01;
2. number of epochs = 10 (utilizing DA1, which is the simpler data augmentation method.) or 15 (using DA2, the more sophisticated data augmentation method that generates a larger training set, resulting in slower convergence);
3. momentum = 0.9;
4. L2 Regularization = 0.005;
5. Learning Rate Drop Period = 5;

6. Learning Rate Drop Factor = 0.2;

7. Random selection of training images every epoch;

8. Optimizer = SGD (Stochastic Gradient Descent).

This study employs two distinct protocols: the first protocol follows the recommendation of [73], while the second protocol employs a reduced training set, while the test set remains unchanged. For greater clarity, the following are the specifics of the protocols:

- TRUE_Full: the training set comprises 196 images, and the remaining 49 images are used for the test set.

- TRUE_10: the training set is composed of 10 images, and the test set remains the same as in TRUE_Full, which consists of 49 images.

For the first experiment, we use the TRUE_Full protocol and DA1 as the data augmentation technique. For each image, its mask M is randomly chosen and then mutated. The resultant mask is obtained by assigning the upper left region to the positive class and the lower right region to the negative class:

- M(1:200,1:200) = 1,

- M(301:end,301:end) = 0.

The inspiration for this experiment is based on the discoveries of [115], where it is demonstrated that popular convolutional networks utilized for image classification can effortlessly adapt to random labeling of the training data. Our semantic segmentation results support these observations, as we can see from Figure 4.5, where the loss and accuracy metrics converge despite training being conducted with random masks. However, the performance on the test set is notably poor, with a Dice score of 0.386.

Likewise, for the second experiment, we adopt the protocol recommended in the source paper, but this time we associate each image with its corresponding actual mask. This experiment reveals faster convergence and an improvement in the Dice metric on the test set, which corresponds to 0.776, in comparison to the first experiment. However, it results in a higher final loss.

Several methods were tested, which involved different ensembles generated using the sum rule, where the models were merged by adding their scores. The outcomes of these experiments are provided in Table 4.3.

1. RN101 is a distinctive model that incorporates Resnet101 as the backbone and DeepLabV3+ as the architecture.

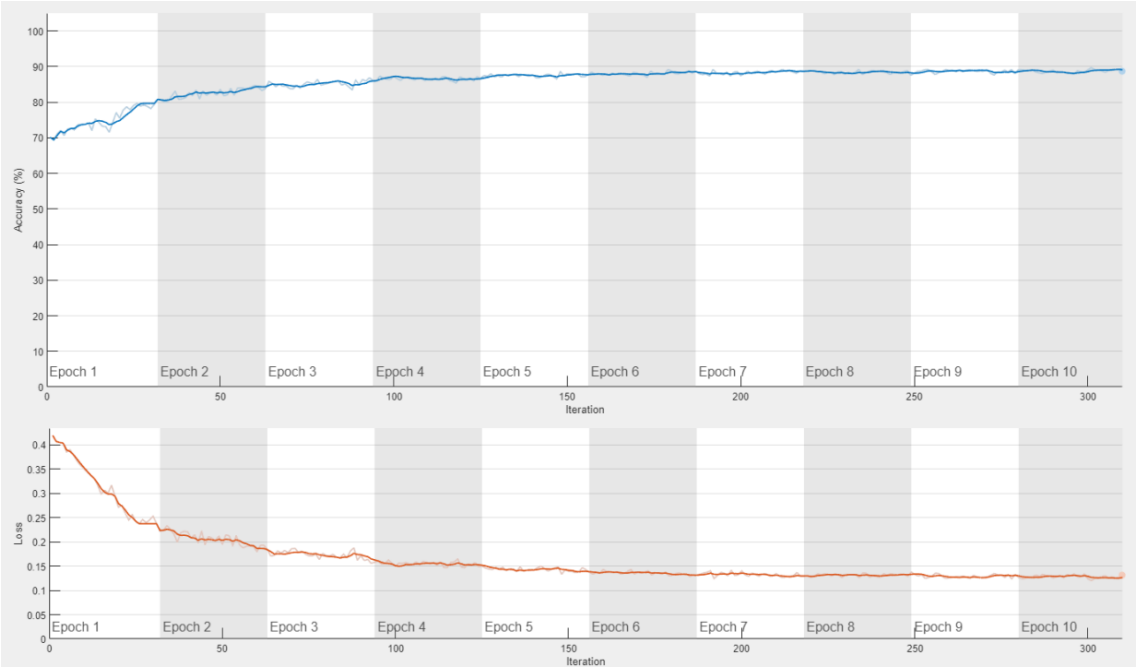

Figure 4.5: Convergence with random masks.

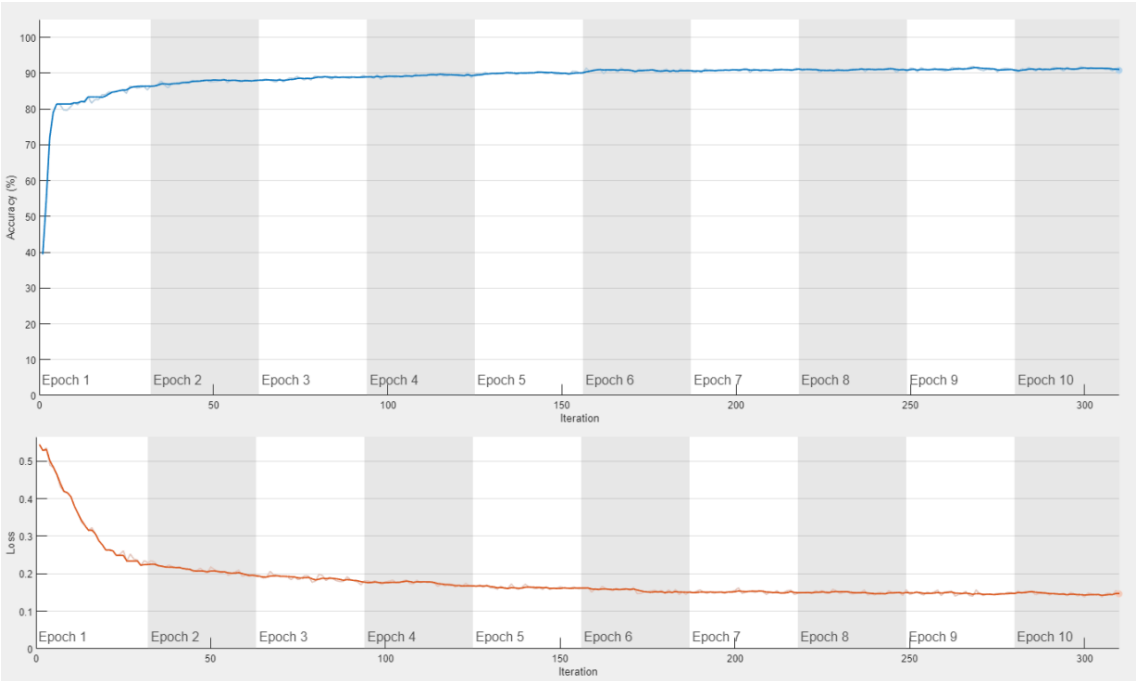

Figure 4.6: Convergence with actual masks.

Model	LOSS	DA	TRUE_10	TRUE_Full
RN101	$L_{GD}3.1$	DA1	0.590	0.776
RN101	$L_{GD}3.1$	DA2	0.711	0.826
ERN101(10)	$L_{GD}3.1$	DA1	0.598	0.779
ERN101(10)	$L_{GD}3.1$	DA2	0.719	0.831
ELoss101(10)	Many Loss	DA1	0.654	0.785
ELoss101(10)	Many Loss	DA2	0.729	0.852
ELoss101_2(10)	Many Loss	DA1	0.656	0.800
ELoss101_2(10)	Many Loss	DA2	0.731	0.862
ELossMix(10)	Many Loss	DA1/DA2	0.719	0.833
ELossMix_2(10)	Many Loss	DA1/DA2	0.719	0.846
Unet [73]				0.765
Feature Pyramid Network [73]				0.829
Unet++ [73]				0.834

Table 4.3: Outcomes achieved with various ensembles.

2. ERN101(N) is a group of N RN101 models combined as an ensemble.

3. ELoss101(10) is made up of 10 RN101 models, each utilizing DA1 as the data augmentation technique, but with a distinct loss function. Specifically, the ensemble is formulated as:

$$2 \times L_{GD} + 2 \times L_T + 2 \times Comb_1 + 2 \times Comb_2 + 2 \times Comb_3$$

For example, the term $2 \times L_{GD}$ indicates two RN101 networks created by utilizing Generalized Dice Loss.

4. ELossMix(10) is an ensemble comprising 10 models created by utilizing the same five loss functions as ELoss101(10). However, this time, for each model associated with a specific loss function, two training sets are computed, one utilizing DA1 and the other employing DA2.

5. ELoss101_2(10) is similar to ELoss101, with the sole difference being the utilization of LDiceBES instead of L_T.

6. ELossMix_2(10) is comparable to ELoss101, except for the fact that LDiceBES is employed instead of L_T.

The objective here is to ascertain the effectiveness of ensembles when applied to small datasets, such as the one used in our study.

The outcomes presented in Table 4.3 indicate that the quality of segmentation tasks is significantly influenced by the training dataset. In our tests, we observed that enlarging the training sample size led to a more reliable and impartial model.

Substituting L_T with LDiceBES in ELoss101_2(10) and ELossMix_2(10) enhanced the outcomes, as shown in [68]. In addition, using the DA2 data augmentation technique resulted in a performance improvement over the DA1 method. This is especially noticeable when comparing the outcomes of ELoss101(10) with those of ELossMix(10). In various scenarios, including polyp segmentation, skin segmentation, leukocyte identification, butterfly and micro-organism identification, as reported in [68], ELossMix(10) outperformed ELoss101(10). One potential explanation for this difference in performance could be the relatively small size of the VinDr-RibCXR dataset used in this study. As part of our future research, we intend to explore this aspect further by utilizing our models for a variety of applications.
Upon comparing our models with those presented in [73] - specifically U-Net, Feature Pyramid Network (FPN), and U-Net++ [73] - we can see that ELoss1012(10) combined with DA2 achieved the highest score for both the reduced and full sample sizes.

4.3.3 Other Segmentation Applications

Given the objective of this thesis to investigate techniques for enhancing the diversity of ensembles, we present the outcomes of several baseline classifiers and ensembles using various network architectures in Table 4.4. All of these ensembles were combined with the DA1 data augmentation technique, as outlined in section 3.3.1. All of the tests presented were conducted solely utilizing the Dice loss. Additionally, due to space constraints, we only provide the average performance value for the polyp dataset among the group of datasets. Every ensemble is created by combining N models (N=1 indicates a standalone model), which are distinguished only by the variation in the randomization of the training process:

1. RN18 is a single DeepLabV3+ segmentator, which incorporates Resnet18 as the backbone (pretrained in ImageNet);

2. ERN18(N) is a collection of N RN18 networks, all pretrained in ImageNet, which are combined as an ensemble;

3. RN50 is a solitary DeepLabV3+ segmentator that uses Resnet50 as the backbone (pretrained in ImageNet);

4. ERN50(N) is a group of N RN50 networks combined as an ensemble;

5. RN101, as outlined in the previous section (4.3.3);

6. ERN101(N), as outlined in the previous section (4.3.3);

The outcomes presented in Table 4.4 reveal that even though the overall performance of the models increases when transitioning from the standalone variant to an ensemble, the gain is not

	Polyp	**Leuko**	**BFly**	**EMicro**	***Avg***
RN18	0.806	0.897	0.960	0.908	0.887
RN50	0.802	0.895	0.968	0.909	0.889
RN101	0.808	0.915	0.976	0.918	0.898
ERN18	0.821	0.913	0.963	0.913	0.895
ERN50	0.807	0.897	0.969	0.918	0.893
ERN101	**0.834**	**0.925**	**0.978**	**0.919**	**0.907**

Table 4.4: The Dice score achieved by the suggested ensembles in the five benchmark datasets is presented in the table, with the last column (AVG) showing the mean performance across all the datasets.

as significant as anticipated, implying that the individual techniques are already quite robust. It is possible that this result is related to the design of the DeepLabV3+ network. The network's internal modules employ atrous convolutions either in series or in parallel to capture multi-scale context by utilizing multiple atrous rates. The proposed solution, intended to address the issue of segmenting objects at varying scales, mimics an ensemble approach by combining activations obtained at different levels of the encoder. As a result, the segmentation obtained is very stable. The most effective approach is to employ ResNet101 as the backbone.

Ablation studies

The initial ablation study pertains to assessing various loss functions to enhance the diversity of the models and augment the ensemble's performance. Table 4.5 displays the results of RN101 utilizing the various tested/proposed loss functions and compares them to the dice loss as the baseline, and DA1 as the data augmentation method. To save space, we only present the average performance of the polyp and skin datasets among the dataset set. Next, the stand-alone networks are merged into ensembles, always employing the sum rule:

1. ELoss101(10) as detailed in the above section 4.3.3.

2. ELossMix(10) as detailed in the above section 4.3.3.

3. ELossLarge(10) is a collection of ten networks. The networks trained using (L_{GD}, L_{T}, Comb1, Comb2, Comb3) employ DA2 as the augmented training set, whereas the networks trained utilizing the novel loss functions assessed in this thesis (L_{STR}, $L_{BoundExpS}$, $L_{DiceBES}$, L_{MS}, L_{CS}) are combined with DA1.

The results shown in Table 4.5 demonstrate that the newly proposed loss functions achieve performance similar to the Dice loss function and can be viewed as a valuable starting point for creating an ensemble. The findings indicate that combining networks trained using diverse loss functions is a successful approach for creating an ensemble, as evidenced by the superior

	LOSS	Polyp	Skin	Leuko	BFly	EMicro	*Avg*
RN101	L_{GD}3.1	0.808	0.871	0.915	0.976	0.918	0.898
RN101	L_{STR}3.20	0.809	0.869	0.930	0.964	0.901	0.895
RN101	$L_{BoundExpS}$3.21	0.803	0.874	0.928	0.978	0.901	0.897
RN101	$L_{DiceBES}$3.24	0.819	0.869	0.922	0.969	0.904	0.897
RN101	L_{MS}	0.813	0.860	0.920	0.972	0.920	0.897
RN101	L_{CS}3.27	0.823	0.873	0.917	0.967	0.911	0.898
ERN101	L_{GD}3.1	0.834	0.878	0.925	0.978	0.919	0.907
ELoss101	Many loss	0.842	0.880	0.925	0.980	0.921	0.910
ELossMix	Many loss	**0.851**	**0.883**	0.936	0.983	**0.924**	0.915
ELossLarge	Many loss	0.848	**0.883**	**0.944**	**0.984**	0.922	**0.916**

Table 4.5: The table presents the performance (Dice) of some stand-alone methods and ensembles in the five benchmark datasets, with variations in the loss function. The last column, AVG, represents the average performance.

performance of ELoss101 and ELossLarge in comparison to ERN101(10). This outcome becomes even more apparent when the approach is employed for dissimilar problems.

It's important not to underestimate the importance of modifying the training set, as demonstrated by the successful combination of various data augmentations and loss functions in the ensemble labeled as ELossMix. It's worth noting that ELossMix and ELossLarge show comparable performance.

The second ablation study focuses on assessing various architectures, and Table 4.6 presents the results of evaluating the performance of the previously mentioned approaches in conjunction with different data augmentation techniques. The labels "DA1" and "DA2" correspond to the techniques outlined in section 3.3. "DA1/2" signifies that the ensemble was created by combining networks trained using both DA1 and DA2. The training of HardNet-MSEG involves the use of two distinct optimizers, namely SGD (referred to as H_S) and Adam (referred to as H_A). The ensemble labeled FH is formed by combining multiple instances of HardNet-MSEG that were trained using different optimizers. The original paper that introduced PVT recommends training the model using the AdamW optimizer, which is the approach adopted for this study. The loss function used for training both HarDNet-MSEG and PVT is identical to that described in their respective original papers.

Table 4.6 presents additional ensembles:

1. PVT(2) refers to an ensemble created by combining two instances of PVT. The "sum rule" approach is employed to combine PVT trained with DA1 and PVT trained with DA2;

2. FH(2) ensemble is composed by two instances of H_S (one trained with DA1 and the other with DA2) and two instances of H_A (one trained with DA1 and the other with DA2). This are combined using the sum rule approach;

	DA	Polyp	Leuko	BFly	EMicro	*Avg*
ELossMix(10)	DA1/2	0.851	0.936	0.983	0.924	0.924
H_A	DA1	0.840	0.923	0.977	0.914	0.914
H_A	DA2	0.854	0.945	0.982	0.912	0.923
H_S	DA1	0.816	0.889	0.969	0.894	0.892
H_S	DA2	0.847	0.917	0.976	0.901	0.910
FH	DA1	0.859	0.913	0.980	0.915	0.917
FH(2)	DA1/2	0.862	0.934	0.982	0.916	0.924
PVT	DA1	0.854	0.954	0.975	0.920	0.926
PVT	DA2	0.855	0.954	0.984	0.919	0.928
PVT(2)	DA1/2	0.855	0.957	0.984	0.922	0.930
FH(2)+2×PVT(2)	DA1/2	0.875	0.955	0.985	0.924	0.935
E10_FH2_PVT2	DA1/2	0.875	0.953	0.985	0.926	0.935

Table 4.6: The performance (measured by Dice score) of individual methods and ensembles across five benchmark datasets is reported.

3. FH(2)+2xPVT(2) is formed by applying the weighted sum rule to the FH(2) and PVT(2) ensembles. The weight assigned to PVT(2) is such that its contribution to the ensemble is equivalent to that of FH(2), taking into account the fact that FH(2) includes four networks while PVT(2) comprises only two.

4. E10_FH2_PVT2 = ELossMix(10)+(10/4)×FH(2)+(10/2)×PVT(2), is created by applying the weighted sum rule to the ELossMix(10), FH(2), and PVT(2) ensembles. The weights assigned to each ensemble member ensure that they contribute equally to the overall ensemble, noting that ELossMix(10) is formed by applying the sum rule to 10 instances of DeepLabV3+.

Based on the results presented in Table 4.6, the following conclusions can be made:

1. PVT(2), FH(2), and ElossMix(10) demonstrate comparable performance, with the exception of the Leuko dataset, where PVT(2) outperforms both FH(2) and ElossMix(10);

2. PVT(2) yields only a marginal improvement over the performance of the standalone PVT model. Similarly, the performance gain achieved by FH(2) over the best performing standalone HardNet model (i.e., H_A combined with DA2) is also modest;

3. The optimal performance is achieved by combining multiple architectures, with the "***FH(2)+2×PVT(2)***" ensemble striking the ideal balance between complexity and performance.

Comparison with the literature

Method	Kvasir		ClinicalDB		ColonDB	
	IoU	Dice	IoU	Dice	IoU	Dice
FH(2)+2×PVT(2)	**0.874**	**0.920**	0.894	0.937	**0.751**	**0.826**
ensemble in [69]	0.871	0.917	0.886	0.931	0.697	0.769
HarDNet-MSEG [44]	0.857	0.912	0.882	0.932	0.66	0.731
PraNet [44]	0.84	0.898	0.849	0.899	0.64	0.709
SFA[44]	0.611	0.723	0.607	0.700	0.347	0.469
U-Net++ [44]	0.743	0.821	0.729	0.794	0.41	0.483
U-Net [44]	0.746	0.818	0.755	0.823	0.444	0.512
SETR [122]	0.854	0.911	0.885	0.934	0.69	0.773
TransUnet [18]	0.857	0.913	0.887	0.935	0.699	0.781
TransFuse [117]	0.870	**0.920**	**0.897**	**0.942**	0.706	0.781
UACANet [55]	0.859	0.912	0.88	0.926	0.678	0.751
SANet [107]	0.847	0.904	0.859	0.916	0.670	0.753
MSNet [120]	0.862	0.907	0.879	0.921	0.678	0.755
PVT [32]	0.864	0.917	0.889	0.937	0.727	0.808
SwinE-Net [76]	0.870	**0.920**	0.892	0.938	0.725	0.804
AMNet [94]	0.865	0.912	0.888	0.936	0.690	0.762

Table 4.7: The performance of various models (measured by Dice and IoU scores) on the Kvasir, ClinicalDB, and ColonDB datasets for polyp segmentation is reported.

To facilitate comparison with other methods in the literature, we provide a comprehensive evaluation of our top-performing ensembles across the various datasets for polyp segmentation. The results are presented in full in Table 4.7 and Table 4.8.

The creators of the LEUKO dataset reported an IoU score of 0.842, while the ensemble FH(2)+2xPVT(2) achieves a higher IoU of 0.916.

The authors of the EMicro dataset reported a Dice score of 0.884, while the ensemble FH(2)+2xPVT(2) achieves a higher Dice score of 0.924.

Several approaches have been evaluated for the BFLY dataset (refer to [39]), and the two most effective methods reported in prior research are:

1. According to the findings presented in [39], an IoU score of 0.950 was achieved;

2. According to the findings presented in [97], an IoU score of 0.945 was achieved.

The proposed ensemble of this thesis, FH(2)+2xPVT(2), surpasses the previous state-of-the-art methods, achieving an IoU score of 0.970.

Undoubtedly, the ensemble enhances the performance of the best performing standalone network (i.e., PVT combined with DA2). However, a notable drawback of this approach is that the optimal ensemble consists of six distinct networks, which entails six times the RAM requirements and six times the inference time compared to a single network. Moreover, despite the use of an ensemble, the inference time remains quite fast, and with the current GPU

Method	ETIS		CVC−T		Average	
	IoU	Dice	IoU	Dice	IoU	Dice
FH(2)+2×PVT(2)	**0.717**	**0.787**	0.842	0.904	**0.816**	**0.875**
ensemble in [69]	0.663	0.740	0.829	0.901	0.790	0.852
HarDNet-MSEG [44]	0.613	0.677	0.821	0.887	0.767	0.828
PraNet [44]	0.567	0.628	0.797	0.871	0.739	0.801
SFA[44]	0.217	0.297	0.329	0.467	0.422	0.531
U-Net++ [44]	0.344	0.401	0.624	0.707	0.570	0.641
U-Net [44]	0.335	0.398	0.627	0.710	0.581	0.652
SETR [122]	0.646	0.726	0.814	0.889	0.778	0.847
TransUnet [18]	0.66	0.731	0.824	0.893	0.785	0.851
TransFuse [117]	0.663	0.737	0.826	0.894	0.792	0.855
UACANet [55]	0.678	0.751	**0.849**	**0.910**	0.789	0.850
SANet [107]	0.654	0.750	0.815	0.888	0.769	0.842
MSNet [120]	0.664	0.719	0.807	0.869	0.778	0.834
PVT [32]	0.706	**0.787**	0.833	0.900	0.804	0.869
SwinE-Net [76]	0.687	0.758	0.842	0.906	0.803	0.865
AMNet [94]	0.679	0.756	-	-	-	-

Table 4.8: The Dice and IoU scores for polyp segmentation on the ETIS and CVC-T datasets are reported, and the overall average across all five datasets (Kvasir, ClinicalDB, ColonDB, ETIS, and CVC-T) is also presented.

architectures, it does not pose a challenge in most cases. While it could be a concern in certain applications, such as autonomous driving, it is not a significant issue in the context of the segmentation problems addressed in this thesis. For instance, a solitary network of HarDNet-MSEG can achieve a processing speed of 86.7 images per second when executed on a GeForce RTX 2080 Ti GPU.

We carried out an extra experiment to identify the best possible set of models to include in the final ensemble. To accomplish this, we set aside a validation set and focused our analysis solely on the two problem domains that comprised multiple test sets, namely polyp and skin segmentation. To validate our results for the polyp segmentation problem, we utilized the Kvasir test set, and for the skin segmentation problem, we utilized the ECU test set. To identify the most effective subset of networks in terms of the Dice performance metric on the validation set, we employed sequential forward floating selection (SFFS) [79]. The performance of both ensembles did not meet our expectations, and in both datasets, our top-performing approach (i.e., FH(2) + 2xPVT(2)) outperformed them. We have encountered an overfitting issue in both instances where the test images vary greatly from one another. Thus, to ensure the selection of a dependable network, a more extensive validation set is necessary, along with a wider range of diverse variations that can potentially occur in an image.

In conclusion, we conduct Q-statistic tests to further verify our concept of constructing ensembles. To demonstrate the level of diversity among the networks in the ensemble, Yule's

Q-statistic [58] was employed. The Q-statistic ranges from -1 to 1 after computation, with a value of zero indicating statistically independent classifiers.

Table 4.9: Average Q-statistic.

Ensembles	Average Q-Statistic
ERN101(10)	0.975
ELOSS101(10)	0.952
ELOSSMIX(10)	0.921
FH(2) + 2 × PVT(2)	0.925

Chapter 5

Conclusion

In computer vision, the task of classifying each pixel in an image is referred to as semantic segmentation.
Semantic segmentation is a crucial task in various fields, such as autonomous vehicles, where it enables the identification of surrounding objects, and in medical diagnosis, where it enhances the early detection of potentially harmful pathologies and reduces the risk of severe consequences.
In this study, we achieve state-of-the-art performance by proposing various segmentation approach ensembles. Our evaluation includes:

1. Experimenting with different loss functions;
2. Implementing various data augmentation techniques;
3. Utilizing different network topologies, including convolutional neural networks and transformers (such as DeepLabV3+, HarDNet-MSEG, and Pyramid Vision Transformers);

The ensemble is ultimately aggregated using the sum rule.
We evaluated our proposed ensemble on six benchmark datasets, which include polyp detection, skin detection, leukocyte recognition, environmental microorganism detection, butterfly recognition, and radiology segmentation, and achieved state-of-the-art results.
Our experiments on rib semantic segmentation also revealed that deep networks for semantic segmentation, as observed in [115] for image classification, have the ability to learn random masks quite well.
The fact that deep networks still hold some level of ambiguity is supported by this evidence. Furthermore, there exists a positive correlation between sample size and segmentation performance.
In our future work, we plan to reduce the complexity of ensembles through methods such as pruning, quantization, low-ranking factorization, and distillation.

List of Figures

3.1 DeepLabv3+ [21] employs an encoder-decoder structure to enhance the capabilities of DeepLabv3. The encoder module uses atrous convolution at multiple scales to encode multi-scale contextual information, while the decoder module, which is simple yet effective, refines the segmentation results along the boundaries of objects. 14
3.2 The Pyramid Vision Transformer (PVT) [31] has an architecture consisting of four stages, each containing a patch embedding layer and a Li-layer Transformer encoder. These stages follow a pyramid structure, with the output resolution progressively decreasing from high (4-stride) to low (32-stride). 15
3.3 Original image and label. 24
3.4 Application of DA1 to images. 24
3.5 Original image and label. 25
3.6 Application of DA2 to images. 26
3.7 Application of DA2 to labels. 26
3.8 Application of the jpeg data augmentation technique. 29

4.1 Examples of images from the VinDr-RibCXR dataset together with ground truth masks adopted in our experiments. 37
4.2 The figures (a) to (o) show examples of the various types of polyps present in each colonoscopy video of the COLON-DB database. The polyps are highlighted by blue contours, corresponding to video 1 to video 15 respectively. [9] 39
4.3 Here is an example of how the learning/testing ETIS database [92] is created from the primary data in [9]. 40
4.4 The provided samples include images and corresponding masks for polyp segmentation, skin segmentation, leukocyte identification, butterfly recognition, and microorganism identification [67]. 42
4.5 Convergence with random masks. 45
4.6 Convergence with actual masks. 45

List of Tables

4.1 Datasets for Skin Segmentation. ECU dataset is split in 2000 images for training and 2000 for test set. For ECU, we considered the subset of images that were not used in the training phase. 34
4.2 Performance (Dice=F1-score) in the skin detection problem. Best performance in bold. 42
4.3 Outcomes achieved with various ensembles. 46
4.4 The Dice score achieved by the suggested ensembles in the five benchmark datasets is presented in the table, with the last column (AVG) showing the mean performance across all the datasets. 48
4.5 The table presents the performance (Dice) of some stand-alone methods and ensembles in the five benchmark datasets, with variations in the loss function. The last column, AVG, represents the average performance. 49
4.6 The performance (measured by Dice score) of individual methods and ensembles across five benchmark datasets is reported. 50
4.7 The performance of various models (measured by Dice and IoU scores) on the Kvasir, ClinicalDB, and ColonDB datasets for polyp segmentation is reported. . 51
4.8 The Dice and IoU scores for polyp segmentation on the ETIS and CVC-T datasets are reported, and the overall average across all five datasets (Kvasir, ClinicalDB, ColonDB, ETIS, and CVC-T) is also presented. 52
4.9 Average Q-statistic. 53

Bibliography

[1] Abdallah S Abdallah, Mohamad Abou El-Nasr, and A Lynn Abbott. "A new color image database for benchmarking of automatic face detection and human skin segmentation techniques". In: *Proceedings of World Academy of Science, Engineering and Technology*. Vol. 20. Citeseer. 2007, pp. 353–357.

[2] Muhammad Arsalan et al. "OR-Skip-Net: Outer residual skip network for skin segmentation in non-ideal situations". In: *Expert Systems with Applications* 141 (2020), p. 112922.

[3] Yuri Sousa Aurelio et al. "Learning from imbalanced data sets with weighted cross-entropy function". In: *Neural processing letters* 50.2 (2019), pp. 1937–1949.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[6] Federico Baldassarre, Diego González Morın, and Lucas Rodés-Guirao. "Deep koalarization: Image colorization using cnns and inception-resnet-v2". In: *arXiv preprint arXiv:1712.03400* (2017).

[7] Indranil Balki et al. "Sample-Size Determination Methodologies for Machine Learning in Medical Imaging Research: A Systematic Review". In: *Canadian Association of Radiologists Journal* 70.4 (2019), pp. 344–353. DOI: 10.1016/j.carj.2019.06.002.

[8] Claudia Beleites et al. "Sample size planning for classification models". In: *Analytica Chimica Acta* 760 (2013), pp. 25–33. ISSN: 0003-2670. DOI: 10.1016/j.aca.2012.11.007.

[9] Jorge Bernal, Javier Sánchez, and Fernando Vilarino. "Towards automatic polyp detection with a polyp appearance model". In: *Pattern Recognition* 45.9 (2012), pp. 3166–3182.

[10] Jorge Bernal et al. “WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians”. In: *Computerized medical imaging and graphics* 43 (2015), pp. 99–111.

[11] Patrick Brandao et al. “Towards a computed-aided diagnosis system in colonoscopy: automatic polyp segmentation using convolution neural networks”. In: *Journal of Medical Robotics Research* 3.02 (2018), p. 1840002.

[12] Patrick Brandao et al. “Towards a computed-aided diagnosis system in colonoscopy: automatic polyp segmentation using convolution neural networks”. In: *Journal of Medical Robotics Research* 3.02 (2018), p. 1840002.

[13] Joseph Bullock, Carolina Cuesta-Lázaro, and Arnau Quera-Bofarull. “XNet: A convolutional neural network (CNN) implementation for medical X-ray image segmentation suitable for small datasets”. In: *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*. Vol. 10953. SPIE. 2019, pp. 453–463.

[14] Joao Paulo Brognoni Casati, Diego Rafael Moraes, and Evandro Luis Linhari Rodrigues. “SFA: A human skin image database based on FERET and AR facial images”. In: *IX workshop de Visao Computational, Rio de Janeiro*. 2013.

[15] Isabella Castiglioni et al. “AI applications to medical images: From machine learning to deep learning”. In: *Physica Medica* 83 (2021), pp. 9–24. ISSN: 1120-1797. DOI: 10.1016/j.ejmp.2021.02.006.

[16] Ping Chao et al. “Hardnet: A low memory traffic network”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 3552–3561.

[17] Ping Chao et al. “Hardnet: A low memory traffic network”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 3552–3561.

[18] Jieneng Chen et al. “Transunet: Transformers make strong encoders for medical image segmentation”. In: *arXiv preprint arXiv:2102.04306* (2021).

[19] Liang-Chieh Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

[20] Liang-Chieh Chen et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[21] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 833–851. ISBN: 978-3-030-01234-2. DOI: 10.1007/978-3-030-01234-2_49.

[22] Liang-Chieh Chen et al. “Encoder-decoder with atrous separable convolution for semantic image segmentation”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

[23] Siwei Chen, Gregor Urban, and Pierre Baldi. “Weakly Supervised Polyp Segmentation in Colonoscopy Images Using Deep Neural Networks”. In: *Journal of Imaging* 8.5 (2022), p. 121.

[24] Zixuan Chen et al. “Contour-aware loss: Boundary-aware learning for salient object segmentation”. In: *IEEE Transactions on Image Processing* 30 (2020), pp. 431–443.

[25] Junghwan Cho et al. “How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?” In: *arXiv preprint arXiv:1511.06348* (2015).

[26] Yeong-Jun Cho. “Weighted Intersection over Union (wIoU): A New Evaluation Metric for Image Segmentation”. In: *arXiv preprint arXiv:2107.09858* (2021).

[27] Cristina Cornelio et al. “Voting with random classifiers (VORACE): theoretical and experimental analysis”. In: *Autonomous Agents and Multi-Agent Systems* 35.2 (2021), p. 22. ISSN: 1573-7454. DOI: 10.1007/s10458-021-09504-y. URL: https://doi.org/10.1007/s10458-021-09504-y.

[28] Daniela Cuza et al. “Deep semantic segmentation in skin detection”. In: *ESANN 2022 proceedings European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 2022.

[29] Djamila Dahmani, Mehdi Cheref, and Slimane Larabi. “Zero-sum game theory model for segmenting skin regions”. In: *Image and Vision Computing* 99 (2020), p. 103925.

[30] Mehwish Dildar et al. “Skin Cancer Detection: A Review Using Deep Learning Techniques”. In: *International Journal of Environmental Research and Public Health* 18.10 (2021). ISSN: 1660-4601. DOI: 10.3390/ijerph18105479. URL: https://www.mdpi.com/1660-4601/18/10/5479.

[31] Bo Dong et al. “Polyp-PVT: Polyp Segmentation with Pyramid Vision Transformers”. In: arXiv, 2021. DOI: 10.48550/ARXIV.2108.06932. URL: https://arxiv.org/abs/2108.06932.

[32] Bo Dong et al. “Polyp-pvt: Polyp segmentation with pyramid vision transformers”. In: *arXiv preprint arXiv:2108.06932* (2021).

[33] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[34] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[35] Yingtao Fang et al. “The impact of training sample size on deep learning-based organ auto-segmentation for head-and-neck patients”. In: *Physics in Medicine & Biology* 66.18 (Sept. 2021), p. 185012. DOI: 10.1088/1361-6560/ac2206.

[36] Di Feng et al. “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1341–1360.

[37] Di Feng et al. “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1341–1360.

[38] Rosa L Figueroa et al. “Predicting sample size required for classification performance”. In: *BMC medical informatics and decision making* 12.1 (2012), pp. 1–10. DOI: 10.1186/1472-6947-12-8.

[39] Idir Filali et al. “Graph ranking based butterfly segmentation in ecological images”. In: *Ecological Informatics* 68 (2022), p. 101553.

[40] Jhony H Giraldo et al. “Hypergraph convolutional networks for weakly-supervised semantic segmentation”. In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 16–20.

[41] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[42] Yun Bo Guo and Bogdan Matuszewski. “Giana polyp segmentation with fully convolutional dilation neural networks”. In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS-Science and Technology Publications. 2019, pp. 632–641.

[43] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[44] CH Huang, HY Wu, and Y-L HarDNet-MSEG Lin. “A Simple Encoder-Decoder Polyp Segmentation Neural Network that Achieves over 0.9 Mean Dice and 86 FPS. arXiv 2021”. In: *arXiv preprint arXiv:2101.07172* ().

[45] Chien-Hsiang Huang, Hung-Yu Wu, and Youn-Long Lin. “Hardnet-mseg: A simple encoder-decoder polyp segmentation neural network that achieves over 0.9 mean dice and 86 fps”. In: *arXiv preprint arXiv:2101.07172* (2021).

[46] Chien-Hsiang Huang, Hung-Yu Wu, and Youn-Long Lin. "Hardnet-mseg: A simple encoder-decoder polyp segmentation neural network that achieves over 0.9 mean dice and 86 fps". In: *arXiv preprint arXiv:2101.07172* (2021).

[47] Lei Huang et al. "Human skin detection in images by MSER analysis". In: *18th IEEE International Conference on Image Processing, ICIP 2011, Brussels, Belgium, September 11-14, 2011*. Ed. by Benoıt Macq and Peter Schelkens. IEEE, 2011, pp. 1257–1260. DOI: 10.1109/ICIP.2011.6115661. URL: https://doi.org/10.1109/ICIP.2011.6115661.

[48] Fabian Isensee et al. "nnU-Net for brain tumor segmentation". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II 6*. Springer. 2021, pp. 118–132.

[49] Debesh Jha et al. "Real-time polyp detection, localization and segmentation in colonoscopy using deep learning". In: *Ieee Access* 9 (2021), pp. 40496–40510.

[50] Debesh Jha et al. "Real-time polyp detection, localization and segmentation in colonoscopy using deep learning". In: *Ieee Access* 9 (2021), pp. 40496–40510.

[51] Michael J. Jones and James M. Rehg. "Statistical Color Models with Application to Skin Detection". In: *Int. J. Comput. Vis.* 46.1 (2002), pp. 81–96. DOI: 10.1023/A:1013200319198. URL: https://doi.org/10.1023/A:1013200319198.

[52] Michal Kawulok et al. "Self-adaptive algorithm for segmenting skin regions". In: *EURASIP Journal on Advances in Signal Processing* 2014.1 (2014), pp. 1–22.

[53] Salman Khan et al. "Transformers in vision: A survey". In: *ACM computing surveys (CSUR)* 54.10s (2022), pp. 1–41.

[54] Zia Khan et al. "Evaluation of deep neural networks for semantic segmentation of prostate in T2W MRI". In: *Sensors* 20.11 (2020), p. 3183.

[55] Taehun Kim, Hyemin Lee, and Daijin Kim. "Uacanet: Uncertainty augmented context attention for polyp segmentation". In: *Proceedings of the 29th ACM International Conference on Multimedia*. 2021, pp. 2167–2175.

[56] Taehun Kim, Hyemin Lee, and Daijin Kim. "Uacanet: Uncertainty augmented context attention for polyp segmentation". In: *Proceedings of the 29th ACM International Conference on Multimedia*. 2021, pp. 2167–2175.

[57] Alexander Kirillov et al. *Segment Anything*. 2023. arXiv: 2304.02643 [cs.CV].

[58] Ludmila I Kuncheva and Christopher J Whitaker. "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy". In: *Machine learning* 51.2 (2003), p. 181.

[59] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[60] Songtao Liu, Di Huang, et al. “Receptive field block net for accurate and fast object detection”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 385–400.

[61] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[62] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[63] Alessandra Lumini and Loris Nanni. “Fair comparison of skin detection approaches on publicly available datasets”. In: *Expert Systems with Applications* 160 (2020), p. 113677.

[64] Shervin Minaee et al. “Image segmentation using deep learning: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[65] Sajaa G Mohammed et al. “Image Segmentation for Skin Detection”. In: *Journal of Southwest Jiaotong University* 55.1 (2020).

[66] Yuichi Mori et al. “Computer-aided diagnosis for colonoscopy”. In: *Endoscopy* 49.08 (2017), pp. 813–819.

[67] Loris Nanni et al. “An Empirical Study on Ensemble of Segmentation Approaches”. In: *Signals* 3.2 (2022), pp. 341–358.

[68] Loris Nanni et al. “An Empirical Study on Ensemble of Segmentation Approaches”. In: *Signals* 3.2 (2022), pp. 341–358.

[69] Loris Nanni et al. “Deep ensembles in bioimage segmentation”. In: *arXiv preprint arXiv:2112.12955* (2021).

[70] Loris Nanni et al. *Feature transforms for image data augmentation*. 2022. DOI: 10.48550/ARXIV.2201.09700. URL: https://arxiv.org/abs/2201.09700.

[71] Loris Nanni et al. “Polyp Segmentation with Deep Ensembles and Data Augmentation”. In: *Artificial Intelligence and Machine Learning for Healthcare: Vol. 1: Image and Data Analytics*. Springer, 2022, pp. 133–153.

[72] Ponnada A. Narayana et al. “Deep-Learning-Based Neural Tissue Segmentation of MRI in Multiple Sclerosis: Effect of Training Set Size”. In: *Journal of Magnetic Resonance Imaging* 51.5 (2020), pp. 1487–1496. DOI: 10.1002/jmri.26959.

[73] Hoang C Nguyen et al. “VinDr-RibCXR: A Benchmark Dataset for Automatic Segmentation and Labeling of Individual Ribs on Chest X-rays”. In: *arXiv preprint arXiv:2107.01327* (2021).

[74] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.

[75] Marco Paracchini et al. “Deep skin detection on low resolution grayscale images”. In: *Pattern Recognition Letters* 131 (2020), pp. 322–328.

[76] Kyeong-Beom Park and Jae Yeol Lee. “SwinE-Net: hybrid deep learning approach to novel polyp segmentation using convolutional neural network and Swin Transformer”. In: *Journal of Computational Design and Engineering* 9.2 (2022), pp. 616–632.

[77] Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai. “Skin segmentation using color pixel classification: analysis and comparison”. In: *IEEE transactions on pattern analysis and machine intelligence* 27.1 (2005), pp. 148–154.

[78] Bharath Srinivas Prabakaran, Erik Ostrowski, and Muhammad Shafique. “BoundaryCAM: A Boundary-based Refinement Framework for Weakly Supervised Semantic Segmentation of Medical Images”. In: *arXiv preprint arXiv:2303.07853* (2023).

[79] Pavel Pudil, Jana Novovičová, and Josef Kittler. “Floating search methods in feature selection”. In: *Pattern recognition letters* 15.11 (1994), pp. 1119–1125.

[80] Xuebin Qin et al. “Basnet: Boundary-aware salient object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 7479–7489.

[81] Md Atiqur Rahman and Yang Wang. “Optimizing intersection-over-union in deep neural networks for image segmentation”. In: *International symposium on visual computing*. Springer. 2016, pp. 234–244.

[82] M Roy Reena and PM Ameer. “Localization and recognition of leukocytes in peripheral blood: a deep learning approach”. In: *Computers in Biology and Medicine* 126 (2020), p. 104034.

[83] Ariel Rokem, Yue Wu, and Aaron Lee. “Assessment of the need for separate test set and number of medical images necessary for deep learning: a sub-sampling study”. In: *bioRxiv* (2017), p. 196659.

[84] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[85] Javier Ruiz-del-Solar and Rodrigo Verschae. "Skin Detection using Neighborhood Information". In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*. IEEE Computer Society, 2004, pp. 463–468. DOI: 10.1109/AFGR.2004.1301576. URL: https://doi.org/10.1109/AFGR.2004.1301576.

[86] Khawla Ben Salah, Mohamed Othmani, and Monji Kherallah. "A novel approach for human skin detection using convolutional neural network". In: *The Visual Computer* 38.5 (2022), pp. 1833–1843.

[87] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. "Tversky loss function for image segmentation using 3D fully convolutional deep networks". In: *International workshop on machine learning in medical imaging*. Springer. 2017, pp. 379–387.

[88] Juan C Sanmiguel and Sergio Suja. "Skin detection by dual maximization of detectors agreement for video monitoring". In: *Pattern Recognition Letters* 34.16 (2013), pp. 2102–2109.

[89] Stephen J Schmugge et al. "Objective evaluation of approaches of skin detection using ROC analysis". In: *Computer vision and image understanding* 108.1-2 (2007), pp. 41–51.

[90] Li Shao et al. "Determination of Minimum Training Sample Size for Microarray-Based Cancer Outcome Prediction–An Empirical Assessment". In: *PLOS ONE* 8.7 (July 2013), pp. 1–9. DOI: 10.1371/journal.pone.0068579.

[91] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48. DOI: 10.1186/s40537-019-0197-0.

[92] Juan Silva et al. "Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer". In: *International journal of computer assisted radiology and surgery* 9 (2014), pp. 283–293.

[93] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[94] Pengfei Song, Jinjiang Li, and Hui Fan. "Attention based multi-scale parallel network for polyp segmentation". In: *Computers in Biology and Medicine* 146 (2022), p. 105476.

[95] Carole H Sudre et al. "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations". In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248.

[96] Wei Ren Tan et al. "A Fusion Approach for Efficient Human Skin Detection". In: *IEEE Trans. Ind. Informatics* 8.1 (2012), pp. 138–147. DOI: 10.1109/TII.2011.2172451. URL: https://doi.org/10.1109/TII.2011.2172451.

[97] Hui Tang, Bin Wang, and Xin Chen. "Deep learning techniques for automatic butterfly segmentation in ecological images". In: *Computers and Electronics in Agriculture* 178 (2020), p. 105739.

[98] Tomasz Tarasiewicz, Jakub Nalepa, and Michal Kawulok. "Skinny: A lightweight U-net for skin detection and segmentation". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 2386–2390.

[99] Vajira Thambawita et al. "The medico-task 2018: Disease detection in the gastrointestinal tract using global features and deep learning". In: *arXiv preprint arXiv:1810.13278* (2018).

[100] Anirudh Topiwala et al. "Adaptation and evaluation of deep learning techniques for skin segmentation on novel abdominal dataset". In: *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE. 2019, pp. 752–759.

[101] David Vázquez et al. "A benchmark for endoluminal scene segmentation of colonoscopy images". In: *Journal of healthcare engineering* 2017 (2017).

[102] Giuseppe Vecchio et al. "MASK-RL: Multiagent video object segmentation framework through reinforcement learning". In: *IEEE transactions on neural networks and learning systems* 31.12 (2020), pp. 5103–5115.

[103] Josiah Wang, Katja Markert, Mark Everingham, et al. "Learning Models for Object Recognition from Natural Language Descriptions." In: *BMVC*. Vol. 1. 2009. Citeseer. 2009, p. 2.

[104] Pu Wang et al. "Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy". In: *Nature biomedical engineering* 2.10 (2018), pp. 741–748.

[105] Yi Wang et al. "Part-based multiderivative edge cross-sectional profiles for polyp detection in colonoscopy". In: *IEEE Journal of Biomedical and Health Informatics* 18.4 (2013), pp. 1379–1389.

[106] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[107] Jun Wei et al. "Shallow attention network for polyp segmentation". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*. Springer. 2021, pp. 699–708.

[108] Bernard Widrow. "Adaline and Madaline - 1963". In: *Proceedings of the IEEE First International Conference on Neural Networks*. San Diego, CA, USA, June 1987, pp. 145–157.

[109] Martin J. Willemink et al. "Preparing Medical Imaging Data for Machine Learning". In: *Radiology* 295.1 (2020), pp. 4–15. DOI: 10.1148/radiol.2020192224.

[110] Ken CL Wong et al. "3D segmentation with exponential logarithmic loss for highly unbalanced object sizes". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 612–619.

[111] Zhe Wu, Li Su, and Qingming Huang. "Cascaded partial decoder for fast and accurate salient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3907–3916.

[112] Niklas Wulms et al. "The Effect of Training Sample Size on the Prediction of White Matter Hyperintensity Volume in a Healthy Population Using BIANCA". In: *Frontiers in Aging Neuroscience* 13 (2022). ISSN: 1663-4365. DOI: 10.3389/fnagi.2021.720636.

[113] Dong Yang et al. "Enhancing Foreground Boundaries for Medical Image Segmentation". In: *arXiv preprint arXiv:2005.14355* (2020).

[114] Wei Yuan and Wenbo Xu. "NeighborLoss: a loss function considering spatial correlation for semantic segmentation of remote sensing image". In: *IEEE Access* 9 (2021), pp. 75641–75649.

[115] Chiyuan Zhang et al. "Understanding Deep Learning (Still) Requires Rethinking Generalization". In: *Communications of the ACM* 64.3 (Feb. 2021), pp. 107–115. ISSN: 0001-0782. DOI: 10.1145/3446776.

[116] Shaoteng Zhang, Jianpeng Zhang, and Yong Xia. "TransWS: Transformer-Based Weakly Supervised Histology Image Segmentation". In: *International Workshop on Machine Learning in Medical Imaging*. Springer. 2022, pp. 367–376.

[117] Yundong Zhang, Huiye Liu, and Qiang Hu. "Transfuse: Fusing transformers and cnns for medical image segmentation". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*. Springer. 2021, pp. 14–24.

[118] Yundong Zhang, Huiye Liu, and Qiang Hu. "Transfuse: Fusing transformers and cnns for medical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2021, pp. 14–24.

[119] Peng Zhao et al. "Emds-6: Environmental microorganism image dataset sixth version for image denoising, segmentation, feature extraction, classification, and detection method evaluation". In: *Frontiers in Microbiology* (2022), p. 1334.

[120] Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. "Automatic polyp segmentation via multi-scale subtraction network". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*. Springer. 2021, pp. 120–130.

[121] Sixiao Zheng et al. "Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 6877–6886. DOI: 10.1109/CVPR46437.2021.00681.

[122] Sixiao Zheng et al. "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.

[123] Zhun Zhong et al. "Random Erasing Data Augmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Apr. 2020, pp. 13001–13008. DOI: 10.1609/aaai.v34i07.7000.

Acknowledgements

I would like to express my gratitude to my supervisor, Loris Nanni, for his mentorship and unwavering support throughout this journey. My heartfelt thanks go out to my fellow classmates who stood by me and assisted in various projects. Lastly, to my family, whose faith in me never wavered and who were always there by my side, thank you.