National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Medical Image Segmentation

A review of modern approaches

Diploma Thesis

of

NATALIA E. SALPEA

Supervisor: Stefanos Kollias
Professor

Athens, July 2022

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Medical Image Segmentation

A review of modern approaches

Diploma Thesis

of

NATALIA E. SALPEA

Supervisor: Stefanos Kollias
Professor

Approved by the examination committee on 13th July 2022.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Stefanos Kollias	Andreas-Georgios Stafylopatis	Georgios Stamou
Professor	Professor	Professor

Athens, July 2022

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....
Natalia E. Salpea

13th July 2022

Abstract

Medical image segmentation involves identifying regions of interest in medical images. In modern times, there is a great need to develop robust computer vision algorithms to perform this task in order to reduce the time and cost of diagnosis and thus to aid quicker prevention and treatment of a variety of diseases. The approaches presented so far, mainly follow the U-type architecture proposed along with the UNet model, implement encoder-decoder type architectures with fully convolutional networks,and also transformer architectures, exploiting both attention mechanisms and residual learning, and emphasizing information gathering at different resolution scales. Many of these architectural variants achieve significant improvements in quantitative and qualitative results in comparison to the pioneer UNet, while some fail to outperform it. In this thesis, 11 models designed for medical image segmentation, and other types of segmentation, are trained and tested, evaluated on specific evaluation metrics, on 4 publicly available datasets related to gastric polyps and cell nuclei, which are first augmented to increase their size in an attempt to address the problem of the lack of a large amount of medical data. In addition, their generalizability and the effect of data augmentation on the scores of the experiments are also examined. Finally, conclusions on the performance of the models are provided and future extensions that can improve their performance in the task of medical image segmentation are discussed.

Keywords

medical image segmentation, polyps, cell nuclei, encoder-decoder, transformers, atrous convolution, convolutional neural networks, squeeze and excitation, residual learning, attention, shape stream, UNet, DeepLabv3+, Vnet,Attention UNet, TransUNet, Swin-UNet,ResUNet, ResUNet++, ResUNet-a,R2U-Net, MSRF-Net, CVC-ClinicDB, Kvasir-SEG, 2018 Data Science Bowl, SegPC

to my parents

Acknowledgements

First of all, I would like to thank Professor Mr. Stefanos Kollias for supervising this thesis and for giving me the opportunity to work on it at the Artificial Intelligence and Learning Systems Laboratory. I would also like to thank Ms. Paraskevi Tzouveli for her guidance and the excellent cooperation we had. Finally, I would like to thank my parents for the guidance and moral support they have given me over the years.

Athens, July 2022

Natalia E. Salpea

Table of Contents

List of Figures

List of Images

List of Tables

Preface

This thesis was carried out in Athens, in the year 2022, at the Artificial Intelligence and Learning Systems Laboratory of the Department of Information and Computer Technology of the School of Electrical and Computer Engineering of the National Technical University of Athens, under the supervision of Professor Mr. Stefanos Kollias and Ms. Paraskevi Tzouveli.

Chapter 1

Introduction

1.1 Subject of the thesis

In this paper we perform a thorough study of the most important architectures presented in the field of medical image segmentation, test and evaluate them on 4 publicly available and widely used datasets and compare the results to draw conclusions on which architectures are most favorable to the task at hand and which are most promising for future developments. The process followed is illustrated in Figure 1.1. Specifically, after the data were collected, split into training, validation and test sets and appropriately transformed to be in the format required by the models, they were subjected to a data augmentation process in order to increase their number and introduce an additional diversity factor. The 11 models, discussed in following sections, are then trained on the augmented data and on the un-augmented data, and on supersets resulting from combining sets, with appropriately selected hyperparameters, loss functions and evaluation metrics, and their snapshots that achieve the maximum scores on the evaluation metrics are stored. The models are then tested over the test subsets and the segmentation masks are generated which are qualitatively compared with the actual masks, and also the selected evaluation metrics are calculated. Once all these data are collected, a qualitative and quantitative comparison of the architectures and analysis of the datasets follows. In addition, experiments and comparisons are performed to test the effect of data augmentation on the performance of the models, as well as testing the generalizability of the models through training and evaluation on different datasets. Finally, some conclusions and some ideas for future expansions of this work are provided, which could lead to an improvement of the state-of-the-art scores.

1.2 Organisation of the volume

This thesis is organized in 7 chapters:

1. Chapter 1 provides an introduction to this thesis.

2. Chapter 2 details the topic of the thesis and provides a historical review of previous related work in the field.

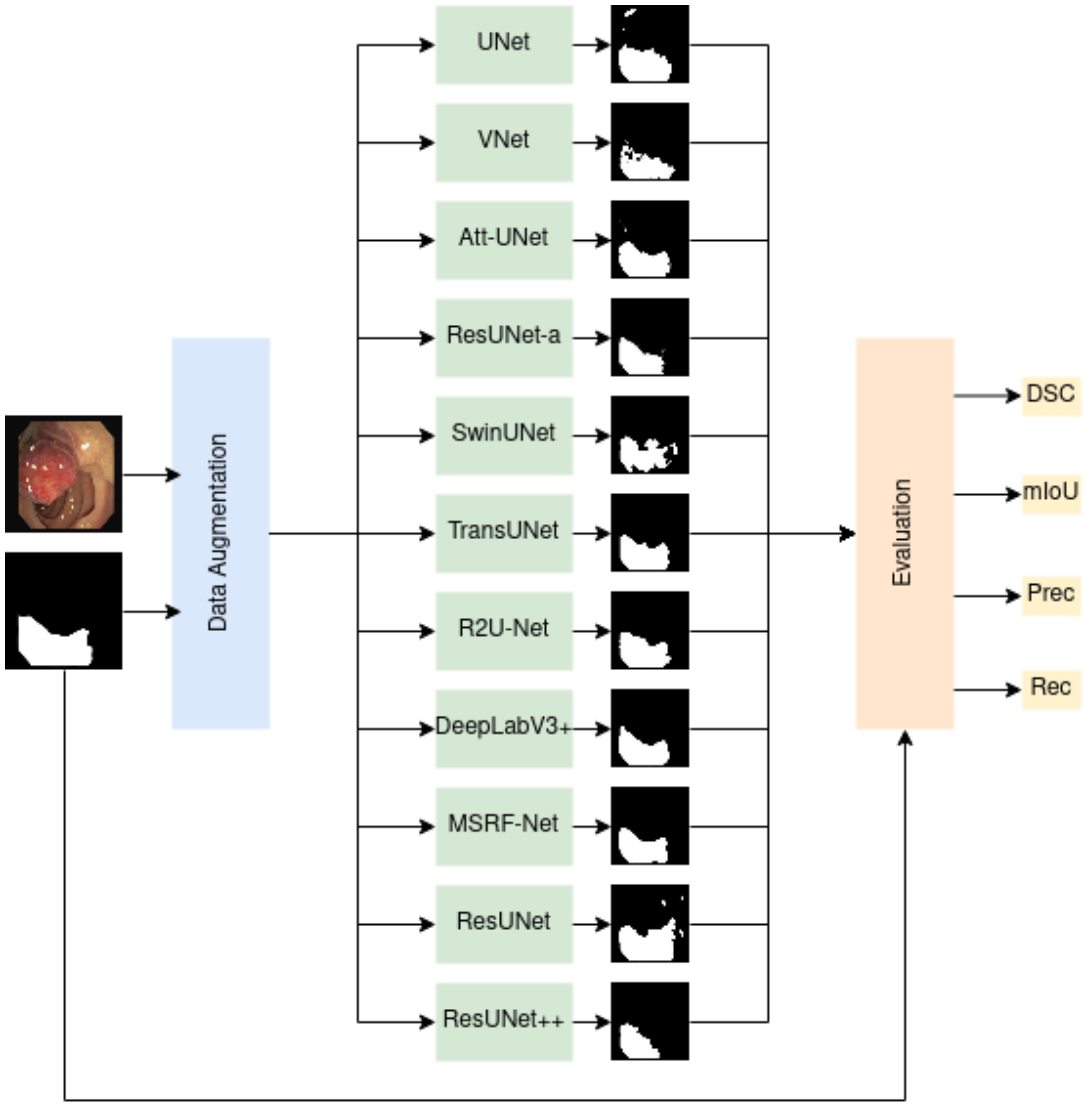

Figure 1.1. *Complete pipeline of the thesis*

3. Chapter 3 gives the theoretical background of the main architectures and the basic mechanisms that have been used in the thesis.

4. Chapter 4 presents the datasets and the preprocessing and augmentation procedures they have undergone.

5. Chapter 5 provides details of the models and architectures used in the experiments of the thesis, the evaluation metrics as well as the parameters and the computational system.

6. Chapter 6 describes the experiments performed and presents their results, but also studies the generalizability of the models and the effect of data augmentation on their performance.

7. Chapter 7 is the epilogue of the thesis, where conclusions are drawn about the architectures presented in the above chapters and future possible extensions are discussed.

8. In Appendix A some more specific concepts are discussed, for completeness of the theoretical background.

Chapter 2

Subject Description

In this chapter the topic of this thesis is described, which is medical image segmentation, and a historical review of the architectures and algorithms that have been used in this area is provided.

2.1 Medical Image Segmentation

Medical image segmentation is an essential task for faster and better diagnosis of diseases and for ensuring faster and better treatment of large numbers of patients simultaneously, reducing the need for human intervention as well as human errors, time and cost. It belongs to the field of semantic segmentation and involves the identification of regions of interest in medical data such as 2D images, 3D brain MRIs (MRI [32, 33, 34]), CT scans [35, 36], ultrasounds and [37] X-rays, among others. The results of segmentation can help identify areas associated with disease, such as polyps in the intestine, in order to analyze them and decide if they are cancerous and need to be removed. This diagnosis can be life-critical for patients as it can help prevent serious forms of cancer. Due to the slow and tedious process of manual segmentation [38], there is a great need for computer vision algorithms that can perform segmentation quickly and accurately without the requirement for human involvement. Other challenges in the field of medical imaging are the small amount of data available, and the need for professionals specialized in this field when generating the data. When annotating polyps in images for example, protocols and guidelines are defined by the expert performing the annotation at that time. However, there are discrepancies between experts, for example in deciding whether a region is cancerous or not. Furthermore, due to the quite significant variation between data due to different imaging quality, the software used to perform the annotation, the expert performing the annotation, the brightness of the images and general biological variations, segmentation algorithms need to be robust and have a high generalization ability.

2.2 Related works

In recent years, Convolutional Neural Networks (CNNs) have dominated the field of segmentation in various types of medical data, and more generally [39, 40, 41]. The most modern architectures for semantic and instance segmentation are usually encoder-

decoder networks, whose success is due to skip-connections, which allow semantically important and dense feature maps to be forwarded from the encoder to the decoder and reduce the semantic gap between the 2 subnetworks. The implementation parameters of this type of models differ depending on the biomedical application. Moreover, the small amount of data makes the models' task more difficult and requires specialized design in order to be addressed as a problem. One of the first and most popular approaches for the task of semantic medical image segmentation that addresses the above problem is UNet [7], which converted the Fully Convolutional Network (FCN) of [2] that has only convolutional layers into a U-type encoder-decoder architecture in which low- and high-level features are combined via skip-connections. However, a semantic gap may arise between these features. In order to bridge the gap, it is proposed in [42] to add convolutional units to the skip-connections. In [10], a version of UNet with an attention mechanism is proposed which helps to implement a pruning mechanism to eliminate unnecessary spatial features passing through the skip-connections. UNet was not only restricted to the medical imaging domain but versions of it for different types of segmentation, such as [13], appeared. The 3D version of UNet, VNet [8] performs segmentation on sparsely annotated volumetric images, and consists of an FCN with skip-connections. Residual links improve the information flow while reducing the convergence time, which is why they are preferred in several approaches, such as [11]. In [12], Squeeze and Excitation blocks, attention mechanisms as well as Atrous Spatial Pyramid Pooling (ASPP) are added to the ResUNet model. In [9], residual connections are combined into an iterative network which leads to better feature representations. The ASPP mechanism appeared in [43] to pool global features, and evolved in [16] into an architecture that uses skip-connections between the encoder and decoder. Squeeze and Excitation blocks were proposed in [44] and model interdependencies between channels and extract global maps that emphasize important features and ignore less relevant features. The idea of a gated shape stream appeared in [45] and it serves to produce more detailed segmentation maps by taking into account the shape of the region of interest. To maintain high resolution representations, it was proposed to exchange low and high level features between different resolution scales, which is shown in [46] to lead to more spatially accurate final segmentation maps. Finally, [17] combines many of the above techniques such as the Squeeze and Excitation block, the gated shape stream, the attention mechanism, with an emphasis on feature exchange between analysis scales (multi-scale fusion), into a robust model that consistently performs well on the medical data types contained in this work.

Chapter 3

Theoretical background

This chapter provides the theoretical background for the basic architectures and mechanisms used in this thesis.

3.1 Machine Learning and Neural Networks

3.1.1 Machine Learning

Machine Learning falls under the field of Artificial Intelligence and, according to Arthur Samuel, allows computers to learn from data, make accurate predictions, and even improve them automatically, without being explicitly programmed to do so. Machine learning algorithms are fed input data, perform statistical analysis to predict an output, and update themselves appropriately as new data appears. While machine learning belongs to the field of computer science, it is quite different from traditional computational approaches, in which algorithms are explicitly programmed with specific instructions to solve a particular problem.

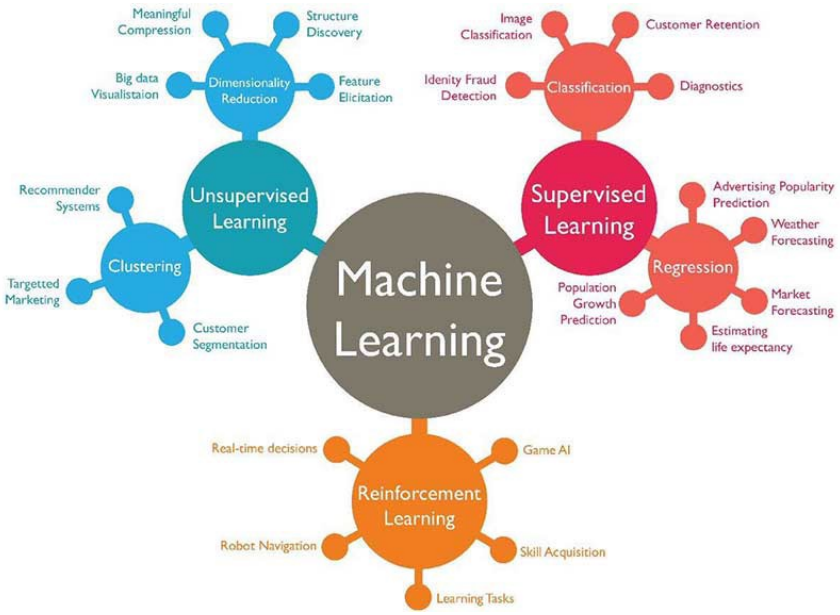

Image 3.1: *Domains of machine learning [22]*

Modern technology has benefited a lot from machine learning. For example, facial recognition technologies, digit and letter recognition, object tracking, recommendation systems, and even self-driving vehicles and disease diagnosis are just some of the areas where machine learning has made significant contributions, and it is an ever-evolving field that has much to offer in the future.

Machine learning algorithms are divided into 3 categories:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

In supervised learning, systems are fed with data, each of which is assigned a label [47]. The goal is to approximate the mapping function with enough precision that when a new input is presented to the model, the output can be correctly predicted. Examples of tasks that fall under supervised learning are classification, in which categories are assigned to the input data, and regression, in which a continuous value is approximated, for example the price of a stock at some time in the future.

In unsupervised learning, the model is fed with unlabeled or unclassified data and acts on it without prior training. Examples of tasks in this area are clustering, in which internal similarities between data are identified on the basis of which clusters are created, and association, in which rules are sought that dictate a particular association between data, for example whether people who buy product X will also buy product Y.

In reinforcement learning, an algorithm, or agent, learns through interaction with its environment. It receives reward when it performs a correct step and punishment when it performs an incorrect step. It attempts to maximize its reward and minimize its punishment without human intervention.

3.1.2 Neural Networks

Neural networks are the building blocks of machine learning models. Many tests involving intelligence, pattern recognition and object detection are difficult to automate yet are easily performed by humans, even young children and animals. The biological neural networks present in the brains of living organisms with intelligence can perform these complex tasks, and machine learning artificial neural networks attempt to model them in order to mimic them, hence their name. Artificial neural networks involve directed graph structures where each node (neuron) performs a computation. Each connection carries a signal, along with a weight that indicates whether the signal is amplified or diminished and determines its significance for the output.

The building blocks of artificial neural networks are neurons, just like in the brain. In the brain, each neuron is connected to about 10,000 other neurons, from which they receive and to which they send electrochemical signals that lead to activation of the neuron if they are strong enough. This activation is a binary state that can be modelled by a computer. Thus the idea of artificial neural networks was born [48].

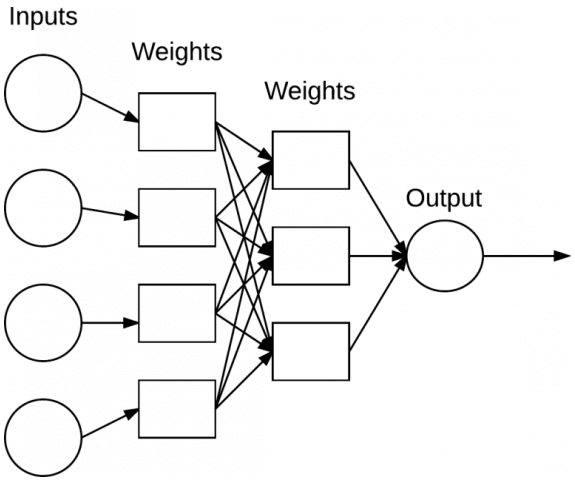

Figure 3.1. *Example of an Artificial Neural Network*

3.2 Convolutional Neural Networks

Convolutional Neural Networks are a deep learning algorithm that takes as input images and locates objects in them by assigning meaning to them through weights [1]. They are primarily intended for the task of classification but are very powerful in feature extraction. They have the ability to detect spatial and temporal dependencies in images through the application of appropriate filters, reducing the number of parameters compared to previous approaches. Their role is to bring images into a simpler form that is easier to process, without losing important information. They consist of 3 main neural layers, the convolutional layer, the pooling layer and the dense fully connected layer, as shown in figure 3.2.

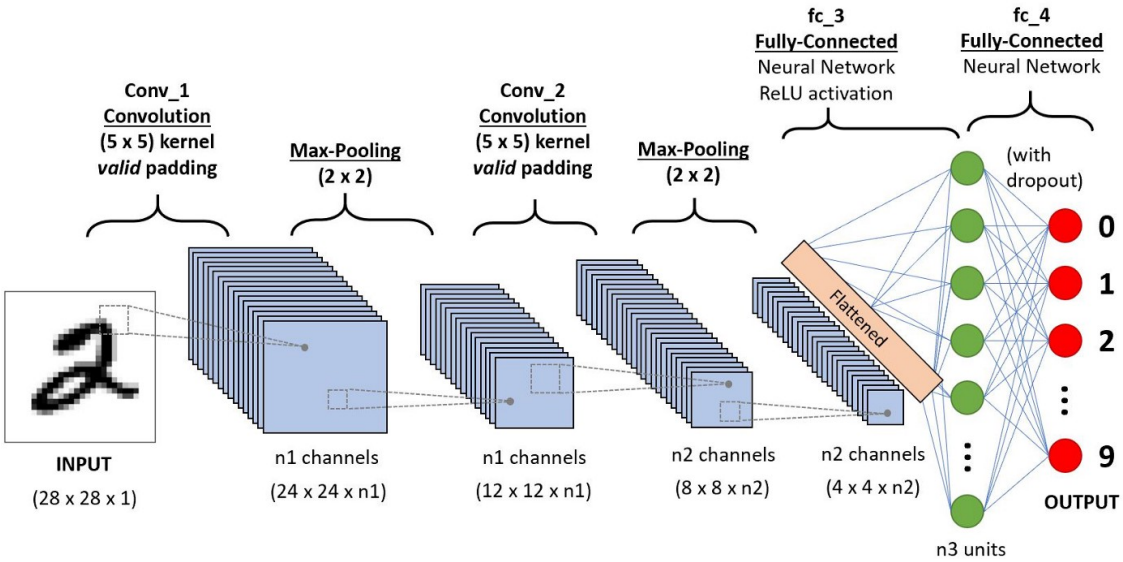

Figure 3.2. *Example of a convolutional neural network (CNN) for classifying digit images [1]*

The purpose of the convolutional layer is to extract high-level features [49] such as

edges or other visual elements, by performing the operation of convolution with several kernels in the image-entry. There is not only one such layer in convolutional networks, and the combination of many such layers leads to a total understanding of the image and all its high- and low-level features, for example edges, corners and color.

The pooling layer reduces the spatial size of the features resulting from the convolutional layers by performing downsampling, thus also reducing the computational requirements of the network. Moreover, it is useful in extracting dominant features that are rotation and position invariant, preserving the quality of model training. There are two types of pooling, Max Pooling which returns the maximum value of the image slice covered by the kernel achieving dimensionality reduction as well as de-noising, and Average Pooling which returns the average value of the image slice covered by the kernel achieving dimensionality reduction.

The fully connected layer learns nonlinear combinations of high-level features derived from the final convolutional layer. The disadvantages of convolutional networks are that they cannot work with data of different sizes, and also they cannot be used in the segmentation task since the number of objects/areas of interest is not specific and therefore the size of the output layer cannot be fixed.

3.3 Fully Convolutional Networks (FCN)

In fully convolutional networks there are only convolutional layers. Their difference from simple convolutional networks is that the last dense fully connected layer is replaced by a fully convolutional layer, as shown in Figure 3.3. They were proposed in [2] and can produce spatial segmentation maps and dense pixel-level predictions at the output for the whole image rather than for patches.

They use skip-connections that upsample the feature tables from the last layer and combine them with the feature tables of previous layers. This produces detailed segmentation maps. However, they have some limitations such as being quite slow models for real-time inference and that they greatly reduce the resolution of features passing through multiple layers of convolution and pooling, resulting in low-resolution predictions with inaccurate object contours.

3.4 Transformers

Transformers were originally designed for the task of language translation. They allow modeling of long dependencies between input sequences and support parallel sequence processing as opposed to iterative networks. In addition, their design allows the processing of different types of data, such as images, video, text and voice. For this reason, they have recently become prevalent as architectures and are spreading to more and more domains beyond Natural Language Processing, with the prospect of completely replacing Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) [50].

At their core they consist of a stack of encoder and decoder layers. The encoder contains the very important Self-Attention layer that computes the relationship between

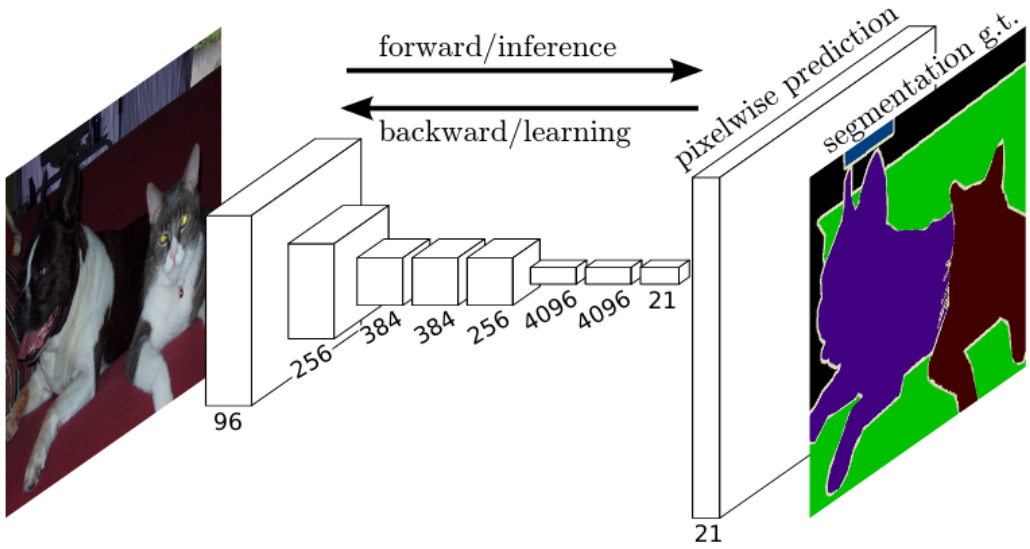

Figure 3.3. *Example of a Fully Convolutional Network (FCN) for semantic segmentation [2]*

different parts of the input sequence as well as a Feed-forward layer. It also includes residual connections as well as normalization layers. The Self-Attention mechanism of the encoder generates a representation for each token of the input sequence that describes its semantic similarity to the other tokens [51]. Its architecture is shown in Figure 3.4 .

The decoder generates one token at a time, taking into account both the embeddings and the output so far. The Self-Attention mechanism does not care about the location of the tokens and its results are invariant to shuffling operations. However, in processes such as image segmentation, spatial information and the relative position of objects/areas of interest are important. The way to introduce this information into the training process is to append it to each token before it is passed to the attention mechanism, so that it participates in the learning process [52]. This makes transformers more demanding in terms of the data needed for their training than CNNs.

3.5 Encoder-Decoder Networks

Encoder-Decoder networks are closely related to the Transformers. They are used in tasks such as image captioning, sentiment analysis and translation. They were designed to solve sequence problems, i.e. problems in which the input and output are sequences. At a high level, the encoder and decoder are two blocks connected by a context vector [4], as shown in Figure 3.5 .

The encoder processes each token of the input sequence, accumulating all the information about the input into a fixed-length vector. The context vector contains all the semantic information about the input sequence, which is necessary for the decoder to produce its predictions in turn, i.e. the output sequence, token by token. The internal ar-

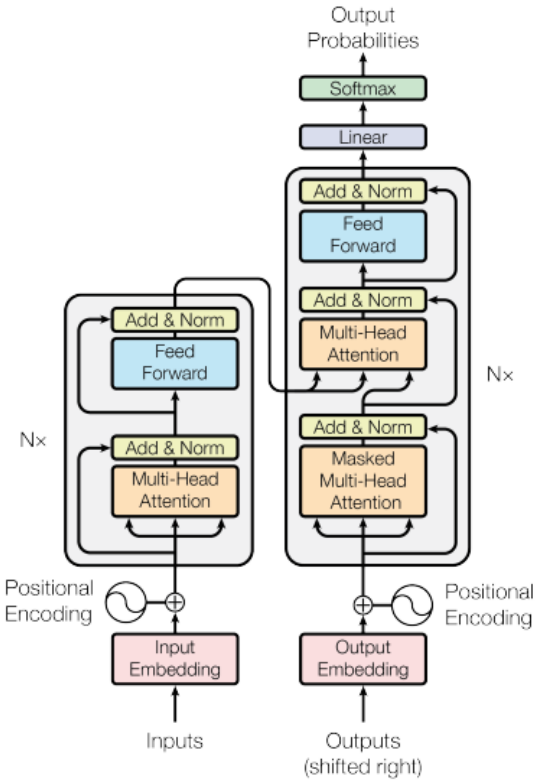

Figure 3.4. *Example transformer architecture [3]*

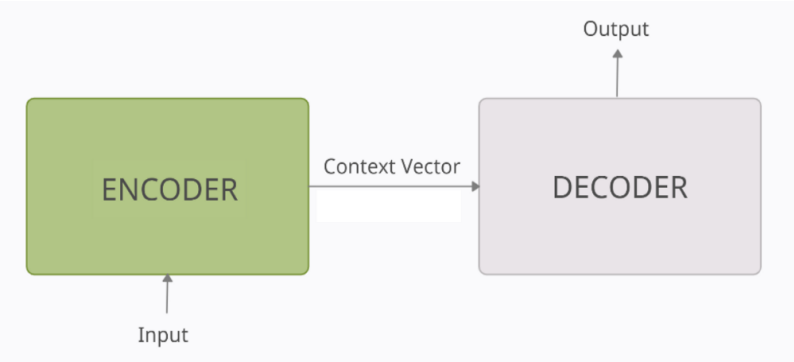

Figure 3.5. *Example of encoder-decoder architecture[4]*

chitecture of the two blocks may differ depending on the task for which they are intended, for example for the translation task, the two blocks consist of LSTM modules. Another example of a different architecture, for the task of generating caption for image is shown in Figure 3.6 , in which the image is first passed through a Convolutional Network (CNN) and the features from the last dense network are fed into the LSTM network. This layer acts as the context vector since it contains all the essence of the image [5].

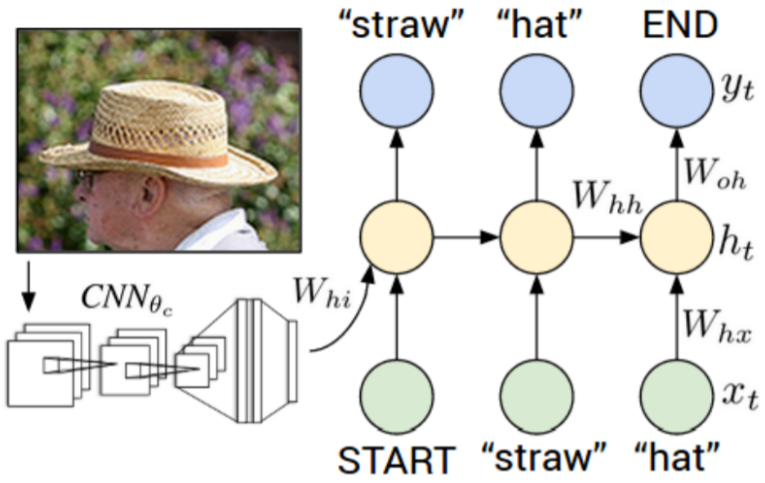

Figure 3.6. *Example encoder-decoder architecture for the image captioning task [5]*

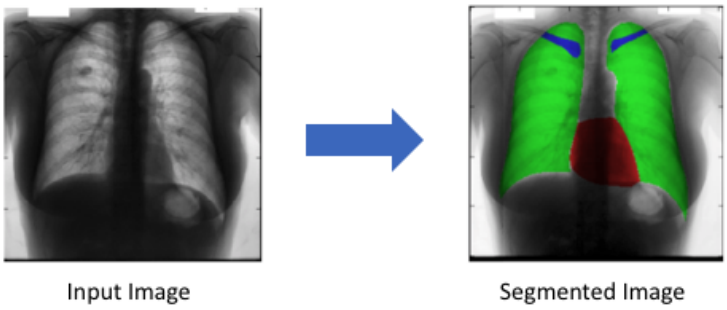

Image 3.2: *Example of semantic segmentation in a chest X-ray image in which the heart (red), the sternum (green) and the clavicles (blue) have been segmented [23]*

3.6 Semantic Segmentation

Semantic image segmentation is the task of categorizing each pixel into one of the classes of a predefined set [53]. For example, in Figure 3.2 the pixels belonging to the heart have been categorized into the class "heart" and similarly for the lungs and collarbones and for the background. Semantic segmentation is different from object detection because it does not produce bounding boxes for localized objects, it is also different from instance segmentation which would give a different label to each instance of the same type of object. Semantic segmentation places all objects of the same type in the same category no matter how many times they appear in the image. Consequently, this segmentation produces an output image of the same size as the original (single-channel) image that contains labels for all pixels according to the class to which they were calculated to belong.

3.7 Attention mechanism

In psychology, attention is the cognitive process of selectively focusing on one or more things while ignoring the rest. Just as neural networks are an attempt to model human brain processes, the attention mechanism mimics human attention by focusing on a few relevant things while ignoring the rest in deep neural networks. The basic idea is that whenever the model predicts an output token, it only utilizes the parts of the input where the most relevant information is clustered instead of the entire sequence. The attention mechanism connects the encoder to the decoder and provides the decoder with information about the hidden states of the encoder so that the model can pay more attention to the most relevant parts of the input sequence by assigning scores. The attention mechanism serves to model longer distance dependencies within the model, which is an important problem in recurrent neural networks. While it originally appeared for the task of language translation it has spread to areas such as computer vision [54]. Before it was first proposed in [55], language translation relied on the use of iterative encoder-decoder models (RNNs/LSTMs), which exhibit difficulty in understanding longer input sequences and therefore produce unsatisfactory results. Even LSTMs (Long Short Term Memory) designed to model longer dependencies tend to 'forget' important information in many cases. There are different popular attention mechanisms such as Content-base attention, Additive, Location-Base, General, Dot-Product, Scaled Dot-Product as well as some more general categories such as Self-Attention, which assigns weights to the tokens of a single sequence and produces its representation, Multi-Head Self-Attention which combines information from different subspaces of representations, Global/Soft Attention and Soft/Hard Attention [56].

3.8 Residual Connections

Residual connections appeared in order to help faster convergence when training deeper neural networks. In classical feed-forward networks information proceeds through the layers serially, the output of one layer being the input of the next layer. Residual connections provide an alternative path for data to reach subsequent layers of the network. If F is the function describing the layers of the network, as illustrated in Figure 3.7, then in a classical feed-forward network the output of the network would be F(x), but in a network with residual links the output is F(x) + x since the input x passes directly to the output. Along the residual connections there may be functions instead of just passing the input straight to the output [57].

In deep neural networks, the problems of exploding and vanishing derivatives [58] usually occur. Residual connections help combat these problems and promote faster convergence during training.

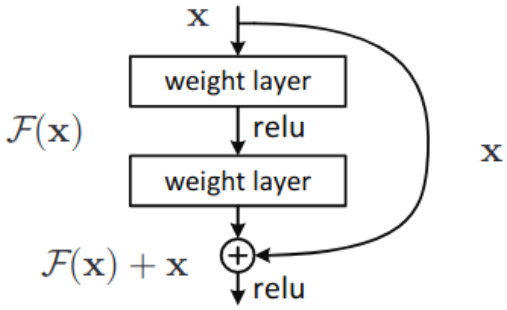

Figure 3.7. *The architecture of a residual block [6]*

Chapter 4

Data

This chapter presents the datasets used in this work, for training the models as well as for testing them. For testing medical image segmentation, unfortunately there are not enough public datasets of sufficient size and quality. To address the first problem, that of insufficient sample number, data augmentation works catalytically [59]. However, regarding the second problem, that of quality, it is not considered an easy challenge to address because quality is determined by the machines used by each laboratory, therefore the quality of images can only be improved when there is a significant development in the technology of the sample acquisition machines. The sets used belong to two broad categories involving polyp segmentation and cell nucleus segmentation respectively. Different types of sets were chosen in order to test the robustness, i.e. the ability of the model to perform similarly on both easy and difficult images, and the generalizability of the models, i.e. their ability to perform correctly on different datasets, such as being trained on data from one laboratory and tested on data from another laboratory acquired with different technology [60].

4.1 Datasets

4.1.1 CVC-ClinicDB

One of the datasets used in this paper is CVC-ClinicDB [24], which is a database of frames from colonoscopy videos. These frames contain various examples of polyps. The set contains 612 images of 384x288 pixels and the segmentation masks corresponding to the areas covered by polyps in the images, as shown in Figure 4.1. The masks are binary and pixels depicting the polyp tissue are positive (white mask/foreground) while the rest of the image is black.

The dataset consists of two folders, 'Original' containing the frame images in .tiff format, which is the property of Hospital Clinic, Barcelona, Spain, and 'Ground Truth' containing the polyp masks in the images, also in .tiff format, which is the property of Computer Vision Center, Barcelona, Spain. Figure 4.2 shows the correspondence between the images and the videos. CVC-ClinicDB is the official dataset used in the training phase of the MICCAI 2015 Sub-Challenge for Automatic Polyp Detection in colonoscopy videos. Its use is free for research and educational purposes.

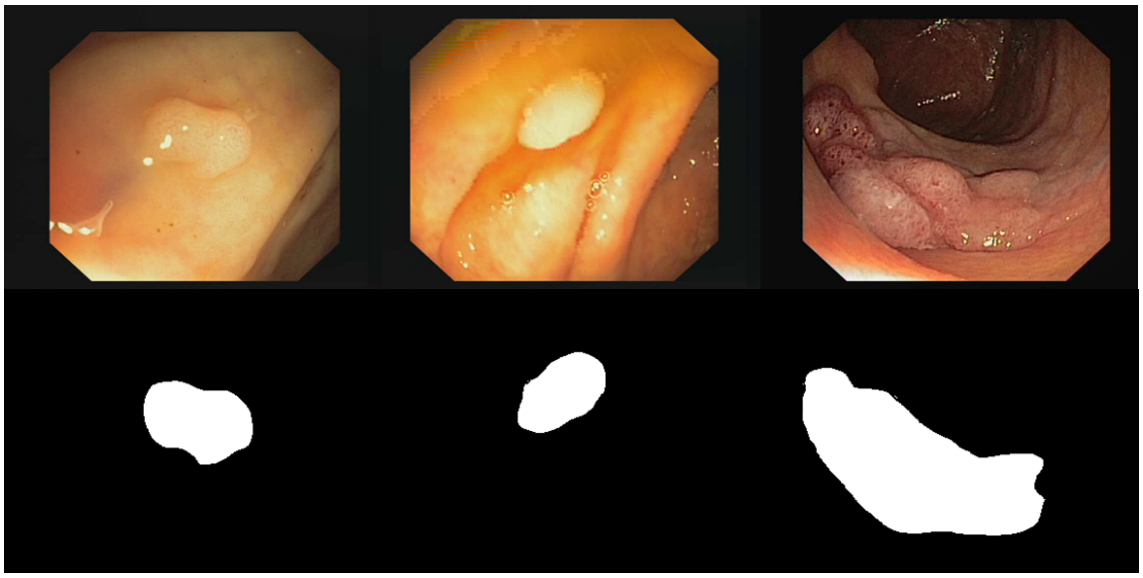

Image 4.1: *Examples of image-mask pairs from the CVC-ClinicDB set [24]*

Sequence	1	2	3	4	5	6	7	8
Frames	1-25	26-50	51-67	68-78	79-103	104-126	127-151	152-177
Sequence	9	10	11	12	13	14	15	16
Frames	178-199	200-205	206-227	228-252	253-277	278-297	298-317	318-342
Sequence	17	18	19	20	21	22	23	24
Frames	343-363	364 -383	384 -408	409 -428	429 -447	448 -466	467 -478	479 -503
Sequence	25	26	27	28	29			
Frames	504-528	529 -546	547 -571	572-591	592-612			

Image 4.2: *Video-frames mapping from the CVC-ClinicDB set*

4.1.2 Kvasir-SEG

Another dataset used in this work is the Kvasir-SEG [25]. Kvasir-SEG is an open access dataset with images of gastrointestinal polyps and corresponding segmentation masks, with manual annotation and verification by an experienced gastroenterologist. Adding segmentation masks to the Kvasir dataset [61], which until then consisted only of contextual annotations, enables computer vision researchers to contribute to the field of polyp segmentation and automatic analysis of colonoscopy videos. Early detection and evaluation of these polyps with subsequent biopsy and removal of polyps has a huge impact on colorectal cancer survival. Several studies have shown that polyps are often overlooked during colonoscopies, with polyp loss rates of 14%-30% depending on the type and size of polyps. Increasing the detection of polyps has been shown to reduce the risk of colorectal cancer. Thus, automatic detection of more polyps at an early stage may play a critical role in improving both prevention and survival from colorectal cancer. This is the main motivation behind the development of the Kvasir-SEG dataset.

This dataset consists of 1000 images of polyps and the corresponding segmentation

masks of the detected polyps. The resolution of the images varies from 332x487 to 1920x1072 pixels. The images and the corresponding masks shall be stored in two separate folders with the same file name. The image files are encoded with JPEG compression. In addition, the coordinates of the polyp bounding boxes in the images are stored in a JSON file, which however was not utilized in this work.

The segmentation masks were created using the Labelbox software, which is a tool used to label the region of interest (ROI) in image frames, i.e. the polyp regions in our case. The creators manually marked polyps in all 1000 images with the help of expert physicians. The pixels depicting the polyp tissue, the region of interest, are represented by the foreground (white mask), while the background (in black) contains no positive pixels. Some of the original images contain the image of the endoscope position marking probe, ScopeGuide TM, Olympus Tokyo Japan, located in one of the lower corners and shown as a small green box. As this information is unnecessary for the segmentation task, it has been replaced with black boxes.

Data were collected using endoscopic equipment at Vestre Viken Health Trust (VV) in Norway. The VV consists of 4 hospitals. One of these hospitals (Bærum Hospital) has a large gastroenterology department from which training data have been collected. In addition, the images are carefully annotated by one or more expert physicians from VV and the Cancer Registry of Norway (CRN). CRN provides new insights into cancer through cancer research. It is part of the Regional Health Authority of South-East Norway and is organised as an independent institution within Oslo University Hospital. CRN is responsible for national cancer screening programmes with the aim of preventing death from cancer by detecting cancers or precancerous lesions as early as possible. The use of this dataset is free for educational and research purposes.

4.1.3 2018 Data Science Bowl

The 2018 Data Science Bowl dataset [26] contains a large number of segmented cell nucleus images, and was created as part of a competition on the Kaggle platform, having the same title. The images were acquired under a variety of conditions and vary in terms of cell type, magnification and imaging mode (brightfield vs. fluorescence). The purpose of the dataset is to test and evaluate the ability of an algorithm to perform under these variations.

Each image is represented by an ImageId. The files corresponding to an image are contained in a folder with this ImageId. Within this folder there are two subfolders,one containing the image file and the other containing the segmented masks of each kernel. Each mask contains one nucleus, so each image can have more than one mask assigned to it. Masks are not allowed to overlap, so no pixel belongs to two masks.

For the competition, a dataset of 37,333 nuclei was generated with manual annotation on 841 2D images from more than 30 experiments on different samples, cell strains, microscopy instruments, imaging conditions, operators, research facilities and staining protocols. The annotations were manually performed by a team of expert biologists at the Broad Institute. These biologists manually delineated each object in the images using one

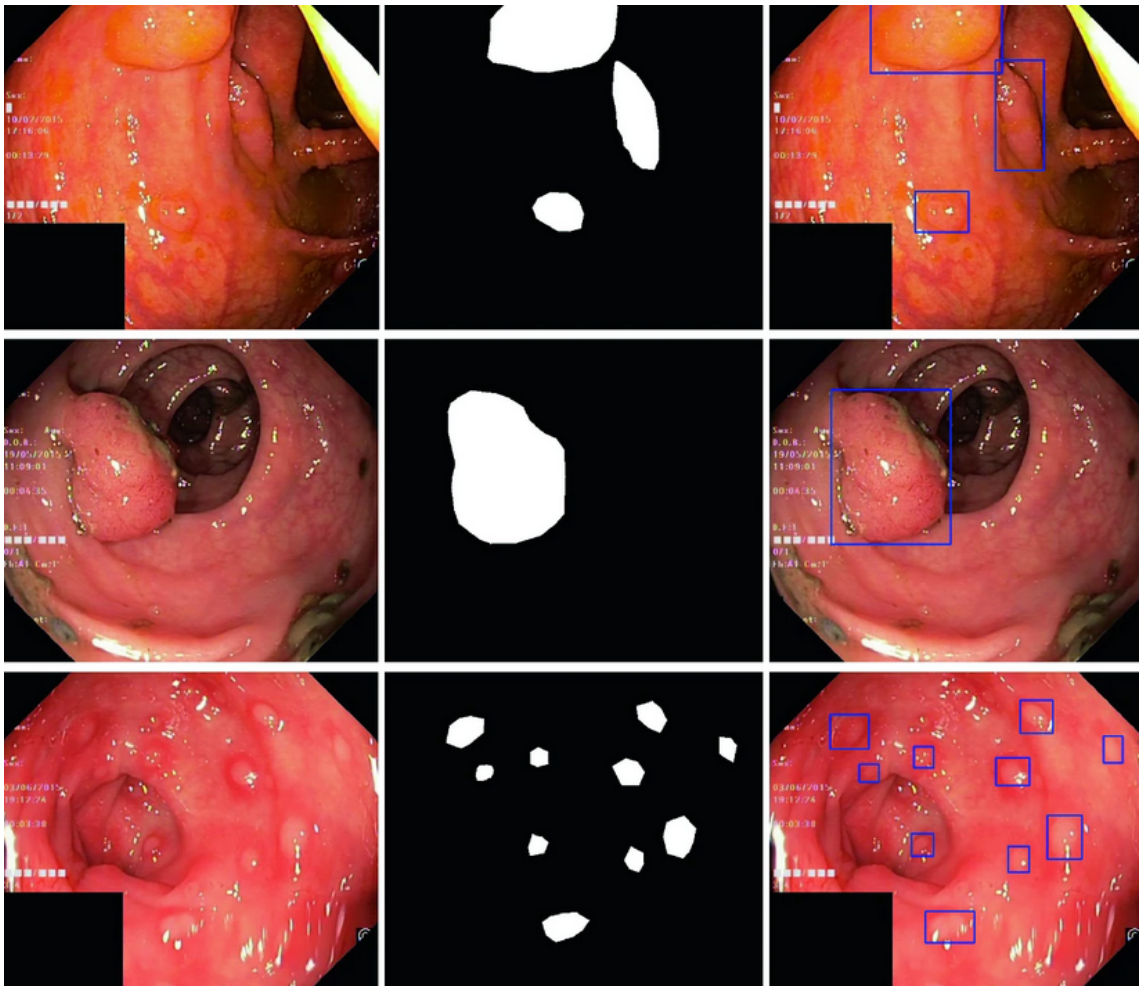

Image 4.3: *Examples of image-mask-bounding boxes from the Kvasir-SEG set [25]*

of two tools: (1) an annotation utility that presets superpixel segmentations to facilitate selection of foreground or background regions and (2) the image editing software GIMP to create annotation masks by coloring individual pixels describing each nucleus. In total, the dataset contains images from more than 30 different biological experiments, which were divided into 16 experiments for training (670 images, 29,464 nuclei) and first-stage evaluation (65 images, 4,152 nuclei) and exactly 15 experiments for second-stage evaluation (106 images, 3,717 nuclei). In this paper, only data from the first stage of the competition were used, and only the training set as the segmentation masks are not provided in the available test set.

The competition was run for a total of 3 months during which participants had access to the training set (with target masks) and the first-stage test sets (in the absence of a target mask). The main goal of the competition and dataset was to investigate general segmentation strategies that could be automatically applied to many imaging experiments without further user intervention. This approach can reduce the time to quantify images, enabling future generations of biologists to adopt and perform more quantitative imaging experiments for research and clinical practice.

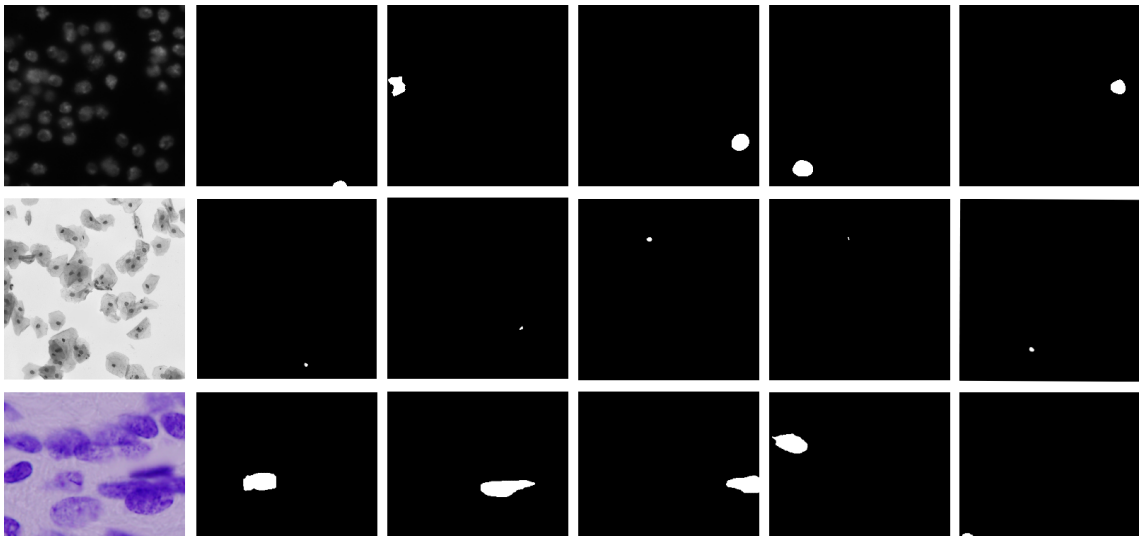

Image 4.4: *Examples of image-mask pairs from the 2018 Data Science Bowl set [26]*

4.1.4 SegPC

For this dataset, called SegPC [27, 28, 29, 30] and accompanying a competition on the Kaggle platform under the same title, microscopic images were recorded from bone marrow aspiration slides of patients diagnosed with Multiple Myeloma (MM), a type of white blood cancer. Slides were stained using Jenner-Giemsa staining and plasma cells, which are cells of interest, were segmented. Images were recorded in raw BMP format using two cameras, 1) at 2040x1536 pixel size using cellSens Version 2.1 software (Olympus) connected to the microscope and 2) at 1920x2560 pixel size from a Nikon camera connected to the microscope.

Recently, efforts have been made to build computer-assisted diagnostic tools for cancer diagnosis through image processing. Such computer-assisted tools require taking images, normalizing the color of the images, segmenting the cells of interest, and classifying them to count malignant versus healthy cells. This dataset aims at a robust cell segmentation which is the first step in creating such a tool for plasma cell cancer, namely multiple myeloma (MM), which is a type of blood cancer. The images are provided after colour normalisation.

The problem of plasma cell segmentation in MM is difficult for several reasons; 1) There is a varying amount of nucleus and cytoplasm from one cell to another. 2) Cells may appear in clusters or as isolated single cells. 3) Cells appearing in clusters may have three situations- (a) the cytoplasm of two cells touch each other (b) the cytoplasm of one cell and the nucleus of another cell touch each other, (c) the nucleus of the cells touch each other. Since the cytoplasm and nucleus have different colours, cell segmentation can pose challenges. 4) There may be many cells touching each other in the cluster. 5) There may be non-stained cells, say a red blood cell under the cell of interest, changing its color and shade. 6) The cytoplasm of a cell may be close to the background of the entire image, making it difficult to identify the cell's boundary and segment it. Therefore, the problem is very challenging and interesting. This is an attempt to build an automated

procedure for cancer detection in multiple myeloma.

The SegPC dataset was used in the IEEE ISBI 2021 medical image competition. The training dataset consists of 298 images and their corresponding masks. For each image in subfolder x, there are the corresponding segmentation masks in subfolder y, only of the cells of interest. The validation dataset consists of 200 images and their corresponding masks, while the test dataset consists of 277 images without the segmentation masks. In the masks, both the nuclei and the cytoplasm of the cells have been marked as different pixel classes. For the same reason as above, only the training and validation sets have been used in this paper.

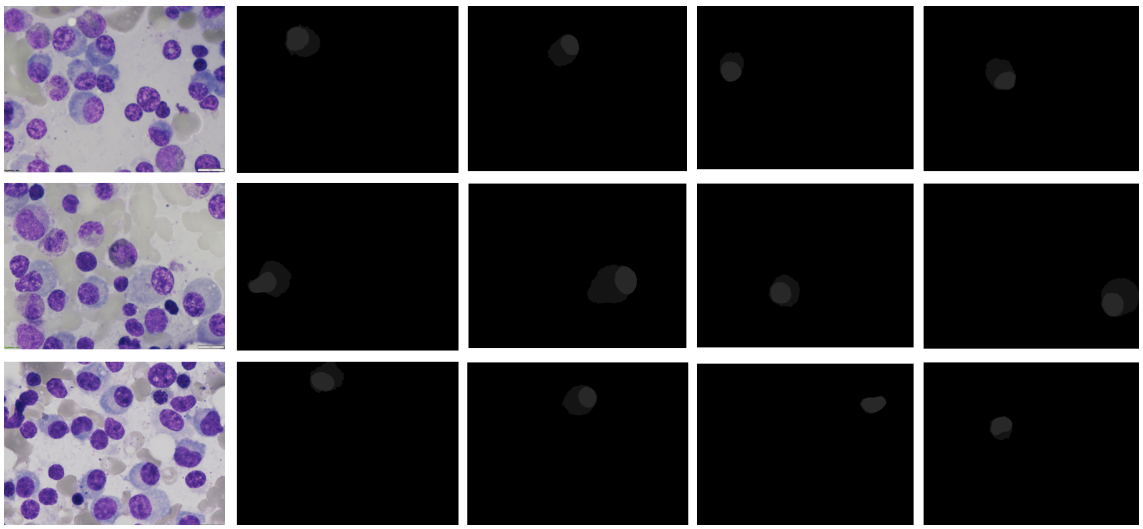

Image 4.5: *Examples of mask images for the set SegPC [27, 28, 29, 30]*

4.2 Data Preprocessing

Some of the datasets were not in the format dictated by the experimental procedure from the start, so a pre-processing procedure was needed. Specifically, all the datasets were divided into training, validation and test folders in the ratio of 80%, 10% and 10% respectively. Within each of these folders there are two folders containing the images and masks respectively. For those sets where the test set (2018 Data Science Bowl, SegPC) was not present, it was re-created from the remaining data, training and validation, in the ratio mentioned above. The number of images in each folder is shown in Table 4.1 .

For the 2018 Data Science Bowl and SegPC datasets there was additional preprocessing because the accompanying competition tasks are somewhat different from the purpose of this work. Specifically, in the 2018 Data Science Bowl a separate mask is given for each kernel identified in the image. These masks were merged so that for each image there is a unique binary mask that includes all cores. Similarly for the SegPC set in which for each image a mask of three values is given (one value for the foreground, one value for the nucleus, one value for the cytoplasm), the masks are merged into a final binary mask containing all the nuclei in each image (the cytoplasm is ignored for consistency with the 2018 Data Science Bowl set). The final forms of the two sets are shown in Figures 4.7

and 4.8.

Finally, joining the Kvasir-SEG and CVC-ClinicDB datasets created a new larger dataset that for convenience was named "Polyps dataset", since both datasets are intended for polyp segmentation. The creation of this set was deemed necessary for some experiments to be reported in later chapters concerning the generalizability of the models. Similarly, the "Cells Dataset" was created by merging the 2018 Data Science Bowl and SegPC. The sample numbers of these new sets are shown in Table 4.2 .

Dataset	**Images**	**Train**	**Valid**	**Test**
CVC-ClinicDB	612	489	61	62
Kvasir-SEG	1000	800	100	100
2018 Data Science Bowl	670	536	67	67
SegPC	497	397	50	50

Table 4.1. *Number of samples in each dataset*

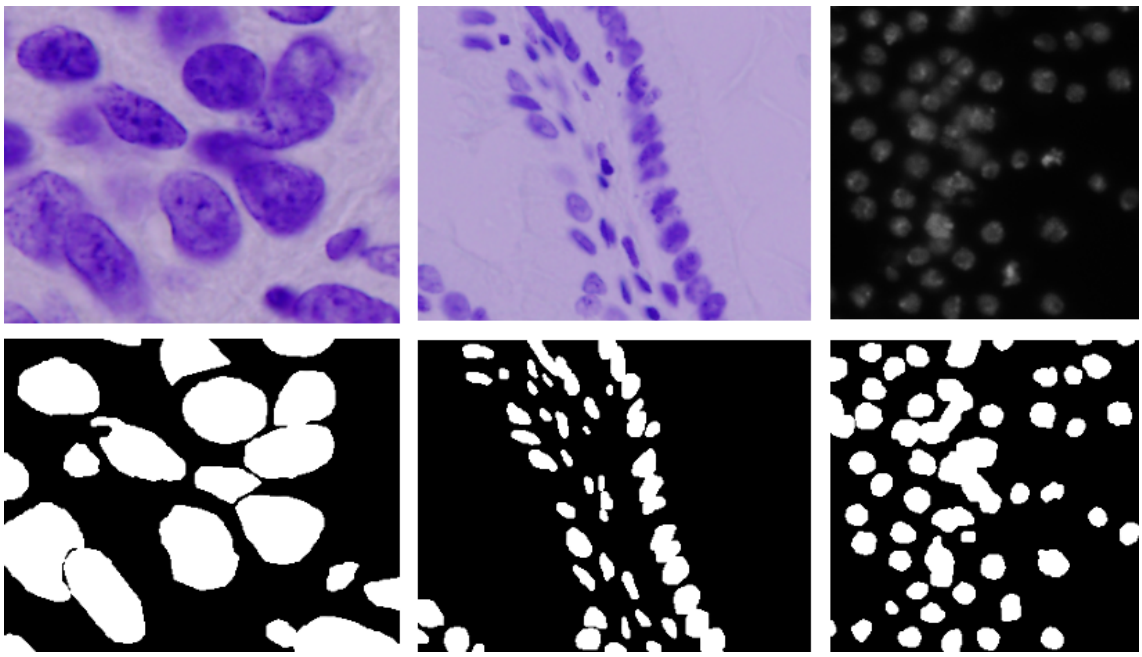

Image 4.6: *Examples of the new format of the samples in the 2018 Data Science Bowl set [26]*

Dataset	**Images**	**Train**	**Valid**	**Test**
Polyps Dataset	1612	1289	161	162
Cells Dataset	1167	933	117	117

Table 4.2. *Number of samples in new datasets*

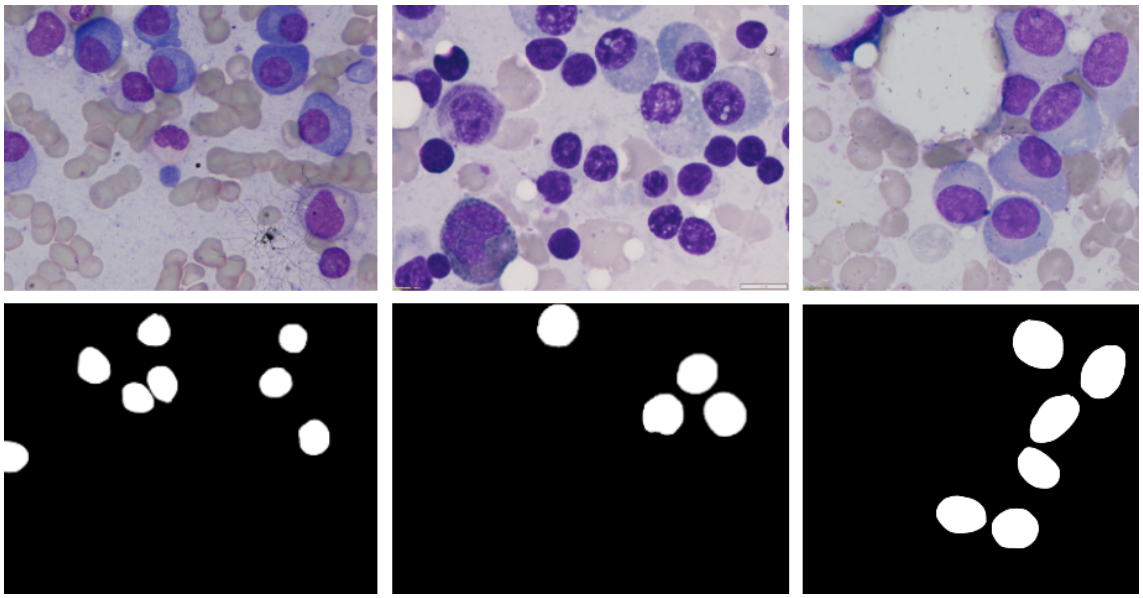

Image 4.7: *Examples of the new format of the samples in the SegPC set [27, 28, 29, 30]*

4.3 Data Augmentation

Due to the relatively small number of samples that are publicly available for testing medical image segmentation in the areas considered in this paper, it is necessary to augment the data. Techniques such as random rotation, shear, zoom, horizontal and vertical flip were applied to the training sets, resulting in a large increase in the number of samples, not exceeding the limits, in memory and running time, of the machines available to perform the experiments in this work. The Keras library and the Tensorflow framework were used as in the rest of the work. Examples of the augmented data are shown in Figure 4.8 and the numbers of new samples are shown in Table 4.3 .

Dataset	Images	Train	Valid	Test
CVC-ClinicDB	860	737 (+50%)	61	62
Kvasir-SEG	1496	1296 (+62%)	100	100
2018 Data Science Bowl	918	784 (+46%)	67	67
SegPC	745	645 (+62%)	50	50
Polyps Dataset	2356	2033 (+58%)	161	162
Cells Dataset	1664	1429 (+60%)	117	117

Table 4.3. *Number of samples in augmented datasets*

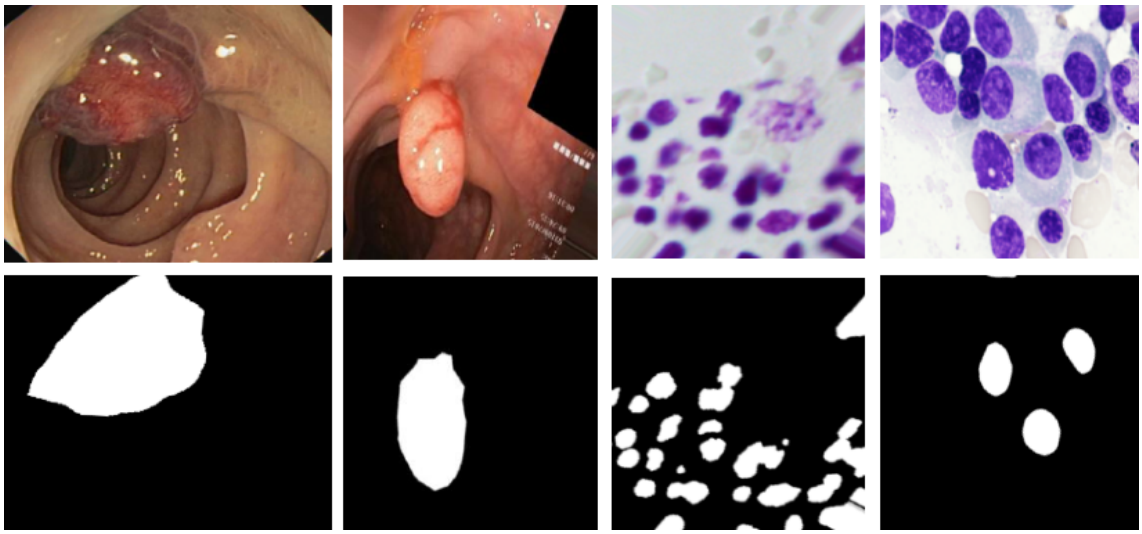

Image 4.8: *Examples of samples from new augmented datasets, from left to right, CVC-ClinicDB, Kvasir-SEG, 2018 Data Science Bowl, SegPC*

Chapter 5

Implementation

This chapter presents the models along with their architectures and hyperparameters as used in the experiments of this thesis. Several of the most widely used models in the field of medical image segmentation (and beyond) were chosen as well as some State-of-the-Art models. In addition, further details of the implementation such as the evaluation metrics as well as the computational system on which the work was performed are reported. The whole work has been implemented with the Keras library and the Tensorflow framework.

5.1 Implementation with different models

5.1.1 UNet

The UNet [7] is perhaps the most basic of the models involved in the challenge of medical image segmentation, since it formed the basis for the design of most of the models below, and is furthermore very successful. It is a deep convolutional network designed to address the problem of the small amount of data available by utilizing the data more efficiently.

The architecture, depicted in Figure 5.1, consists of a contractive path to capture the general context and a symmetric expansive path that allows for precise localization. The contractive path follows the standard architecture of a convolutional network. It consists of the iterative application of two convolutions of 3x3 each followed by a rectified linear unit (ReLU) and a max pooling 2x2 procedure with a 2 step downsampling. At each downsampling stage the number of feature channels is doubled. Each step of the expanding path consists of an upsampling of the feature map followed by a 2x2 convolution that reduces the feature channels by half, a convolution with the corresponding truncated feature map from the contractive path, and two 3x3 convolutions followed by a ReLU. At the end there is a final layer with a 1x1 convolution to map each feature vector of size 64 to the desired number of classes (in our case, the classes are two). In total, the network has 23 convolutional layers. The important differentiation of this model is the large number of feature channels in the extended path part that allows the network to transmit information about the general context to layers with higher resolution.

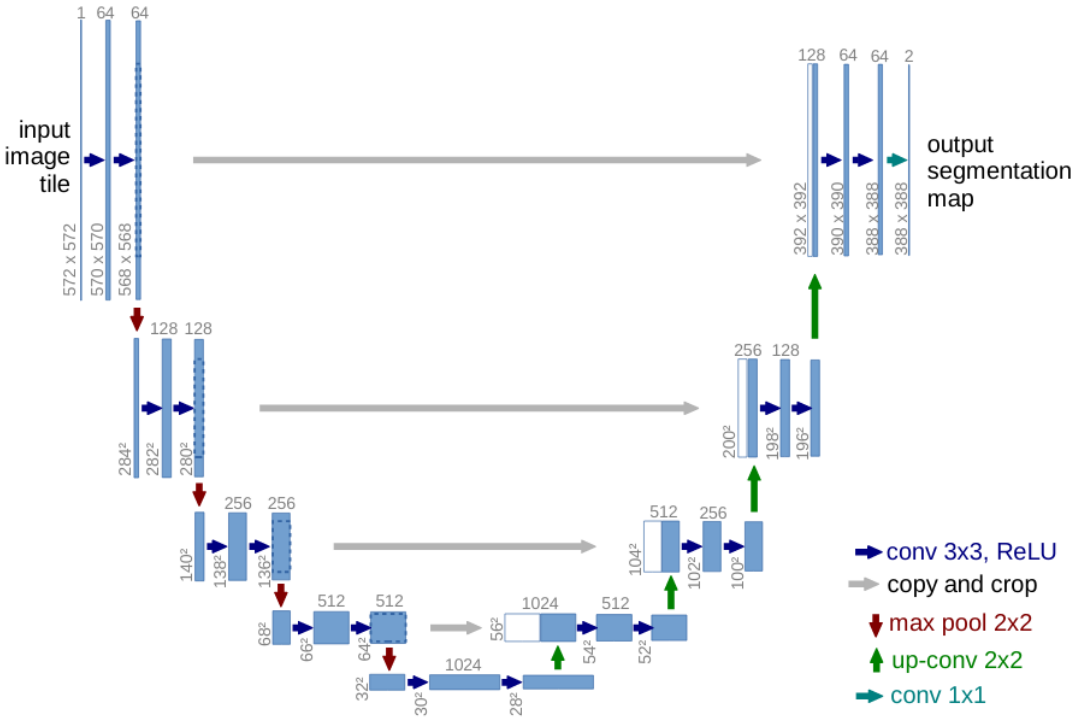

Figure 5.1. *The architecture of the UNet model [7]*

5.1.2 VNet

VNet [8] is a deep convolutional network designed for the field of medical data segmentation, inspired by UNet, but unlike the other convolutional models presented, this one works on three-dimensional (3D) data (MRI scans) instead of two-dimensional (2D) images (however in our case the input data are images, so a modified form of VNet was used).

Its architecture, shown in Figure 5.2, is quite similar to that of UNet as it consists of a contractive path and an expansive path. The left part of the network is divided into different stages operating at different resolutions and consisting of one to three convolutional layers. As in [62] the stages are modified to learn a residual function, in particular the input of each stage in addition to passing through the convolutional layers is also passed to the output of the stage itself. This modification reduces the convergence time.

Kernels of size 5x5x5 are used in the convolutions of each stage. From one stage to the next, the data resolution is reduced by convolution with kernels of size 2x2x2 and step 2. Because this operation halves the size of the feature maps, a convolutional layer is proposed to replace the pooling layers. It turns out, that this change leads to networks with smaller memory requirements, since the de-convolution process is simpler than the un-pooling process. Each stage of the leftmost part of the network computes twice as many features as its predecessor. PReLU and non-linear layers are used in all stages. The right part of the network extracts features and by exploiting the feature maps of the previous lower resolution stages finally produces a volumetric segmentation which is converted at the output to a probabilistic foreground and background segmentation via a

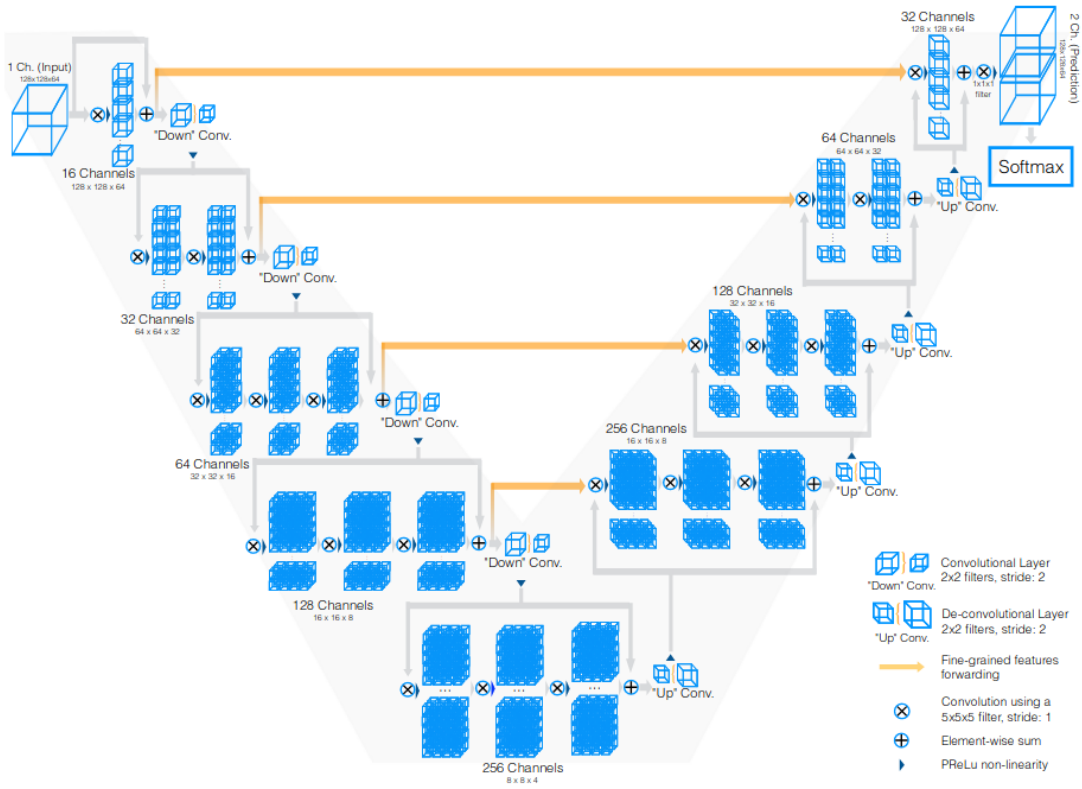

Figure 5.2. *The architecture of the VNet model [8]*

softmax function applied to the voxels. Similarly to the left piece, convolution is performed between stages to increase the data size. Similarly with UNet, features extracted from the first stages of the left part of the network are passed to the right piece, and thus fine-grained information that would otherwise be lost in the contractive path is saved. These connections further help the model to converge faster.

5.1.3 R2U-Net

In the paper [9], two models are presented, RU-Net which is a Recurrent Convolutional Network (RCNN) based on UNet, and R2U-Net which is a Recurrent Residual Convolutional Network (RRCNN) based on UNet. These models combine the strengths of UNet, Residual Networks (RNNs) and RCNNs, and thus show several advantages in the medical image segmentation process. In particular, the residual pieces contribute to faster training [62], the feature accumulation offered by RRCNNs [63] ensures better feature representations, and finally a reduction of network parameters is favored compared to UNet.

The architecture of the models is shown in Figure 5.3. More specifically, the architecture depicted is that of RU-Net, and the difference with R2U-Net is that along with the iterative convolutional layers (RCL) there are residual units (residual units). We consider four different architectural variants within the [9] publication, shown in Figure 5.4 , and highlight the differences of the new proposed models compared to their predecessors.

The first of these is the UNet one with forward convolutional layers and feature fusion,

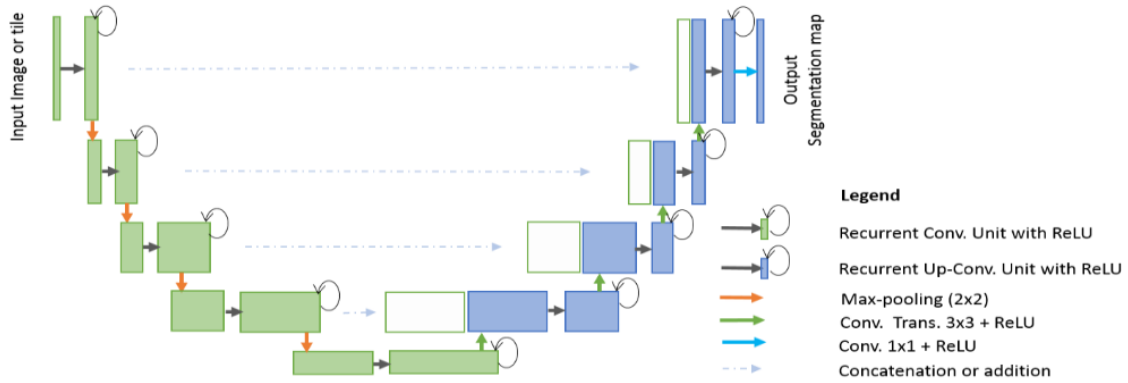

Figure 5.3. *The architecture of the R2U-Net model [9]*

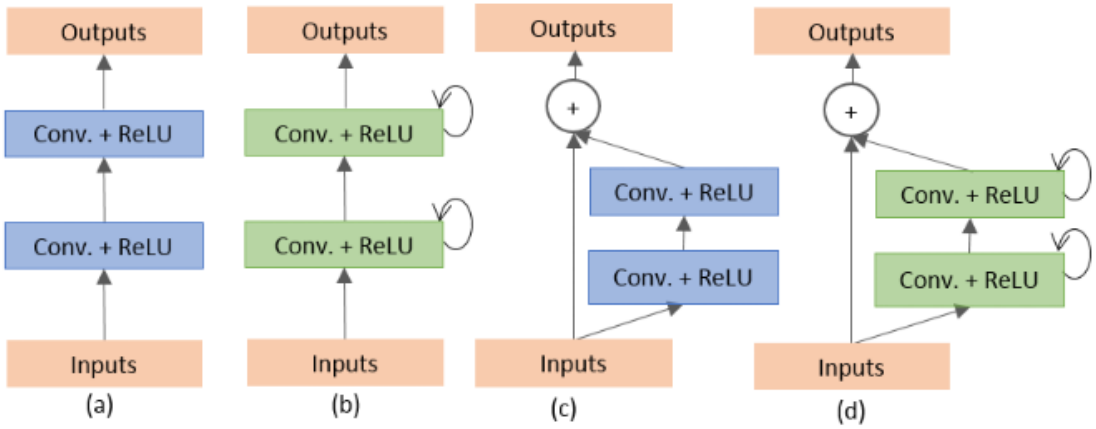

Figure 5.4. *Different architectures tested (a) Forward convolutional units, (b) Recurrent convolutional block (c) Residual convolutional unit, and (d) Recurrent Residual convolutional units (RRCU) [9]*

instead of the cropping method that had been applied to the original form of UNet (Figure 5.3(a)) and the second is the UNet one with forward convolutional layers with residual connection, commonly called ResUNet (Figure 5. 3(c)). The third architecture is the UNet one with forward recurrent convolutional layers called RU-Net (Figure 5.4(b)). The fourth and last architecture is the UNet with recurrent convolutional networks with residual connection, called R2U-Net (Figure 5.4(d)).

In the implementation of RU-Net and R2U-Net, feature concatenation from the encoder to the decoder is applied. According to the above analysis, it is observed that the new models are similar in principle to the well-known UNet, except that recurrent convolutional layers (RCL) with or without residual units are used instead of normal forward convolutional layers. This creates deeper models which are also more efficient due to the accumulation of features from different stages. In addition to shorter convergence time, these new architectures lead to better training and better results, stronger feature representations, without increasing the model complexity and the number of parameters.

5.1.4 Attention UNet

Attention UNet [10] is a gated attention model for medical images that automatically learns to focus on target structures of different sizes and shapes. Models that have these gates learn to ignore insignificant regions and focus on the important features depending on the task they are performing. In addition they remove the need for external CNN modules that deal with localization, reducing complexity. Attention gates can be easily integrated into CNN models such as UNet without requiring additional resources, at the same time increasing the sensitivity and accuracy of the model.

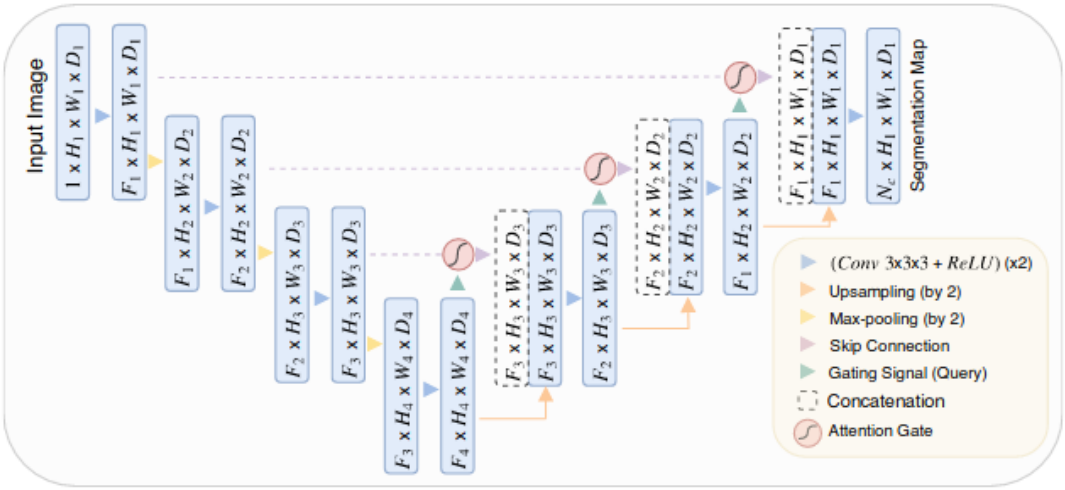

Figure 5.5. *Attention UNet model architecture [10]*

The architecture of the Attention UNet model is shown in Figure 5.5. The images that constitute the input are filtered and downsampled by a factor of 2 at each stage of the part of the model that performs the encoding. The attention gates filter the features that pass through the skip connections. The architecture of the attention gates is shown in Figure 5.6 . These gates are integrated into the UNet architecture in order to emphasize the important features that pass through the skip connections.

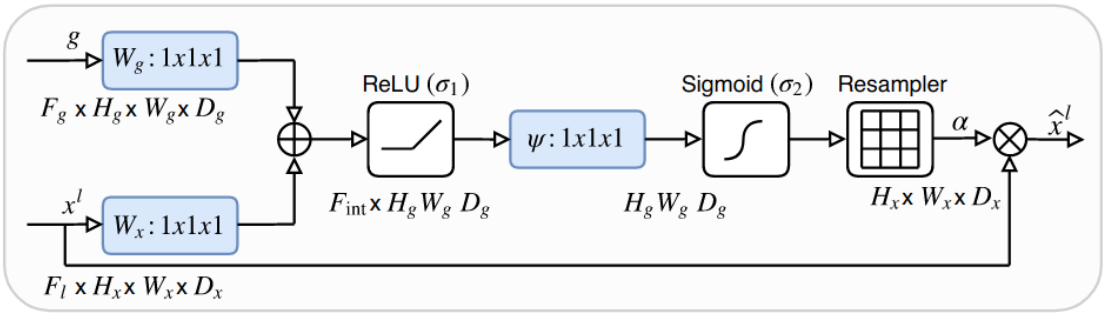

Figure 5.6. *The architecture of Attention Gates [10]*

Information from coarse scales is exploited in the gates to clarify noisy responses to skip connections. This occurs before concatenating to preserve only relevant activations.

In addition, the gates filter neuron activations in both the forward and the backward pass. Deep supervision is used to ensure that attention units at different scales have the potential to influence responses.

5.1.5 ResUNet

ResUNet [11] is another model inspired by UNet. It combines the strong features of residual learning and UNet and was designed for the task of road segmentation. As mentioned before, the residual nature helps a lot in training the models. What differentiates it from UNet is the use of residual units instead of simple neural units as the basic block. In addition, the part involving cropping is removed since it is not considered necessary for the challenge it faces.

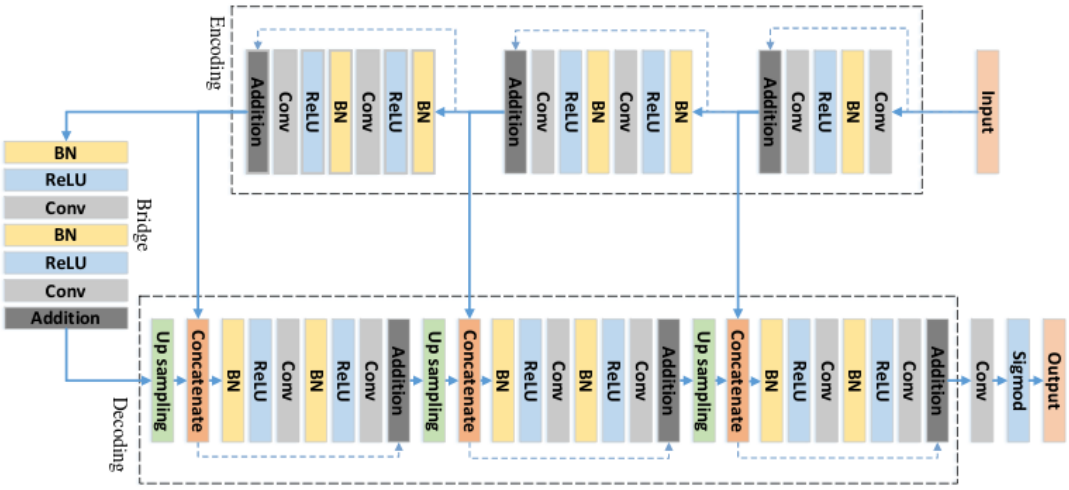

Figure 5.7. *ResUNet model architecture [11]*

The ResUNet architecture, shown in Figure 5.7, consists of 7 layers and is composed of 3 parts, the encoder, the bridge and the decoder. The first part encodes the input into compact representations, while the last part restores the representations to pixel level categorization. The bridge joins the two parts. All 3 parts consist of residual blocks which in turn consist of two 3x3 convolutional blocks and an identity mapping. The convolutional blocks consist of a batch normalization layer, a ReLU activation layer and a convolutional layer. The identity mapping connects the input and output of each block. The encoding path has 3 residual units, in which no pooling is performed but step 2 is introduced in the convolution to reduce the feature map to half the size. Similarly, the decoding path consists of 3 residual units, among which oversampling of the features and pooling with the features of the corresponding stage of the encoding path is performed. At the last level of the decoding path there is a final 1x1 convolution and a sigmoid activation in order to project the features to the final class level. Compared to the 23 layers of UNet, ResUNet consists of 15 convolutional layers.

5.1.6 ResUNet++

ResUNet++ [12] was introduced as an improved version of ResUNet for the task of segmenting colonoscopy images. This model performs semantic segmentation and utilizes residual units, squeeze and excitation units, Atrous Spatial Pyramidal Pooling (ASSP) and attention blocks. Residual blocks transfer information between layers creating a deep network and improving the dependencies between channels and reducing the computational cost.

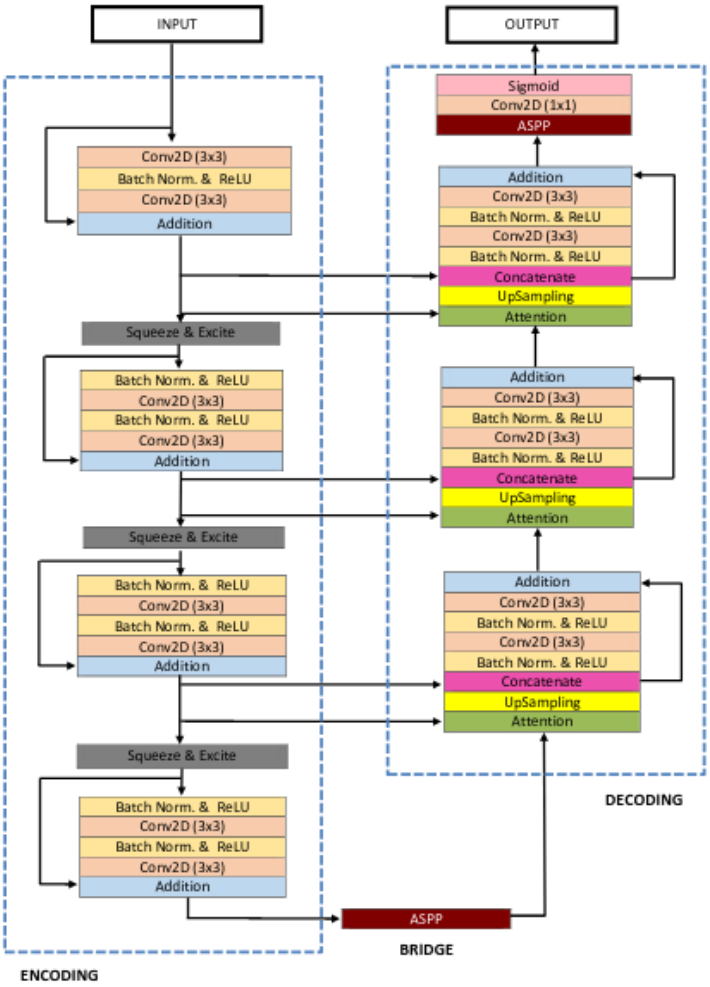

Figure 5.8. *ResUNet++ model architecture [12]*

The model architecture, depicted in Figure 5.8, consists of an initial block followed by 3 encoding blocks, a bridge to ASPP, and 3 decoding blocks. As above, the residual units are a combination of batch normalization, ReLU activation, and convolutional layers. Each encoding block consists of two 3x3 convolutional blocks and an identity mapping. The output of the encoder blocks is first passed through a squeeze-and-excitation block, the purpose of which is to ensure that the network can improve its sensitivity to relevant features and degrade non-essential features, and additionally improves the generalization of the model. The data is then passed through the ASPP which acts as a bridge and resamples the features at various scales, thus providing important information to the

model and maximizing the filters' field-of-view, i.e. the maximum data range to which the filters can be exposed. Similarly, the decoding path also consists of residual blocks. Before each block, the attention block increases the efficiency of the features. This is followed by nearest neighbor oversampling, and a concatenation with the features of the corresponding stage of the encoding path. The output is then passed through ASPP and finally through a 1x1 convolution and a sigmoid activation, which generates the final segmentation map.

5.1.7 ResUNet-a

ResUNet-a [13] is a deep fully convolutional network (FCN) designed for semantic segmentation of very high resolution aerial images. ResUNet-a features a UNet-like encoder-decoder backbone, combined with residual connections, atrous convolutions [43] and pyramid scene parsing pooling [64]. ResUNet-a serially generates the object contour, the distance transformation of segmentation masks, segmentation masks, and a color reconstruction of the input.

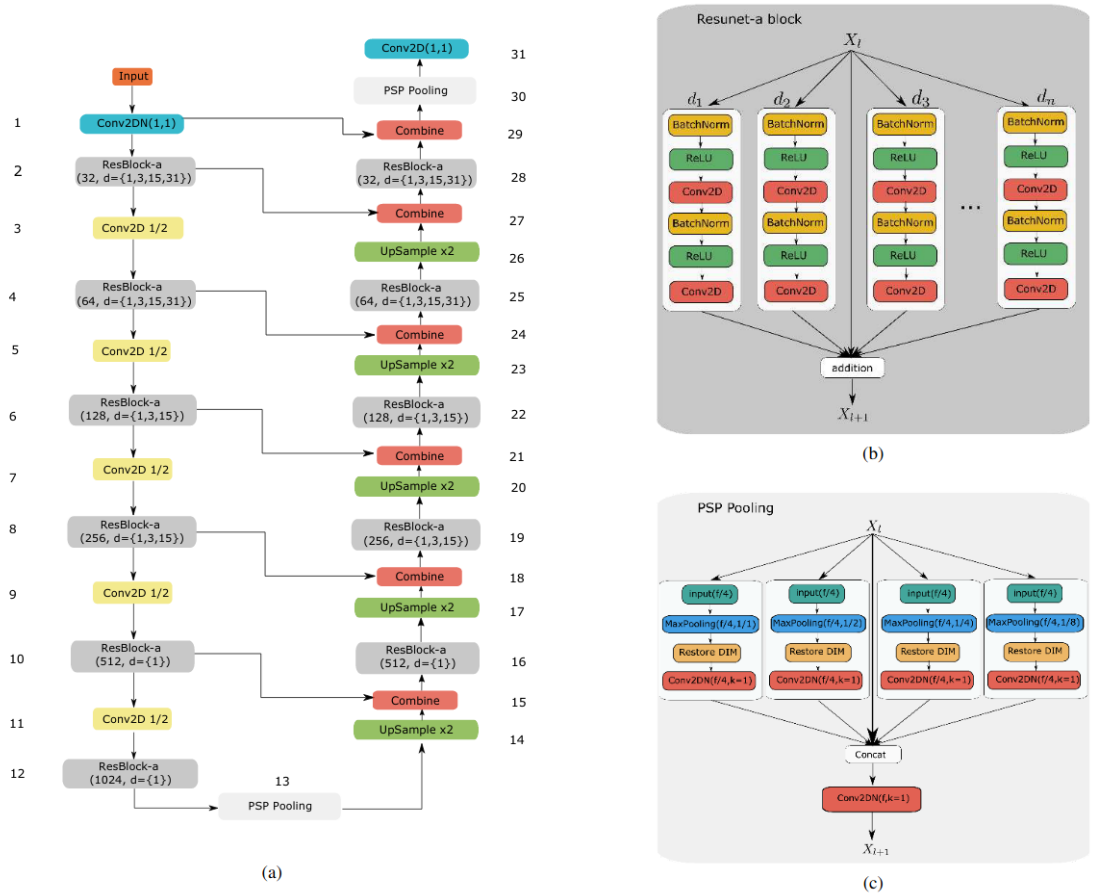

Figure 5.9. *(a) The architecture of ResUNet-a (b) The building block of ResUNet-a (ResBlock-a) (c) The pyramid scene parse pooling layer (PSPP) [13]*

The network architecture, shown in Figure 5.9, consists of the UNet encoding and decoding paths, where the UNet blocks are replaced by residual convolutional blocks, which solve the problem of vanishing and exploding gradients. In addition, multiple atrous con-

volutions with different expansion rates are performed within the residual blocks helping to improve understanding at multiple scales. To improve the model performance by including background information, pyramid scene parsing pooling is used to improve the model performance. Two ResUNet-a models, d6 and d7, which differ in their depth are presented, in one the encoder consists of 6 ResBlock-a and a PSP Pooling layer and the other of 7 ResBlock-a. The input is first passed through a 1x1 convolution to increase the number of features to the desired filter size. This is then followed by a sequence of ResBlock-a. Up to 3 parallel atrous convolutions were used in each block in addition to the standard two convolutions of the residual blocks. The output is then added to the input. From each block to the next, the output is downsampled via a 1x1 convolution with a stride of 2. At the end of both the encoder and decoder there is a pyramid scene parsing pooling layer. In this layer the original input is split into 4 chunks in feature space, then max pooling is performed in sequential bisections, in 1,4,16,64 chunks. In the decoder, upsampling is done using the nearest neighbor technique, followed by a 1x1 convolution and batch normalization. Layers of the encoder and decoder are combined through the Combine layer, and go through a convolution that brings the number of features to the desired value.

5.1.8 TransUNet

TransUNet [14] is a model that combines transformers and UNet, and is a possible alternative to the task of medical image segmentation. While the u-architecture with convolutional networks has been prevalent in the medical image domain, it presents some limitations in modeling long-range dependencies [65]. Transformers have emerged as alternative architectures with internal global attention mechanisms, with the drawback that they may lead to low localization capability due to lack of low-level details. The transformer in TransUNet encodes tokenized image patches from a convolutional neural network (CNN) feature map as an input sequence to extract global contexts. The decoder upsamples the encoded features combined with the high-resolution features of the CNN to achieve accurate localization. It is shown that transformers can function efficiently as encoders, combined with the UNet model that enhances details by recovering localization data.

The architecture of TransUNet is illustrated in Figure 5.10 . Its differentiation from the classical u-architecture can be seen in the encoding path, which has been replaced by a hybrid transformer-CNN module. The CNN is used to extract features of the input. Initially, tokenization of the input into a sequence of flattened 2D patches is performed. The vector patches are projected into latent space. To encode the spatial information of the patches, the position of the embeddings is learned. The transformer-encoder consists of layers of Multihead Self-Attention (MSA) and Multi-Layer Perceptron (MLP) blocks. Patch embedding is applied to 1x1 patches derived from the CNN feature map instead of the input images. This technique is preferred because it helps to exploit the high-resolution features of the CNN in decoding, and furthermore leads to better results than with a simple transformer without CNN. In the decoder part, the Cascaded Upsampler (CUP) is

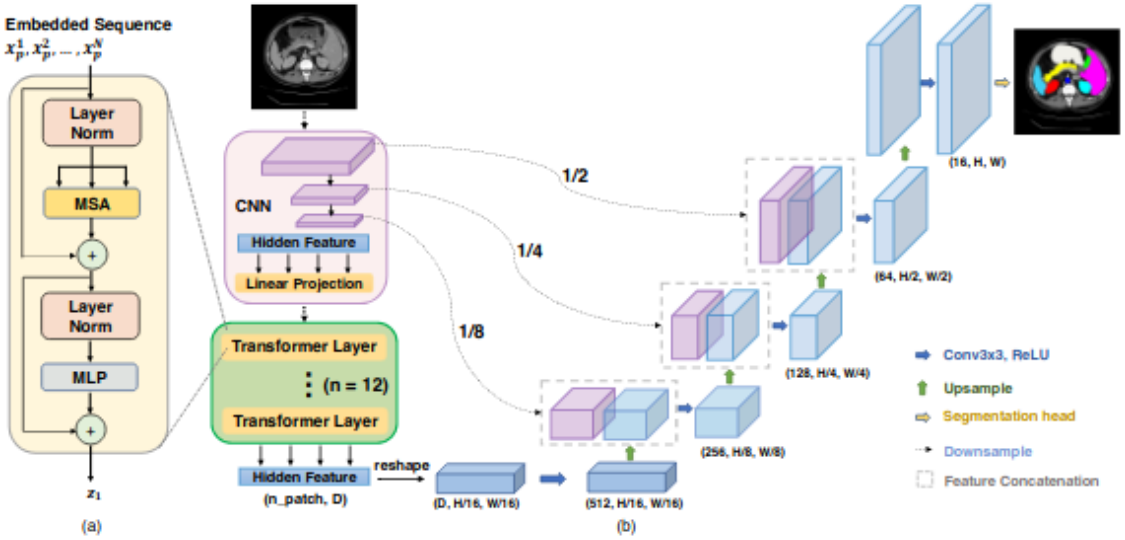

Figure 5.10. *(a) The architecture of the Transformer layer (b) The architecture of TransUNet [14]*

presented which consists of multiple upsampling steps and decodes the features. Each block consists of a 2x upsampling process, a 3x3 convolution layer and a ReLU layer. The CUP together with the hybrid Transformer-CNN encoder form a u-architecture.

5.1.9 SwinUNet

SwinUNet [15] is another u-architecture model based on the use of transformers that attempts to address the problem of long-range dependencies that exists in convolutional networks. The transformers in this model are hierarchical Swin Transformers [66] with shifted windows as encoders which extract the features, and symmetrically as decoders with patch expanding layer to perform upsampling and resample the spatial resolution of the feature maps.

The architecture of SwinUNet is shown in Figure 5.11. It consists of the encoder, the bottleneck, the decoder and skip-connections. The basic block of the model is the Swin Transformer block. The input images are divided into non-overlapping patches, each of which is treated as a token and passed through the Swin Transformer blocks and patch merging layers of the transformer-encoder that produces the deep feature representations. The patch merging layers are responsible for downsampling and increasing the dimensions.

As for the decoder part, the extracted features are upsampled by the decoder with patch expanding layer, which achieves upsampling and feature dimension increase without convolution and interpolation, and are combined with the multiscale features of the encoder via skip-connections. A final linear projection layer converts the upsampled features into the final segmentation map.

Swin Transformer blocks, illustrated in Figure 5.12, are based on shifted windows, and consist of a Layer Norm layer, a multi-head self-attention module, residual connections,

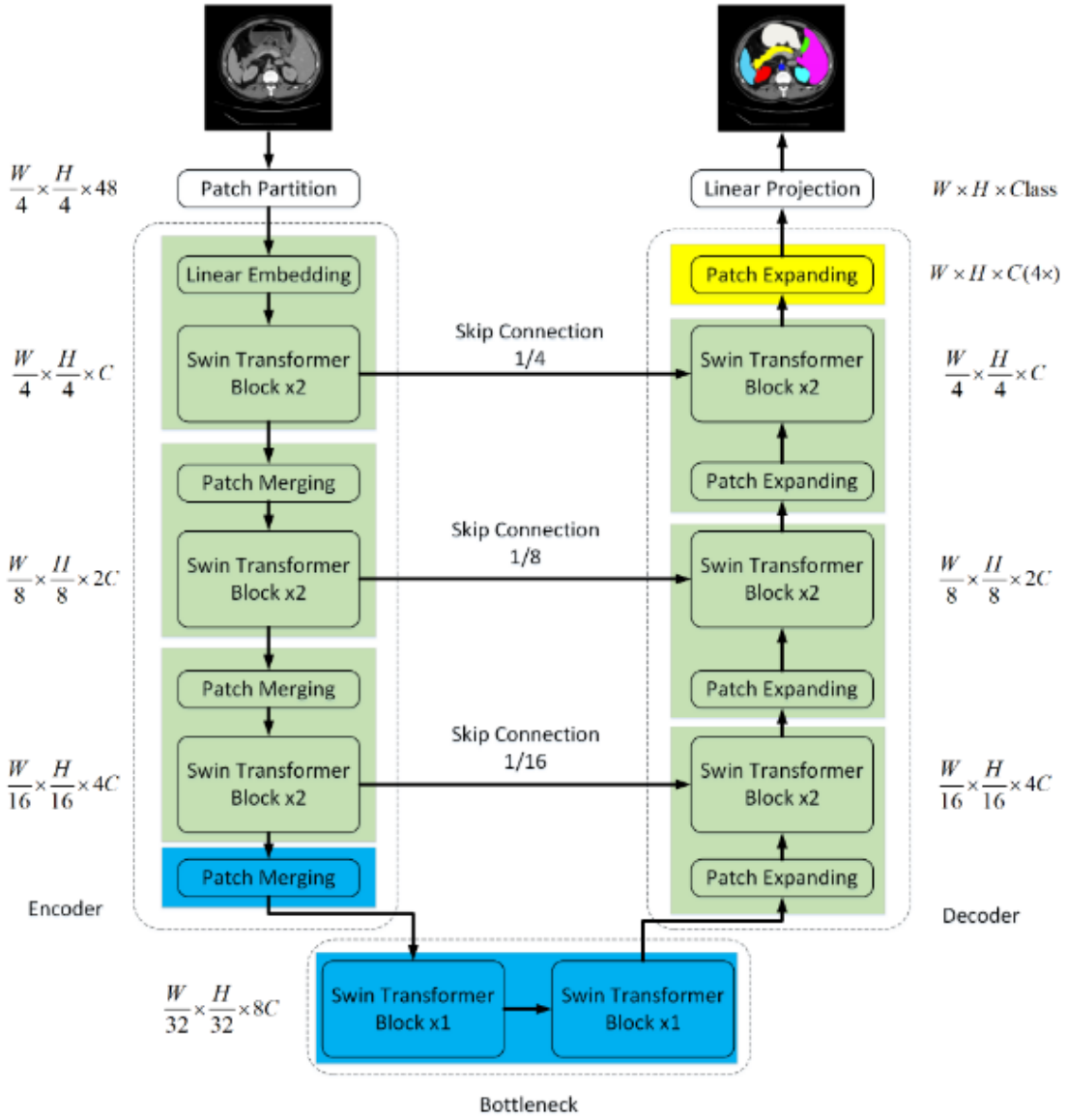

Figure 5.11. *SwinUNet model architecture [15]*

and a 2-layer Multi Layer Perceptron (MLP) with GELU nonlinearity. The window-based multi-head self-attention module (W-MSA) and the shifted multi-head self-attention module (SW-MSA) are applied to the two consecutive blocks. In the encoder the tokenized inputs are passed through the 2 sequential Swin Transformer blocks to learn the representations. The patch merging layer reduces the number of tokens (2x downsampling) and increases the dimension of the features to twice the original. This process is repeated 3 times. The bottleneck consists of 2 consecutive Swin Transformer blocks. The symmetric decoder is also based on the Swin Transformer block and is combined with the patch expanding layer that upsamples the extracted deep features and reshapes the feature maps of neighboring dimensions into a higher resolution feature map while halving the feature dimension. Similarly to UNet, the skip-connections connect the multi-scale features of the encoder to the upsampled features. Shallow and deep features are merged

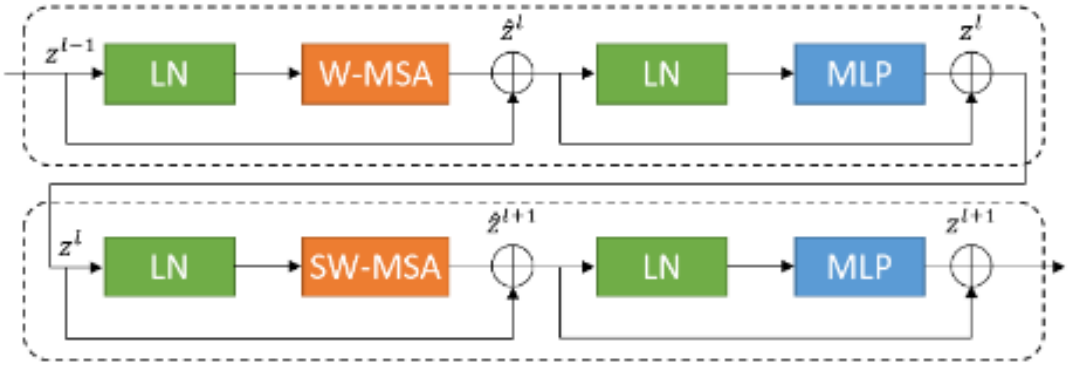

Figure 5.12. *Swin transformer block architecture [15]*

to reduce the loss of spatial information caused by downsampling.

5.1.10 DeepLabv3+

The spatial pyramid pooling and encoder-decoder architectures are widely used in deep neural networks in the semantic segmentation task, and show several advantages. In particular, the former have the ability to encode multi-scale contextual information through filters and pooling processes over features, while the latter can lead to more accurate object contours by incrementally restoring spatial information. The DeepLabv3+ [16] model combines the strong features of both architectures. DeepLabv3+ is an extension of DeepLabv3 [43], and features an additional simple but efficient decoder that improves the segmentation results in terms of object contours. Rich semantic information is encoded in the output of DeepLabv3, and atrous convolutions allow to control the density of features according to the available computational resources.

The atrous convolutions [67, 68, 69, 70] allow to control the feature resolution and adjust the field-of-view of the filter to capture multi-scale information. In the case of two-dimensional signals, for each location i in the output feature map y with a convolution filter w, atrous convolution is applied to the input feature map x as follows:

$$y[i] = \sum_k x[i + r \cdot k]w[k] \tag{5.1}$$

where the atrous coefficient r defines the sampling step of the input signal. The depthwise separable convolutions [71, 72, 73], are classical convolutions converted to depthwise convolutions followed by pointwise convolution, and significantly reduce the computational complexity. In particular, depthwise convolutions perform spatial convolutions independently for each input channel and pointwise convolutions combine the output of depthwise. The atrous separable convolutions exploited in this model are shown in Figure 5.14 and are derived from the depthwise convolutions with rate=2. They are shown to significantly reduce the computational complexity without reducing the quality.

The architecture of DeepLabv3+ is shown in Figure 5.13 . The DeepLabv3 model

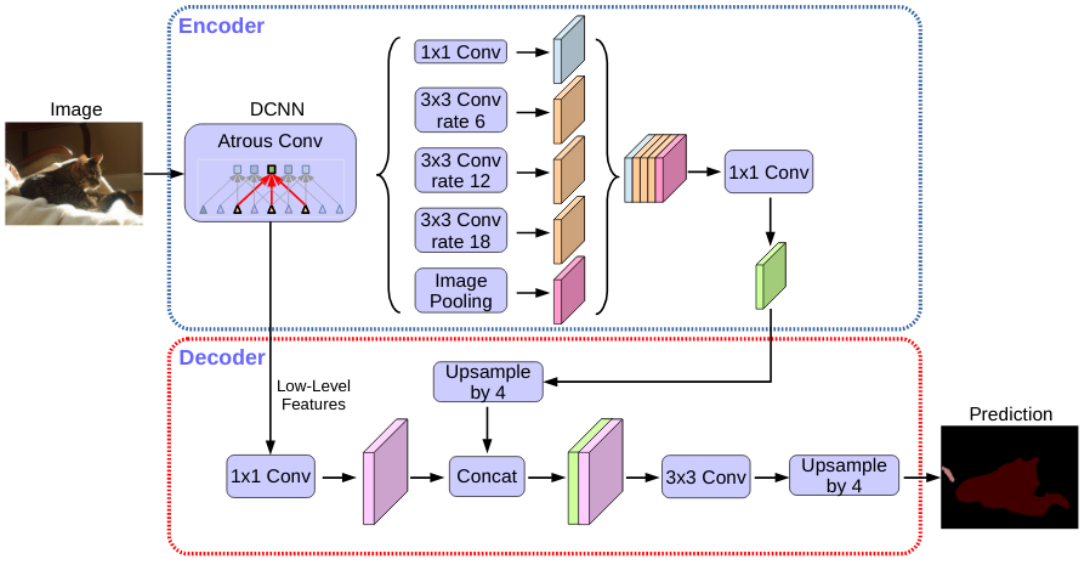

Figure 5.13. *DeepLabv3+ model architecture [16]*

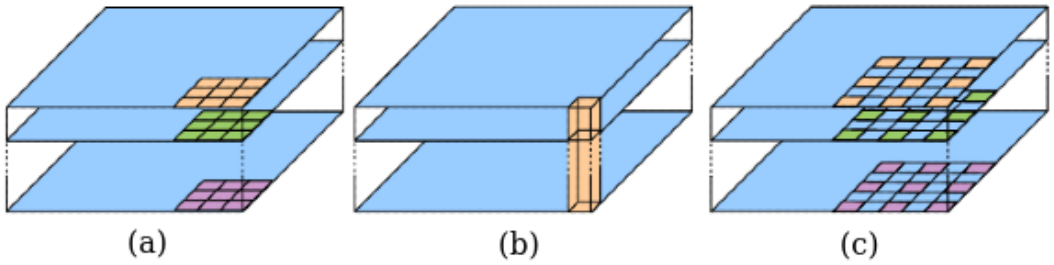

Figure 5.14. *(a) Depthwise convolution (b) Pointwise convolution (c) Atrous Depthwise convolution [16]*

serves as the encoder. It exploits atrous convolutions to extract features at a certain resolution computed by deep convolutional neural networks. For semantic segmentation, the output stride (which is the ratio of the spatial resolution of the input to the output) can have a value of 16 or even 8 for denser features, by removing the striding in the last or even the penultimate block and applying atrous convolution. The last feature map before logits is used as the output of the encoder, consisting of 256 channels and rich semantic information. On the decoder side, initially, the features from the encoder are bilinearly upsampled by a factor of 4 and fused with the corresponding low-level features from the network backbone that have the same spatial resolution. A 1x1 convolution is applied to the low-level features to reduce the number of channels which is quite large (e.g. 256 or 512) and may overwhelm the encoder features and make training difficult. After concatenation, some 3x3 convolutions are applied to increase the quality of the features and then another bilinear upsampling with a factor of 4. The quality of the results is shown to be much better when using output stride = 8 instead of 16, sacrificing computational complexity. In this paper, the DeepLabv3+ model used features ResNet50 [62] pretrained network on ImageNet [74] as a backbone.

5.1.11 MSRF-Net

Although methods based on convolutional neural networks have improved performance in the task of medical image segmentation, they show weakness in variable size object segmentation and are also trained on small and biased datasets. There are methods that implement multi-scale fusion, but they usually use complex models designed for the general task of semantic segmentation. Multi-Scale Residual Fusion Network (MSRF-Net) [17] is a new architecture specifically designed for medical image segmentation, which has the ability to exchange multi-scale and receptive field features using Dual-Scale Dense Fusion (DSDF) block. The DSDF block can exchange information robustly between different resolution scales, and MSRF-Net uses it within a subnetwork to achieve multi-scale fusion. This process allows maintaining resolution and spatial accuracy, improves information flow and propagation of both low and high level features, and achieves accurate segmentation results.

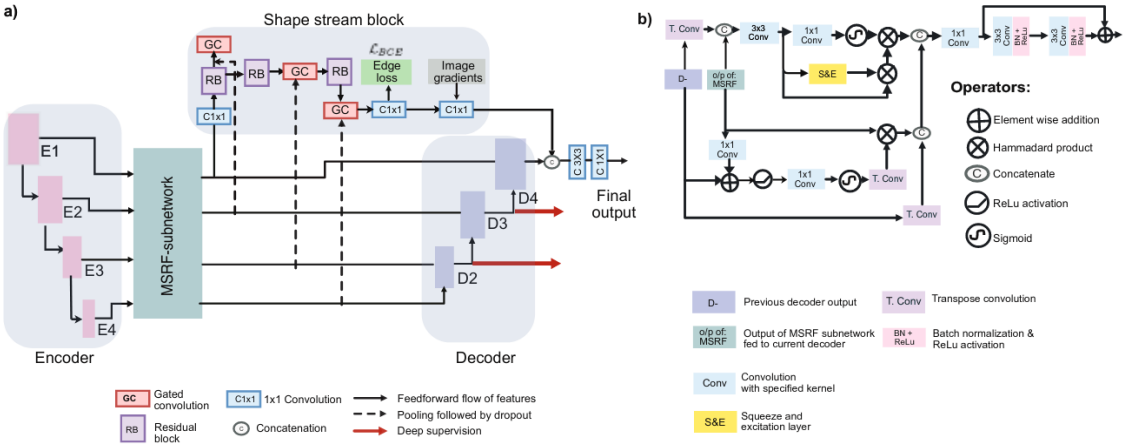

Figure 5.15. *(a) The architecture of MSRF-Net [17] (b) The architecture of the decoder*

The DSDF block, illustrated in figure 5.16(a), takes input from two different scales and through a residual dense block it exchanges information with other scales after each convolution layer. Iterative multi-scale fusion helps augment high-resolution feature representations with information from low-resolution representations. In addition, network residual layers allow redundant DSDF blocks to cease to have an effect, and only the relevant extracted features to contribute to the final segmentation map. In addition to the above modules there is also a complementary gated shape stream that leverages the combination of high and low level features to calculate shape contours accurately. An encoder feeds the MSRF subnetwork (figure 5.16(b)), consisting of multiple DSDF blocks, with feature representations. Then, layers of the decoder with skip-connections from the subnet and a triple attention mechanism process the fused feature maps together with the shape stream.

The architecture of the MSRF network is shown in Figure 5.15 and consists of encoding blocks, the MSRF subnet, a shape stream block, and decoding blocks. The encoding blocks consist of 2 consecutive convolutions followed by a squeeze-and-excitation unit which increases the representational power of the network by accounting for interde-

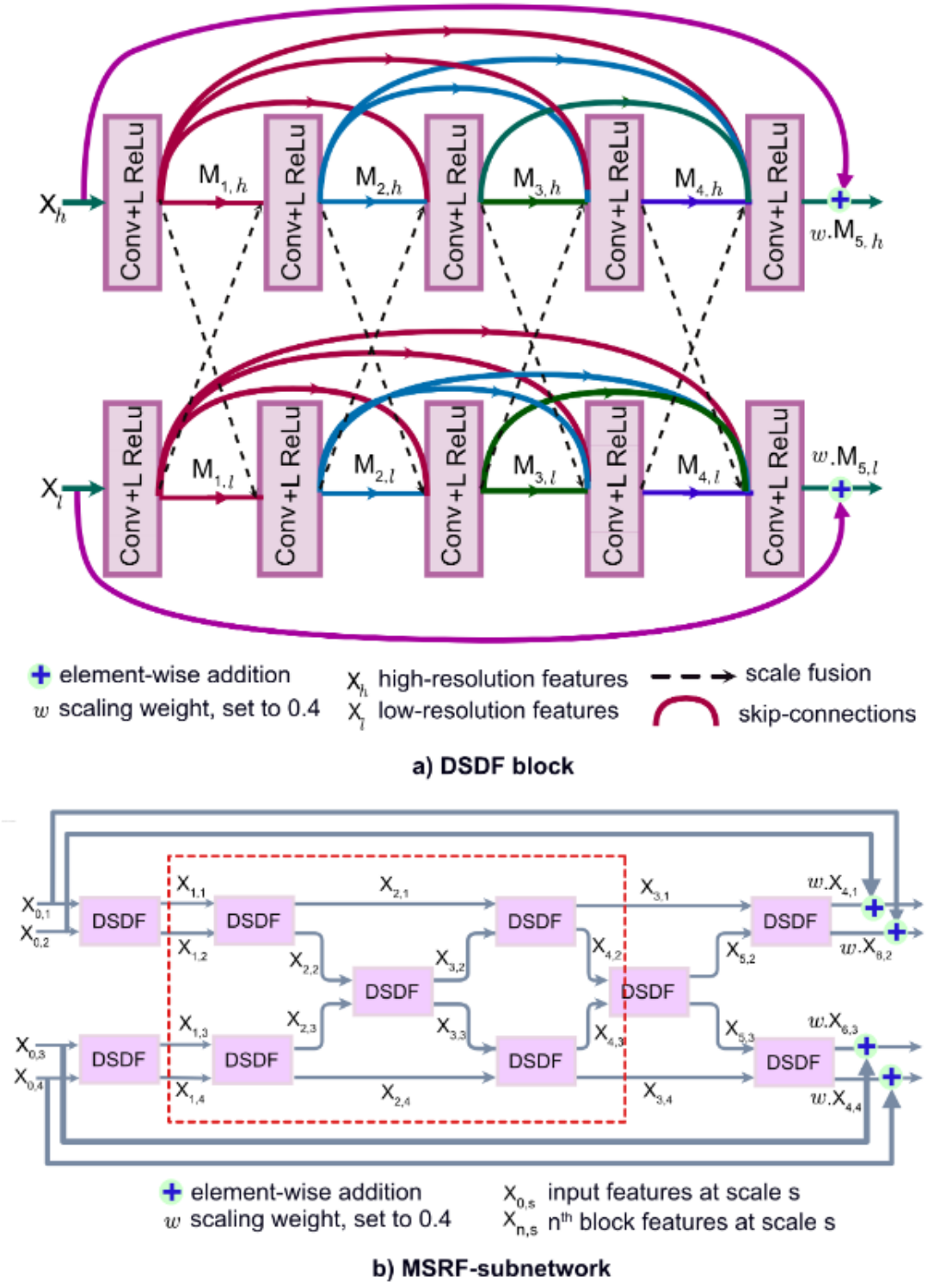

Figure 5.16. *(a) The DSDF block architecture (b) The MSRF subnet architecture [17]*

pendencies between channels. In the squeezing step the features are aggregated in the spatial dimension of the channels through global average pooling, while in the excitation step a collection of weights per channel is produced to express the dependencies between channels. At each stage of the encoder, max pooling is performed with a stride of 2 to

reduce the resolution, and dropout for the regularization of the model. The DSDF block, which helps to exchange information between scales, has 2 parallel streams for 2 different scales. If we call CLR(·) the process of a 3x3 convolution followed by a LeakyReLU activation then each flow has a densely connected residual block with 5 CLR(·) processes. The feature map $M_{d,h}$ of the output is calculated from the high-resolution input X_h as follows:

$$M_{d,h} = CLR(M_{d-1,h} \oplus M_{d-1,l} \oplus M_{d-2,h} \oplus \dots M_{0,h}), 1 \leq d \leq 5 \tag{5.2}$$

on the high resolution stream, and similarly for the low resolution stream as follows:

$$M_{d,l} = CLR(M_{d-1,l} \oplus M_{d-1,h} \oplus M_{d-2,l} \oplus \dots M_{0,l}), 1 \leq d \leq 5 \tag{5.3}$$

The operator $\oplus$ stands for concatenation and the symbols h,l stand for high and low resolution respectively. The output of each CLR(·) has k channels that define the growth factor which regulates the number of new features the layer can extract and forward to the rest of the network. In order not to increase the complexity of the model because the growth factor changes for each scale, only 2 scales are used at a time in the DSDF blocks. Additional residual learning and residual scaling are applied to avoid instability [75, 76]. So the final output of the DSDF block is expressed as:

$$X_r = w \times M_{5,r} + X_r, r \in [h, l], 0 \leq w \leq 1 \tag{5.4}$$

The MSRF subnet consists of several DSDF blocks and achieves global multi-scale context through the dual-scale fusion mechanism. First resolution/scale pairs are delimited and fed into the corresponding DSDF blocks. Starting from the first layer consisting of 4 resolution scales, the DSDF(·) function performs feature fusion between scales in the DSDF block. Already after the fourth layer, an exchange of characteristics has been achieved at all scales. This method can support more than 4 scales. The output of the last layer of the subnet is scaled by a factor w and added to the input.

The model features a gated shape stream [77] which performs shape prediction by exploiting high-level representations extracted from DSDF blocks. The shape stream feature maps are defined as S_l where l is the number of layers, and X is defined as the output from the MSRF subnetwork, which undergoes bilinear interpolation to match the dimensions of S_l. The attention map in gated convolution is calculated as follows:

$$a_l = \sigma(Conv_{1\times1}(S_l X)) \tag{5.5}$$

where $\sigma(\cdot)$ is the sigmoid activation. S_{l+1} is calculated as:

$$S_{l+1} = RB(S_l \times a_l) \tag{5.6}$$

where RB is a residual block with 2 CLR(·) processes followed by a skip-connection. The output shape stream is joined with the gradients of the input images and mixed with the original segmentation stream before the last CLR(·) to increase the spatial accuracy.

The decoder block has skip-connections from the MSRF subnet and the output of the previous decoder. In this piece there are 2 attention mechanisms, the first applies channel and spatial attention and the second uses a gating mechanism. A squeeze-and-excitation block performs the calculation of scale factors per channel $X_{a_{se}}$ and additionally calculates the spatial attention after the channels are reduced through a 1x1 convolution. The sigmoid activation places the values between 0 and 1 and thus forms the activation map X_{a_s}. Thus the output of spatial and channel attention is expressed as follows:

$$D_{sc} = (X_{as} + 1) \odot X_{a_{se}} \tag{5.7}$$

where the coefficient $\odot$ is the Hadamard product. Regarding the gated attention mechanism [10], the attention coefficients are calculated as follows:

$$D_{AG} = \Omega(\sigma(\Psi(\partial(X) + \phi(D^{-})))) \tag{5.8}$$

where X the features from MSRF-Net, D^{-} the output of the previous decoding block, $\partial(\cdot)$ the convolution operator with step 2 and kernel size 1 and G channel outputs, $\phi(\cdot)$ the convolution operator with step 1 and kernel size 1x1, $\Psi(\cdot)$ the convolution operator with 1x1 kernel applied to the combined features $\partial(\cdot)$ and $\phi(\cdot)$, $\sigma(\cdot)$ the sigmoid activation operator, and $\Omega(\cdot)$ the transpose convolution operator. The D_{AG} coefficients contain contextual information and identify the target regions and structures of the image. Through the process $\tilde{D}_{AG} = D_{AG} \oplus X$ irrelevant features are omitted. The coefficients $\tilde{D}_{AG}$ are updated as follows:

$$\tilde{D}_{AG} = \tilde{D}_{AG} \oplus \Omega(D^{-}) \tag{5.9}$$

Finally the output of the triple attention decoder block is

$$D_a = D_{sc} \oplus \tilde{D}_{AG} \tag{5.10}$$

and followed by 2 CLR(·) procedures.

5.2 Evaluation Metrics

5.2.1 Loss Functions

One of the prevailing loss metrics for the task of image segmentation is the Binary Cross Entropy loss [78] described by the formula:

$$\mathscr{L}_{BCE} = (1 - y)log(1 - \hat{y}) + ylog\hat{y} \tag{5.11}$$

where y is the ground truth and $\hat{y}$ is the predicted value. It is a special case of the Cross Entropy metric which calculates the performance of a classification model that produces probabilities 0 to 1 as output. It treats both types of errors equally (false positive, false negative) and increases as the calculated probability moves away from the real one price. In addition to Binary Cross Entropy, there is also the Dice Loss [79] which is a measure

of the overlap between a ground truth image and an image generated by the model and is described by the formula:

$$\mathscr{L}_{DCS} = 1 - \frac{2y\hat{y} + 1}{y + \hat{y} + 1} = 1 - DC \tag{5.12}$$

where y and $\hat{y}$ are defined similarly to before and DC is the Dice Coefficient defined as follows:

$$DC = \frac{2|A \cap B|}{|A| + |B|} \tag{5.13}$$

where $|A \cap B|$ the number of common elements of the sets A, B and $|A|, |B|$ the number of elements in the sets A, B respectively. The numerator of both relationships essentially refers to the shared activations between the real image and the model-generated image, and the denominator refers to the number of activations in the two images separately. In all the models presented above, both error metrics were used in illustrative experiments, and for each the error metric that led to the best results was chosen. A combination of metrics defined below [17] was used in the MSRF-Net model:

$$\mathscr{L}_{MSRF} = \mathscr{L}_{comb} + \mathscr{L}_{comb}^{DS^0} + \mathscr{L}_{comb}^{DS^1} + \mathscr{L}_{BCE}^{SS} \tag{5.14}$$

where $\mathscr{L}_{comb} = \mathscr{L}_{DCS} + \mathscr{L}_{BCE}$, $\mathscr{L}_{comb}^{DS^0}$ and $\mathscr{L}_{comb}^{DS^1}$ the metrics for the two outputs of deep supervision, and $\mathscr{L}_{BCE}^{SS}$ the BCE loss from the shape stream of the MSRF-Net model.

5.2.2 Accuracy Metrics

In addition to the loss functions that play an important role in training the models, there are also the accuracy metrics that serve to evaluate the models. The metrics Dice Coefficient, (mean) Intersection Over Union, Precision and Recall were used in this work.

The metric (mean) Dice Coefficient (DSC) [79] has been analyzed in the previous paragraph and the formula that describes it is (5.13). Its final value is obtained as the average value from the results of Dice Coefficient calculations for all pairs of real and generated images.

The metric (mean) Intersection Over Union (mIoU) [79] estimates the similarity between the images generated by the model and the real images, and as its name suggests it calculates the percentage of shared activations between the two types of images (Intersection) over the union of their activations (Union). The maximum value it can have is 1, and the lower the value, the worse the quality of the model results. It is calculated as follows:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{5.15}$$

where where $|A \cap B|$ the number of common elements of the sets A, B that constitute the activations of the real and computed images respectively, and $|A \cup B|$ the number of elements in union of the sets A, B. The final value of the mIoU metric is obtained as the average value of the IoU calculation results for all pairs of real and computed images.

The Precision metric [80] calculates the percentage of correct positive predictions to

all positive predictions, i.e.:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{5.16}$$

where True Positives are the positive predictions that coincide in the real and computed images and False Positives are the positive predictions in the computed images that do not exist in the real images. It can have values from 0 to 1 and the smaller its value, the worse the performance of the model.

Finally, the Recall metric [80] calculates the percentage of correct positive predictions to the total number of correct positive predictions and false negative predictions, namely:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{5.17}$$

where True Positives are defined as above and False Negatives are the negative predictions in the computed images that are positive in the real images. It is quite similar to the Precision metric but emphasises more on missed positive predictions. The metric that was used to evaluate the models during training (decision whether the model at the particular time will be saved or not) is the Dice Coefficient.

5.3 Implementation Details

5.3.1 Model parameters

In the experiments of this thesis, the size of images and masks from the 4 datasets (CVC-ClinicDB, Kvasir-Seg, 2018 Data Science Bowl, SegPC) was set to 256x256 in most models. Additionally for the CVC-ClinicDB and SegPC sets batch size = 8 was used, in Kvasir-Seg and the 2018 Data Science Bowl batch size = 16 was used in most models. Some models (e.g. ResUNet-a) showed large memory requirements that exceeded the available resources with the given parameters, so dimensions 128x128 and batch size=4 were chosen in order to complete the training properly. Adam [81] was used as optimizer in all models, with learning rate = 0.0001. The number of epochs was set to 200, yet all models reach their maximum performance well before the last epoch. Open access code was used for all models presented. Parameters were chosen based on the literature, and intuition where the literature did not provide the answer.

For the MSRF-Net model the scaling factor (w) for the DSDF block and the MSRF subnet has a value of 0.4, and the growth factor (k) for the resolution pairs in the DSDF has values of 16, 32 and 64.

The DeepLabv3+ model has as its backbone a ResNet50 model which is pretrained on the ImageNet dataset and the low level features from the $conv4_block6_2_relu$ layer of the backbone are used.

In the UNet model the parameters are as follows: 5 down and upsampling levels, 2 convolutional layers for each downsampling level, 1 convolutional layer for each upsampling level, ReLU activation, Sigmoid output activation, Max pooling, upsampling with reflective padding.

In the VNet model the parameters are as follows: 5 down and upsampling levels, the number of convolutional levels of the residual path increases from 1 to 3 with downsampling levels (and symmetrically decreases with upsampling levels), PReLU activation, Sigmoid output activation.

In the Attention UNet model, the parameters are as follows: 4 down and upsampling levels, 2 convolutional layers per each downsampling and upsampling level, ReLU activation, additive attention, ReLU attention activation, Sigmoid output activation, Max pooling and upsampling with reflective padding.

In the R2U-Net model the parameters are as follows: 4 down and upsampling levels, 2 recurrent convolutional layers with 2 iterations per down and upsampling level, ReLU activation, Sigmoid output activation, Max pooling and upsampling with reflective padding.

In the ResUNet-a model the parameters are as follows: 5 downsampling levels followed by an Atrous Spatial Pyramid Pooling (ASPP) layer with 256 filters, 5 upsampling levels followed by an ASPP layer with 128 filters, dilation rates with values 1,3,15, 31, ReLU activation, Sigmoid output activation and upsampling with reflective padding.

In the TransUNet model the parameters are as follows: 4 down and upsampling levels, 2 convolutional layers per down and upsampling level, 2 transformer blocks, 2 attention heads, 3072 MLP nodes per transformer, 768 embedding dimensions, Gaussian Error Linear Unit (GeLU) activation for MLP, ReLU activation, Sigmoid output activation, Max pooling and upsampling via bilinear interpolation.

In the SwinUNet model the parameters are as follows: 4 down and upsampling levels, 2 Swin Transformers per down and upsampling level, patch size (2,2), embedded patches in 64 dimensions, number of attention heads for each down and upsampling level [4, 8,8,8], attention window size for each down and upsampling level [4,2,2,2], 512 MLP nodes per Swin Transformer, Shift Attention windows.

5.3.2 Computational System

The experiments of the present work were carried out entirely on ARIS [82], which is the Greek supercomputer developed and operated by GRNET in Athens. ARIS consists of 532 computing nodes divided as follows:

- 426 thin nodes: regular computing nodes without accelerator.
- 44 gpu nodes: “2 x NVIDIA Tesla k40m” accelerated nodes.
- 18 phi nodes: “2 x INTEL Xeon Phi 7120p” accelerated nodes.
- 44 fat nodes: Fat nodes compute have a higher number of cores and more memory per core than thin nodes.
- 1 ml node: “8 x NVIDIA Volta V100” accelerators.

GPU nodes were used in this work. The experiments are performed by running SLURM jobs which have limitations on computing resources, more specifically the maximum

allowed run time is 96 hours and the maximum allowed memory usage is 56000 MB. These limitations influenced the choice of parameters in the previous paragraph.

Chapter 6

Experimental results

This chapter presents the results of the experiments performed with the models analysed in chapter 5 on the datasets presented in chapter 4. The results are presented both in quantitative form, in the tables below, and in qualitative form with examples of mask images generated by the trained models, which are compared to the ground truth masks. In addition, the effect of data augmentation in improving the results is examined, as well as the generalizability of the models.

6.1 Results presentation

Table 6.1 presents the results of the experiments on the CVC-ClinicDB dataset which, as previously analyzed, includes images of polyps derived from colonoscopy videos. The best performing model in terms of Dice Coefficient (DSC) and mean Intersection over Union (mIoU) metrics is DeepLabv3+ followed by MSRF-Net, R2U-net, TransUnet and UNet. The highest Precision is achieved by MSRF-Net and R2U-Net, while the highest Recall is achieved by DeepLabv3+. The good performance of these models is confirmed by Figure 6.1 as the masks generated by them resemble the real masks satisfactorily. Models such as SwinUNet, VNet and ResUNet seem to fail on this dataset. The masks generated by them show significant imperfections compared to the real masks.

Table 6.2 shows the results of the experiments on the Kvasir-SEG dataset which also includes images of gastrointestinal polyps. The most powerful models with respect to the mIoU metric are MSRF-Net and DeepLabv3+ followed by UNet and TransUNet, while with respect to the DSC metric the most powerful model is DeepLabv3+ followed by R2U-Net, TransUNet and UNet. The MSRF-Net and UNet models achieve the highest Precision, and the DeepLabv3+ model achieves the highest Recall. Observing the visual results in Figure 6.1, the robust models produce fairly high quality masks in which the target polyp is detected, however there may be some small misidentified regions. The models that seem to fail are VNet, ResUNet-a and SwinUNet, which is visually confirmed by the weak generated masks.

Table 6.3 lists the results on the 2018 Data Science Bowl dataset which includes images of nuclei of various cell types. An initial observation is that all models achieve significantly higher performance relative to the other datasets. According to the DSC metric the strongest models are DeepLabv3+, R2U-Net, TransUNet, Attention UNet, Swin-

UNet, UNet and VNet while according to mIoU the strongest models are TransUNet, UNet followed by MSRF-Net, Attention UNet, R2U-Net, SwinUNet, VNet and DeepLabv3+. Highest Precision is achieved by TransUNet and highest Recall is achieved by MSRF-Net and R2U-Net. The only model that shows significantly lower performance than the others is ResUNet-a. Figure 6.1 shows the very high similarity of the model masks to the actual mask in terms of the grayscale image, while in terms of the colored image the masks of the models that achieve the highest scores on the mIoU metric are clearly of higher quality.

Finally, Table 6.4 lists the results on the SegPC dataset which consists of plasma cell images from Multiple Myeloma patients. The model performances for this dataset are lower than the above datasets, which makes SegPC the most challenging dataset. The strongest model based on the DSC metric is DeepLabv3+ followed by TransUNet,R2U-Net,Attention UNet and UNet, while based on the mIoU metric the strongest model is R2U-Net followed by MSRF-Net and TransUNet. The highest Precision is achieved by the R2U-Net model and the highest Recall is achieved by the ResUNet model. These results are visually confirmed in Figure 6.1. The models that appear to fail from both the quantitative results in Table 6.4 and the visual results in Figure 6.1 are SwinUNet and ResUNet-a.

Method	**DSC**	**mIoU**	**Precision**	**Recall**
MSRF-Net	0.87	0.84	**0.95**	0.89
UNet	0.92	0.80	0.91	0.87
VNet	0.81	0.64	0.79	0.76
Att-UNet	0.87	0.69	0.92	0.73
ResUNet-a	0.86	0.71	0.85	0.81
SwinUNet	0.79	0.56	0.70	0.71
TransUNet	0.94	0.80	0.92	0.86
R2U-Net	0.94	0.83	**0.95**	0.88
DeepLabv3+	**0.99**	**0.89**	0.94	**0.94**
ResUNet	0.70	0.64	0.80	0.77
ResUNet++	0.78	0.72	0.90	0.78

Table 6.1. *Quantitative model evaluation results for the CVC-ClinicDB dataset with data augmentation*

6.2 Data augmentation effect study

In this section of this chapter, we examine the effect of data augmentation on the quantitative results of the experiments. Specifically, all experiments shown in the tables above were repeated with the original data sets without augmentation, thus containing a smaller number of samples. It is expected that the results without data augmentation are worse than the results with data augmentation. Furthermore, the greater the increase in scores observed with increasing the size of the datasets, the more the power of the models is confirmed, as it is shown that their performance will be increased in the event that more and larger medical image datasets are produced in the future.

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.86	**0.73**	**0.89**	0.80
UNet	0.90	0.71	**0.89**	0.78
VNet	0.76	0.49	0.73	0.60
Att-UNet	0.86	0.62	0.82	0.72
ResUNet-a	0.78	0.49	0.73	0.60
SwinUNet	0.77	0.50	0.79	0.66
TransUNet	0.90	0.70	0.87	0.79
R2U-Net	0.88	0.69	0.85	0.79
DeepLabv3+	**0.96**	**0.73**	0.87	**0.82**
ResUNet	0.60	0.49	0.66	0.66
ResUNet++	0.65	0.55	0.74	0.69

Table 6.2. *Quantitative model evaluation results for the Kvasir-SEG dataset with data augmentation*

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.91	0.86	0.91	**0.95**
UNet	0.95	**0.87**	0.92	0.94
VNet	0.94	0.85	0.91	0.93
Att-UNet	0.95	0.86	0.92	0.93
ResUNet-a	0.91	0.76	0.90	0.83
SwinUNet	0.94	0.85	0.92	0.92
TransUNet	0.95	**0.87**	**0.93**	0.93
R2U-Net	0.95	0.86	0.90	**0.95**
DeepLabv3+	**0.98**	0.85	0.91	0.93
ResUNet	0.88	0.84	0.89	0.94
ResUNet++	0.89	0.83	0.89	0.93

Table 6.3. *Quantitative model evaluation results for the 2018 Data Science Bowl dataset with data augmentation*

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.80	0.68	0.82	0.79
UNet	0.88	0.66	0.81	0.78
VNet	0.85	0.57	0.75	0.71
Att-UNet	0.89	0.66	0.80	0.79
ResUNet-a	0.77	0.41	0.60	0.57
SwinUNet	0.80	0.45	0.63	0.62
TransUNet	0.90	0.68	0.81	0.80
R2U-Net	0.89	**0.69**	**0.85**	0.78
DeepLabv3+	**0.98**	0.64	0.78	0.78
ResUNet	0.77	0.64	0.74	**0.82**
ResUNet++	0.76	0.62	0.76	0.77

Table 6.4. *Quantitative model evaluation results for the SegPC dataset with data augmentation*

Tables 6.5-6.8 present the results without data augmentation. It is observed that data augmentation causes an approximate 4-5% increase in the mIoU metric, 2-3% increase in the DSC metric, 1-2% increase in the Precision metric and 3-4% increase in the Recall metric. Some models benefit more from larger data volume than others, for example DeepLabv3+ achieves a 2% increase in the mIoU metric while MSRF-Net achieves a 4% increase in the Kvasir-SEG dataset. Other models do not seem to achieve growth through data augmentation, for example ResUNet, ResUNet++ on the CVC-ClinicDB dataset, and ResUNet-a which even achieves a significant reduction in mIoU on the 2018 Data Science Bowl dataset. In general, the increasing trend in model performance as a function of increasing data volume is confirmed. Models such as MSRF-Net,R2U-Net,TransUNet seem to scale faster and to a greater extent with increasing volume of data.

Method	**DSC**	**mIoU**	**Precision**	**Recall**
MSRF-Net	0.89	0.83	0.94	0.80
UNet	0.90	0.78	0.91	0.85
VNet	0.79	0.59	0.78	0.71
Att-UNet	0.83	0.67	0.86	0.75
ResUNet-a	0.84	0.70	0.89	0.76
SwinUNet	0.78	0.54	0.70	0.71
TransUNet	0.91	0.76	0.90	0.83
R2U-Net	0.90	0.76	0.92	0.81
DeepLabv3+	**0.99**	**0.87**	**0.96**	**0.91**
ResUNet	0.66	0.64	0.81	0.76
ResUNet++	0.77	0.72	0.94	0.76

Table 6.5. *Quantitative model evaluation results for the CVC-ClinicDB dataset without data augmentation*

Method	**DSC**	**mIoU**	**Precision**	**Recall**
MSRF-Net	0.81	0.69	0.84	0.79
UNet	0.87	0.64	0.83	0.74
VNet	0.75	0.44	0.69	0.55
Att-UNet	0.84	0.58	0.84	0.66
ResUNet-a	0.75	0.46	0.64	0.62
SwinUNet	0.75	0.45	0.65	0.60
TransUNet	0.87	0.66	0.80	0.80
R2U-Net	0.87	0.66	0.85	0.75
DeepLabv3+	**0.96**	**0.75**	**0.89**	**0.83**
ResUNet	0.54	0.45	0.58	0.67
ResUNet++	0.60	0.50	0.70	0.63

Table 6.6. *Quantitative model evaluation results for the Kvasir-SEG dataset without data augmentation*

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.91	**0.86**	0.91	0.94
UNet	0.95	**0.86**	0.91	**0.95**
VNet	0.94	0.85	0.91	0.93
Att-UNet	0.95	**0.86**	0.91	0.94
ResUNet-a	0.93	0.83	0.88	0.94
SwinUNet	0.94	0.85	0.92	0.92
TransUNet	0.95	**0.86**	**0.95**	0.90
R2U-Net	0.95	**0.86**	0.92	0.93
DeepLabv3+	**0.98**	0.84	0.92	0.91
ResUNet	0.89	0.85	0.90	0.93
ResUNet++	0.89	0.84	0.89	0.94

Table 6.7. *Quantitative model evaluation results for the 2018 Data Science Bowl dataset without data augmentation*

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.79	**0.67**	**0.81**	0.78
UNet	0.86	0.63	0.78	0.77
VNet	0.84	0.57	0.75	0.71
Att-UNet	0.88	0.65	0.78	**0.80**
ResUNet-a	0.74	0.39	0.52	0.39
SwinUNet	0.77	0.43	0.56	0.65
TransUNet	0.88	0.63	0.75	**0.80**
R2U-Net	0.85	0.63	0.77	0.78
DeepLabv3+	**0.97**	0.61	0.76	0.76
ResUNet	0.70	0.56	0.69	0.76
ResUNet++	0.72	0.60	0.79	0.72

Table 6.8. *Quantitative model evaluation results for the SegPC dataset without data augmentation*

6.3 Generalizability study

Generalizability, i.e. the ability of models to perform well on new datasets that may have been obtained with different technology than the dataset on which they have been trained, is an important characteristic of a model and it is deemed necessary to evaluate it. In order to evaluate the generalisation ability, models are trained on one of the two datasets including gastrointestinal polyps (CVC-ClinicDB, Kvasir-SEG) and evaluated on the other dataset. Similarly for the two datasets containing cell nuclei (2018 Data Science Bowl, SegPC). The results of the new experiments are shown in Tables 6.9-6.11 . The results of the training experiment on SegPC and evaluation on the 2018 Data Science Bowl are omitted because they are of extremely low quality, making SegPC an unsuitable dataset for training, in terms of generalization.

While the results are generally unsatisfactory and a model trained on one dataset does not perform well when evaluated on a different dataset, with scores on the mIoU metric not exceeding 53% on the CVC-ClinicDB set, 66% on the Kvasir-SEG set and 30% on the

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.41	0.30	0.42	0.52
UNet	0.53	0.23	0.24	0.84
VNet	0.59	0.26	0.28	0.80
Att-UNet	0.56	0.24	0.26	0.80
ResUNet-a	0.46	0.10	0.25	0.10
SwinUNet	0.55	0.24	0.25	0.76
TransUNet	0.57	0.25	0.27	0.82
R2U-Net	0.55	0.24	0.25	0.78
DeepLabv3+	**0.91**	**0.53**	**0.66**	0.72
ResUNet	0.35	0.22	0.23	**0.85**
ResUNet++	0.30	0.23	0.34	0.40

Table 6.9. *Quantitative model evaluation results on the Kvasir-SEG dataset trained on the CVC-ClinicDB dataset*

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.52	0.42	0.53	0.67
UNet	0.75	0.45	0.69	0.56
VNet	0.66	0.30	0.44	0.49
Att-UNet	0.72	0.42	0.78	0.47
ResUNet-a	0.55	0.14	0.19	0.38
SwinUNet	0.63	0.26	0.30	0.63
TransUNet	0.76	0.47	0.70	0.59
R2U-Net	0.74	0.46	0.77	0.53
DeepLabv3+	**0.97**	**0.66**	**0.93**	**0.70**
ResUNet	0.38	0.30	0.50	0.43
ResUNet++	0.51	0.37	0.52	0.57

Table 6.10. *Quantitative model evaluation results on the CVC-ClinicDB dataset trained on the Kvasir-SEG dataset*

Method	DSC	mIoU	Precision	Recall
MSRF-Net	0.42	0.27	0.27	0.96
UNet	0.69	0.29	0.29	0.94
VNet	0.69	0.29	**0.30**	0.93
Att-UNet	0.68	0.28	0.29	0.95
ResUNet-a	0.54	0.11	0.23	0.14
SwinUNet	0.66	0.26	0.27	0.95
TransUNet	0.68	0.28	0.28	0.96
R2U-Net	0.66	0.26	0.27	**0.97**
DeepLabv3+	**0.87**	0.28	0.29	0.94
ResUNet	0.39	0.25	0.25	**0.97**
ResUNet++	0.45	**0.30**	**0.30**	0.96

Table 6.11. *Quantitative model evaluation results on the SegPC dataset trained on the 2018 Data Science Bowl dataset*

2018 Data Science Bowl set, the results can be used as a benchmark of generalizability. The DeepLabv3+ model shows the highest generalizability by achieving the highest scores in DCS,mIoU, while in contrast the ResUNet-a model fails in this area. Models such as MSRF-Net, VNet, TransUNet, R2U-Net, Attention UNet, UNet and ResUNet++ show quite good results in these experiments, therefore it is judged that they can generalize to a satisfactory degree. An additional observation is that models trained on Kvasir-SEG perform better on CVC-ClinicDB while the opposite is not true.

In addition, as part of the generalizability study as well as in an effort to increase the data, the Polyps and Cells datasets were created, of which the former is a merge of CVC-ClinicDB and Kvasir-SEG and the latter is a merge of the 2018 Data Science Bowl and SegPC. In these experiments, the models were trained on the new Polyps, Cells sets and tested and evaluated on their constituent subsets. The results are shown in Tables 6.12-6.13 .

Tested on	**CVC-ClinicDB**				**Kvasir-SEG**			
Method	**DSC**	**mIoU**	**Precision**	**Recall**	**DSC**	**mIoU**	**Precision**	**Recall**
MSRF-Net	0.90	**0.86**	**0.94**	0.91	0.85	0.75	0.88	0.84
UNet	0.90	0.77	**0.94**	0.80	0.89	0.69	0.88	0.76
VNet	0.78	0.54	0.77	0.64	0.83	0.58	0.82	0.66
Att-UNet	0.91	0.74	**0.94**	0.78	0.87	0.64	0.82	0.75
ResUNet-a	0.81	0.61	0.92	0.65	0.79	0.53	0.77	0.63
SwinUNet	0.78	0.48	0.66	0.64	0.75	0.44	0.56	0.68
TransUNet	0.92	0.74	0.88	0.82	0.89	0.69	0.83	0.81
R2U-Net	0.89	0.75	**0.94**	0.79	0.90	0.73	**0.92**	0.78
DeepLabv3+	**0.99**	**0.86**	0.93	**0.92**	**0.96**	**0.76**	0.86	**0.87**
ResUNet	0.55	0.50	0.58	0.77	0.58	0.46	0.54	0.75
ResUNet++	0.76	0.68	0.93	0.71	0.68	0.59	0.83	0.68

Table 6.12. *Quantitative model evaluation results on the CVC-ClinicDB,Kvasir-SEG trained on the Polyps dataset*

In these experiments, the suitability of the Polyps and Cells datasets for training and their effect on improving the results is also examined. According to Tables 6.12 and 6.13 it is shown that training most models on the Polyps set leads to significantly increased scores when evaluating on the CVC-ClinicDB set and less increased scores when evaluating on the Kvasir-SEG set. In contrast, training the models on the Cells set leads to reduced scores when evaluating on the 2018 Data Science Bowl and SegPC sets. Models that respond positively to training on the Polyps and Cells sets compared to training on the individual sets that comprise them have a higher generalization ability. As demonstrated in the above experiments, models such as DeepLabv3+, MSRF-Net, VNet, Attention UNet, TransUNet, UNet, and R2U-Net generalize better than the other models.

Tested on	**2018 Data Science Bowl**				**SegPC**			
Method	**DSC**	**mIoU**	**Precision**	**Recall**	**DSC**		**Precision**	**Recall**
MSRF-Net	0.91	**0.87**	0.93	**0.93**	0.81	**0.70**	**0.84**	0.81
UNet	0.95	**0.87**	0.93	**0.93**	0.87	0.63	0.75	0.79
VNet	0.92	0.80	**0.94**	0.84	0.76	0.39	0.45	0.74
Att-UNet	0.94	0.86	**0.94**	0.91	0.87	0.63	0.70	0.86
ResUNet-a	0.94	0.84	0.91	0.92	0.75	0.37	0.43	0.73
SwinUNet	0.93	0.82	0.90	0.91	0.77	0.43	0.54	0.69
TransUNet	0.95	0.87	**0.94**	**0.93**	0.87	0.64	0.70	**0.88**
R2U-Net	0.94	0.87	0.92	**0.93**	0.87	0.64	0.74	0.83
DeepLabv3+	**0.98**	0.85	0.92	0.91	**0.97**	0.61	0.75	0.77
ResUNet	0.85	0.72	0.90	0.78	0.57	0.43	0.49	0.79
ResUNet++	0.86	0.74	0.82	0.89	0.69	0.54	0.71	0.69

Table 6.13. *Quantitative model evaluation results on the 2018 Data Science Bowl,SegPC trained on the Cells dataset*

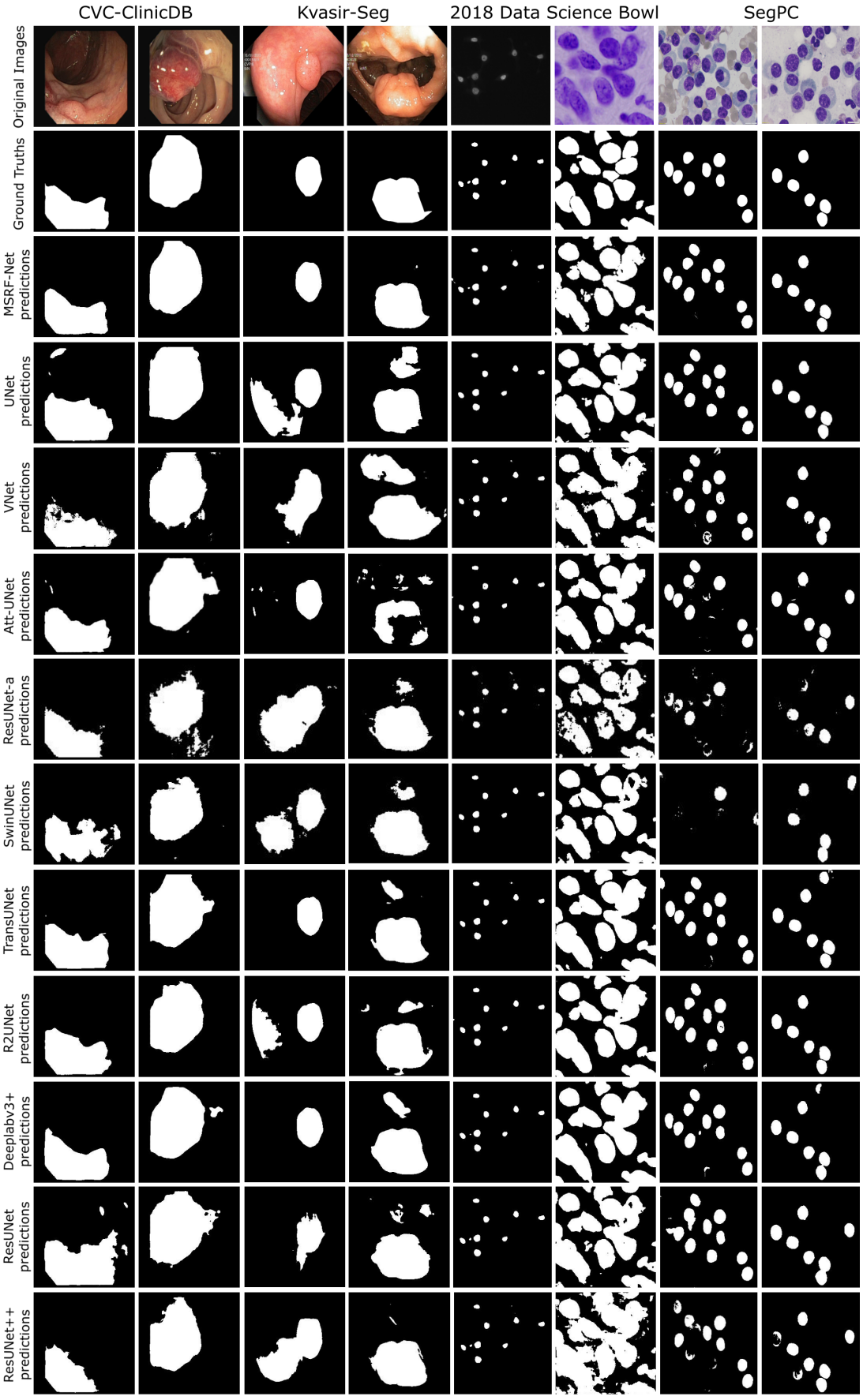

Image 6.1: *Examples of images generated by the models, compared to the actual images*

Part I

Epilogue

Chapter 7

Conclusions and Future Extensions

7.1 Conclusions

The models tested in the experiments of the previous chapter generally follow the U-architecture with variations such as the use of transformers, the existence of an attention mechanism, the utilization of residual learning and the exchange of features at different scales. While the original model that inspired the creation of almost all the others, UNet, seems to consistently perform quite well on all datasets, some of the variations lead to improved results while others are not as successful.

Models such as ResUNet and ResUNet-a, designed for domains other than medical image segmentation, do not perform well enough on datasets involving polyps but achieve satisfactory scores on datasets with cell nuclei, similarly the 2D version of VNet, indicating that this task is quite specialized and requires specialized models to achieve state-of-the-art results. The utilization of residual connections in combination with recurrent networks is found to be very effective in the R2U-Net while their combination with Squeeze and excitation blocks, Atrous Spatial Pyramid Pooling and attention does not lead to any improvement over UNet and at the same time complicates the architecture a lot. The utilization of the attention mechanism does not perform well on the polyp datasets however it leads to better identification of cells with disease in the SegPC set and their separation from healthy cells, as is the case with Attention UNet. The appearance of transformers in U-type architectures, while not promising as an idea at first because by their nature transformers lack localization capability due to the lack of low-level details, in combination with convolutional networks, as is the case in TransUNet, are very efficient encoders bringing a small improvement in UNet scores. On the contrary, without the use of convolutional networks together with transformers, the final scores are not satisfactory, as is the case with SwinUNet. Finally, models that focus more on combining information from different scales and therefore manage to reduce the semantic gap between the low-level encoder features and the high-level decoder features, such as DeepLabv3+ and MSRF-Net, seem to achieve significant improvement in scores on datasets with polyps that are considered more challenging, as well as on SegPC. It is important to note that the MSRF-Net model is considerably more demanding to train in terms of computational resources and runtime than the other models. R2U-Net,TransUNet achieve close scores to MSRF-Net with much less requirements. Moreover, the pre-trained on the large-sized

ImageNet [83] ResNet50 backbone present in DeepLabv3+ makes it more powerful and contributes to fast training.

In realistic scenarios where these models would perform polyp segmentation on images from different laboratories, with different technologies, from different patients, the ability to generalize is considered an essential feature. The models must be able to perform well regardless of parameters such as the equipment with which the images were captured or the patient from whom the images were obtained. It was shown from the study in the previous chapter that the models with the highest scores also have a high generalisation ability.

As for the datasets, those related to polyps are challenging and carry some peculiarities, such as some black frames that show high contrast in the images and confuse the models as to the salient areas of the image, and low general contrast in the images especially in the polyp spots which makes the task of detection more difficult. In the datasets associated with cell nuclei, the task of identifying nuclei is considered fairly easy because there is considerable contrast between them and the plain monochrome background, with difficulties being encountered in identifying specific cells affected by Multiple Myeloma (in SegPC) and in images where nuclei are depicted more faintly or there is considerable overlap between them. These observations are confirmed by the scores presented in the previous chapter.

7.2 Future Expansions

While several models with different architectures were examined in this paper and therefore led to a global study of the task of medical image segmentation, there are additional models that could be tested, such as DCSAU-Net [84], MCGU-Net [85] but also UNet++ [86], U^2-Net [87], UNet3+ [88], which add new details and changes that can significantly favor the results of the experiments. In particular, DCSAU-Net is a deeper and more compact network with U-shape and split-attention architecture that extracts important features using multi-scale split-attention and deeper convolutions, and aims to achieve high performance on more demanding images. MCGU-Net is an extension of UNet with additional Squeeze and Excitation (SE) blocks, Bi-Directional Convolutional LSTM and the dense convolution mechanism. UNet++ is a deeply supervised encoder-decoder network where the encoder and decoder subnets are connected via nested dense skip pathways that reduce the semantic gap between the respective feature maps. Unet3+ takes advantage of full-scale skip connections and deep supervision, achieving lower computational complexity. Full-scale skip connections combine low-level details with semantics from higher levels at different scales while deep supervision helps learning hierarchical representations from full-scale aggregated feature maps. The U^2Net was designed for salient object detection and its architecture consists of two layers of the UNet architecture. It succeeds in capturing more contextual information at different scales and increases the depth of the architecture without adding additional computational cost.

Moreover, in Figure 6.1 it is easily observed that in some examples in the CVC-ClinicDB and Kvasir-SEG datasets, the more efficient models produce masks that misiden-

tify as polyps, regions that are not polyps. This leads to lower scores in the evaluation metrics and less qualitative visual results. These deficiencies can be reduced by combining the above models with a sufficiently powerful Object Detection model, such as Detectron2 [89], which produces bounding boxes for each detected object in its output. If the masks from the models discussed in chapter 5 are combined with the new bounding boxes, the erroneously activated (detected as polyps) areas can be reduced or even disappear. This combination can be done later, i.e. after the masks and bounding boxes have been produced, or even during model training provided that the model architecture is modified to support the learning of 2 types of labels (binary masks, bounding boxes), a process that belongs to the field of Multimodal Learning.

Finally, a larger and perhaps more intense data augmentation could certainly bring about an improvement in the results, since an increasing trend in scores as a function of data volume has already been observed.

Appendices

Appendices

A

Theoretical Background - Special Concepts

A.1 Convolutions

A.1.1 Simple Convolutions

Convolutions in images act like filters. They are a kind of matrix operation, in which a small table of weights called a kernel slides through the entire image by performing elementwise multiplication with the part of the image it covers at any given time, and summing the results to form the output. Through the process of convolution, the dimension of features in a neural network is reduced [31].

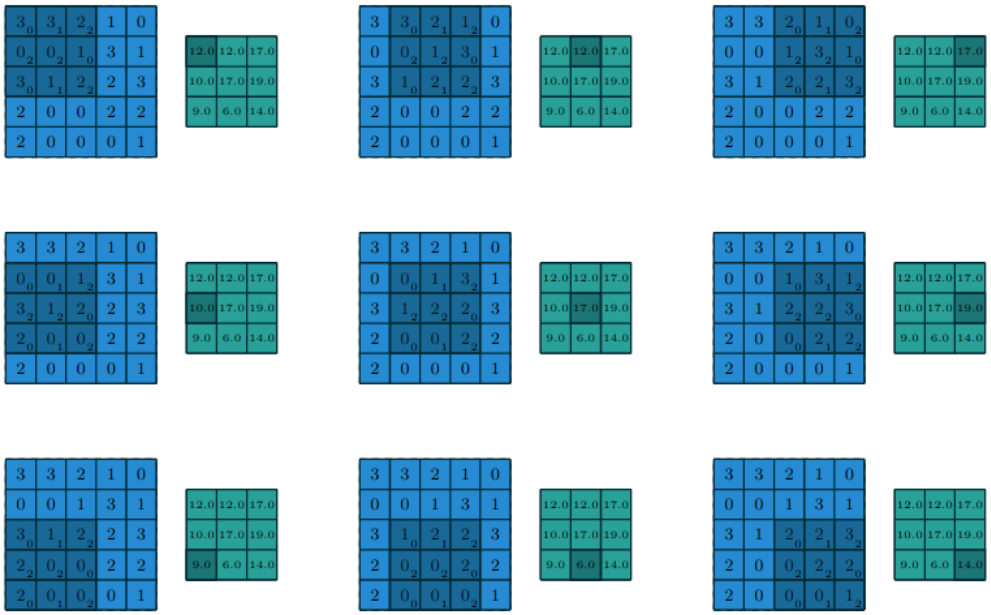

Image A.1: *Example of a simple convolution in an image [31]*

A.1.2 Atrous Convolutions

Dilated/atrous convolutions introduce a parameter to simple convolutions, the dilation factor. This determines the gap between the values of a kernel, as shown in Figure A.2. A 3x3 kernel with a dilation factor of 2 has the same receptive-field as a 5x5 kernel

using only 9 parameters. This increases the receptive-field of the filters with the same computational cost. The dilated convolutions help to maintain high spatial resolution of feature matrices along a convolutional network which matrices go through multiple layers of convolution and clustering leading to significant downsampling. This is why they are widely used in the task of real-time image segmentation [19].

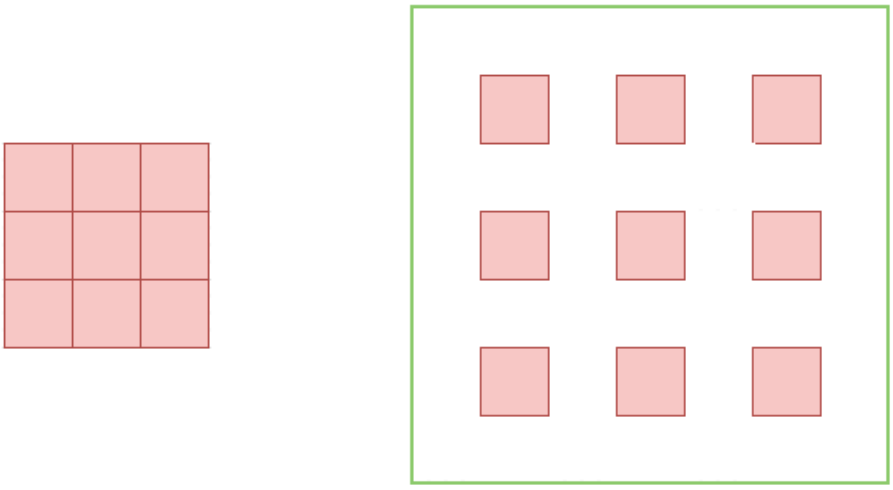

Image A.2: *Example of an expanded kernel [19]*

A.1.3 Depthwise Convolutions

In depthwise convolution on a three-channel RGB image, the depth of the image does not change and so the output image has 3 channels. This is done by using three kernels of depth 1 instead of using one kernel of depth 3. Each kernel performs a convolution on one channel of the image and the results of the 3 convolutions are stacked to form the output image which has 3 channels.

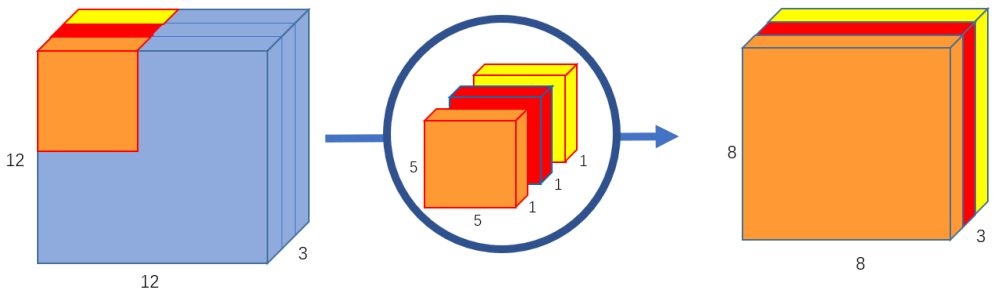

Figure A.1. *Example of a depthwise convolution [18]*

A.1.4 Pointwise Convolutions

Pointwise convolution is called this way because it uses a 1x1 kernel that passes through each point in the image. This kernel has as many channels as the image, so 3 for an RGB image; therefore the output of such a convolution results in a single-channel image with the same length and width as the original input image [18].

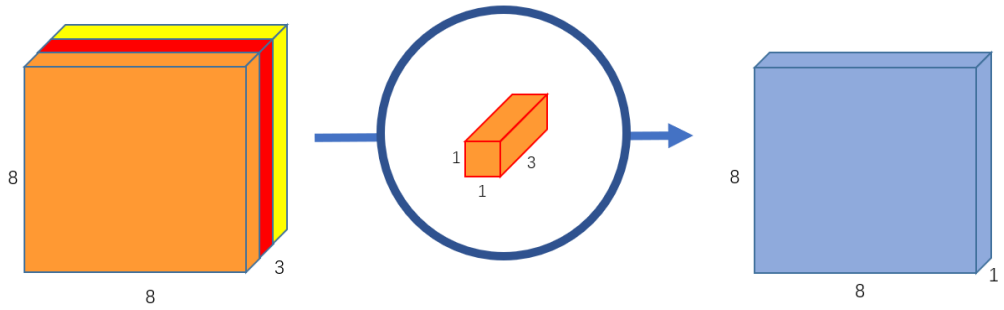

Figure A.2. *Example of a pointwise convolution [18]*

A.1.5 Depthwise Separable Convolutions

Depthwise separable convolutions combine the above two types of convolution, as shown in Figure A.3. It can increase the channels of the output image with less computational complexity than a simple convolution. First a depthwise convolution is performed as in Figure A.1, and then pointwise convolutions are performed with an appropriate number of kernels, depending on the desired number of output channels. In Figure A.4 for example, a convolution with 256 cores 1x1 is performed resulting in an output image with 256 channels. The major advantage offered by depthwise separable convolutions is the low computational complexity. If the same result were to be produced by simple convolution, a lot more multiplications and conversions of the input image would be required. In contrast, with depthwise separable convolution the number of multiplications is significantly smaller and the process is much simpler, so it is a more efficient approach [18].

A.2 Squeeze and Excitation

The Squeeze and Excitation block is an architectural module designed to improve the representational capability of networks by allowing it to perform dynamic recalculation of per-channel features. This block includes a convolutional block as input. Then each channel is converted to a unique numerical value through average pooling (squeeze process). This is followed by a dense layer with a ReLU activation and the output channel complexity is reduced. Another dense layer with a Sigmoid activation imparts to each channel a smooth gating function. Finally, based on these results weights are assigned to each feature matrix of the convolutional block (excitation process) [44]. This architecture

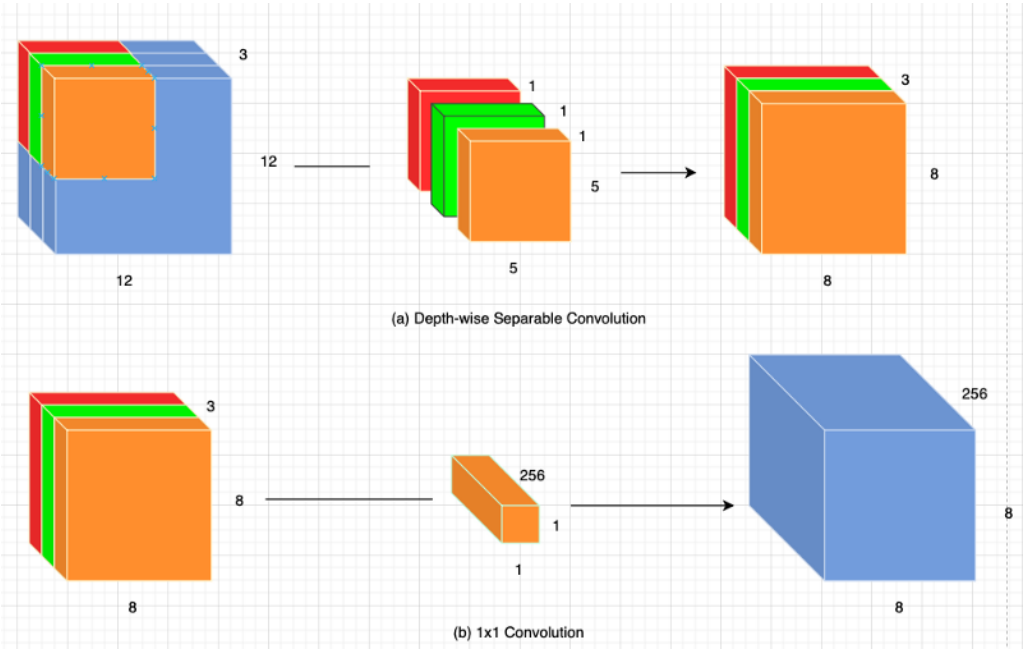

Figure A.3. *Example of a depthwise separable convolution [19]*

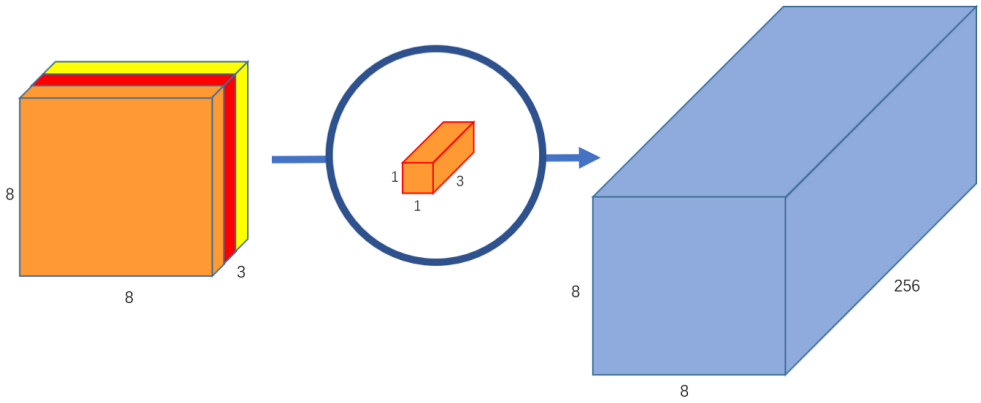

Figure A.4. *Example of a pointwise convolution with 256 kernels 1x1 [19]*

is shown in Figure A'.5.

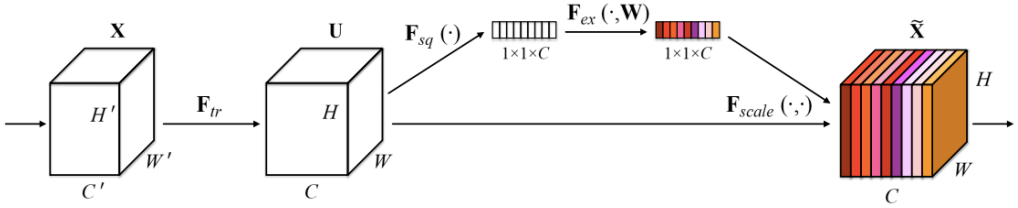

Figure A.5. *Squeeze and Excitation architecture*

A.3 Atrous Spatial Pyramid Pooling

Atrous Spatial Pyramid Pooling serves to extract information for multiple scales. In the feature map extracted from the backbone, 4 parallel atrous convolutions with different dilation rates (namely 1,6,12,18) are performed to manage the segmentation of objects at different scales. It is illustrated in Figure A'.6 and was first proposed in the DeepLabv2 model [43]. The results from the 4 dilated convolutions are clustered, merged and passed through a final 1x1 convolution before the final results.

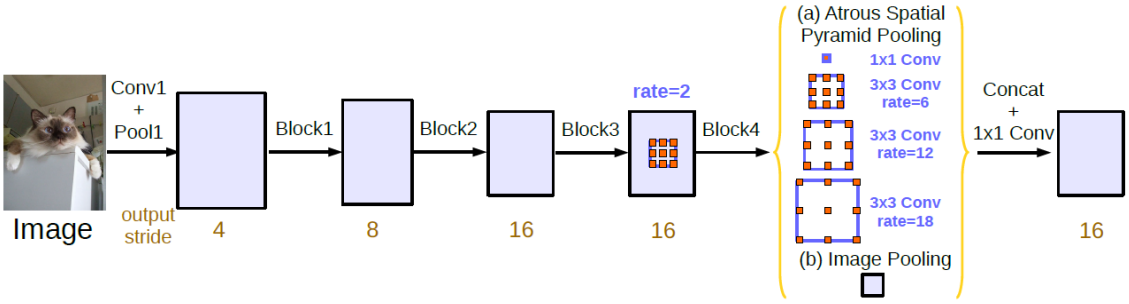

Figure A.6. *The Atrous Spatial Pyramid Pooling mechanism*

A.4 Activation Functions

Activation functions modify the output of the neurons by bringing them into an appropriate format according to the information flow through the rest of the network. Below are the activation functions shown in this paper.

A.4.1 ReLU

ReLU (Rectified Linear Unit is the most popular activation function [20]. It is a half rectified function, i.e. negative values are set to zero while positive values remain unchanged. It is monotonic and has a monotonic derivative. The range of values is $[0, \infty]$. It is described by the formula $y(x) = max(0, x)$ and is shown in Figure A'7.

A.4.2 LeakyReLU

The difference between LeakyReLU and ReLU is that negative values are not zeroed but multiplied by a factor a which has a value of 0.01 [20]. It is shown in figure A'.8 and its formula is

$$y(x) = 0.01x, x < 0$$

$$y(x) = x, x \geq 0$$

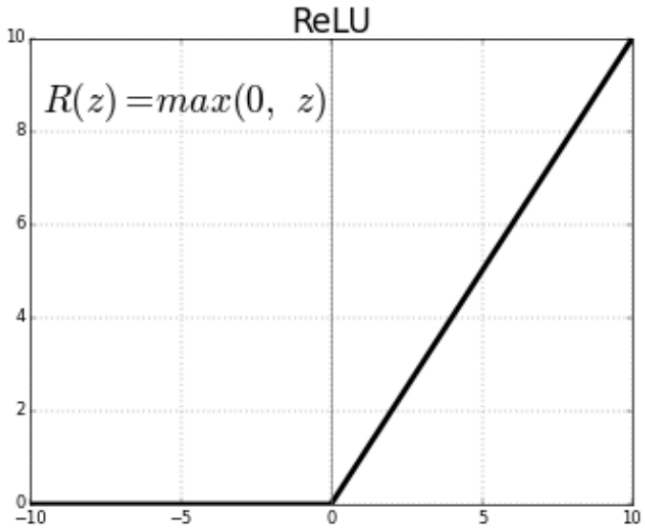

Figure A.7. *The ReLU activation function*

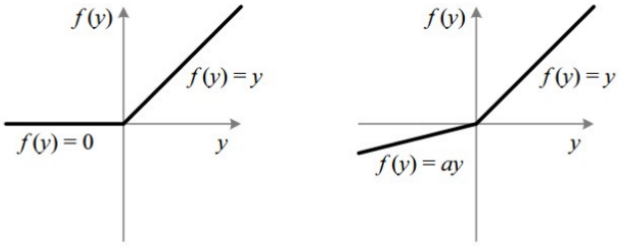

Figure A.8. *The ReLU activation function (left) and LeakyReLU (right) [20]*

A.4.3 PReLU

PReLU (Parametric ReLU) is a type of LeakyReLU where the parameter a has no fixed value but is calculated by the network [90]. The formula describing it is as follows:

$$y(x) = ax, x < 0$$

$$y(x) = x, x \geq 0$$

A.4.4 GELU

The GELU (Gaussian Error Linear Unit) function combines the fast convergence offered by the ReLU, PReLU functions, with the Dropout procedure that randomly zeroes out some activations, and with the Zoneout procedure that stochastically multiplies the input with 1. It therefore stochastically multiplies the input by 0 or 1 and obtains the value of the output deterministically [21]. It is shown in Figure A.9 and its formula is as follows:

$$y(x) = 0.5x(1 + tanh(\sqrt{2/\pi}(x + 0.044715x^3)))$$

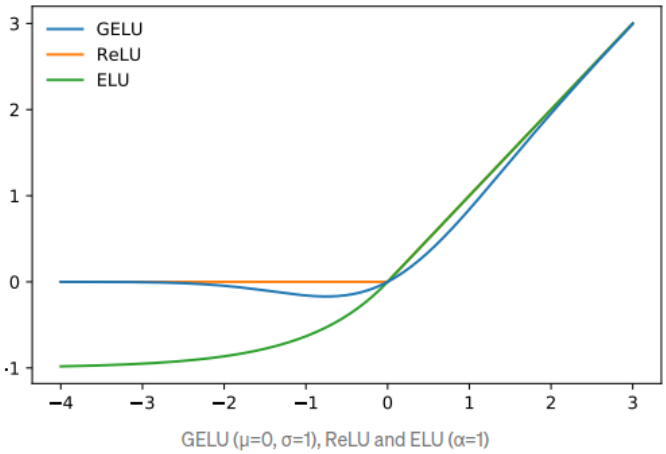

Figure A.9. *The GELU activation function (blue line) [21]*

A.4.5 Sigmoid

The Sigmoid activation function places values in the interval [0,1] and is widely used in models that predict probabilities as output. It is differentiable and monotone and its derivative is non-monotone. It is shown in figure A'10 and it is defined by the formula [20]:

$$y(x) = \frac{1}{1 + e^{-x}}$$

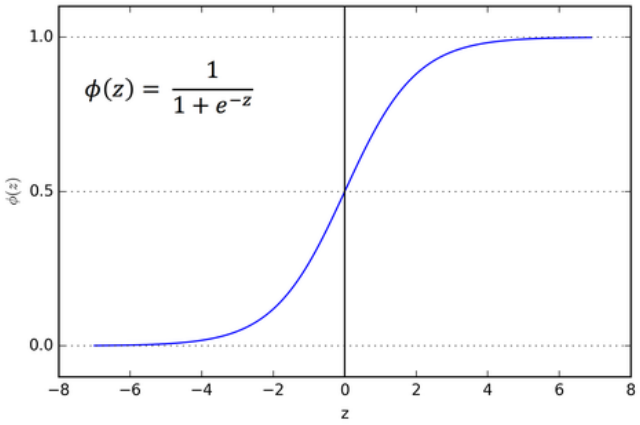

Figure A.10. *The Sigmoid activation function [20]*

Bibliography

[1] Sumit Saha. *A comprehensive guide to Convolutional Neural Networks-the eli5 way*, 2018.

[2] Jonathan Long, Evan Shelhamer και Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*, 2014.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser και Illia Polosukhin. *Attention Is All You Need*, 2017.

[4] Kriz Moses. *Encoder-decoder Seq2Seq models, clearly explained*, 2021.

[5] Andrej Karpathy και Li Fei-Fei. *Deep Visual-Semantic Alignments for Generating Image Descriptions*, 2014.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep Residual Learning for Image Recognition*, 2015.

[7] Olaf Ronneberger, Philipp Fischer και Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015.

[8] Fausto Milletari, Nassir Navab και Seyed Ahmad Ahmadi. *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation*, 2016.

[9] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha και Vijayan K. Asari. *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation*, 2018.

[10] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker και Daniel Rueckert. *Attention U-Net: Learning Where to Look for the Pancreas*, 2018.

[11] Zhengxin Zhang, Qingjie Liu και Yunhong Wang. *Road Extraction by Deep Residual U-Net*. 2017.

[12] Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Dag Johansen, Thomasde Lange, Pal Halvorsen και Havard D. Johansen. *ResUNet++: An Advanced Architecture for Medical Image Segmentation*, 2019.

[13] Foivos I. Diakogiannis, François Waldner, Peter Caccetta και Chen Wu. *ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data*. 2019.

[14] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille και Yuyin Zhou. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*, 2021.

[15] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian και Manning Wang. *Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation*, 2021.

[16] Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff και Hartwig Adam. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*, 2018.

[17] Abhishek Srivastava, Debesh Jha, Sukalpa Chanda, Umapada Pal, Håvard D. Johansen, Dag Johansen, Michael A. Riegler, Sharib Ali και Pål Halvorsen. *MSRF-Net: A Multi-Scale Residual Fusion Network for Biomedical Image Segmentation*. 2021.

[18] Chi Feng Wang. *A basic introduction to separable convolutions*, 2018.

[19] Aadhithya Sankar. *A primer on atrous convolutions and depth-wise separable convolutions*, 2021.

[20] Sagar Sharma. *Activation functions in neural networks*, 2021.

[21] Shaurya Goel. *Gelu (gaussian error linear unit)*, 2019.

[22] Ayush Pant. *Introduction to machine learning for beginners*, 2019.

[23] Alexey A. Novikov, Dimitrios Lenis, David Major, Jiri Hladůvka, Maria Wimmer και Katja Bühler. *Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs*, 2017.

[24] Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez και Fernando Vilariño. *WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians*. *Computerized Medical Imaging and Graphics*, 43:99–111, 2015.

[25] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pl Halvorsen, Thomasde Lange, Dag Johansen και Hvard D Johansen. *Kvasir-seg: A segmented polyp dataset*. *International Conference on Multimedia Modeling*, pages 451–462. Springer, 2020.

[26] Juan C. Caicedo, Allen Goodman, Kyle W. Karhohs, Beth A. Cimini, Jeanelle Ackerman, Marzieh Haghighi, CherKeng Heng, Tim Becker, Minh Doan, Claire McQuin, Mohammad Rohban, Shantanu Singh και Anne E. Carpenter. *Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl*. *Nature Methods*, 16(12):1247–1253, 2019.

[27] Anubha Gupta, Pramit Mallick, Ojaswa Sharma, Ritu Gupta και Rahul Duggal. *PCSeg: Color model driven probabilistic multiphase level set based tool for plasma cell segmentation in multiple myeloma*. *PLOS ONE*, 13(12):e0207908, 2018.

[28] Anubha Gupta, Rahul Duggal, Shiv Gehlot, Ritu Gupta, Anvit Mangal, Lalit Kumar, Nisarg Thakkar και Devprakash Satpathy. *GCTI-SN: Geometry-inspired chemical and tissue invariant stain normalization of microscopic medical images*. *Medical Image Analysis*, 65:101788, 2020.

[29] Shiv Gehlot, Anubha Gupta και Ritu Gupta. *EDNFC-Net: Convolutional Neural Network with Nested Feature Concatenation for Nuclei-Instance Segmentation*. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020.

[30] Anubha Gupta, Ritu Gupta, Shiv Gehlot και Shubham Gehlot. *SegPC-2021: Segmentation of Multiple Myeloma Plasma Cells in Microscopic Images*, 2021.

[31] Vincent Dumoulin και Francesco Visin. *A guide to convolution arithmetic for deep learning*, 2016.

[32] Athanasios Tagaris, Dimitrios Kollias και Andreas Stafylopatis. *Assessment of Parkinson's disease based on deep neural networks*. *International Conference on Engineering Applications of Neural Networks*, pages 391–403. Springer, 2017.

[33] Ilianna Kollia, Andreas Georgios Stafylopatis και Stefanos Kollias. *Predicting Parkinson's disease using latent information extracted from deep neural networks*. *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[34] James Wingate, Ilianna Kollia, Luc Bidaut και Stefanos Kollias. *Unified deep learning approach for prediction of Parkinson39;s disease*. *IET Image Processing*, 14(10):1980–1989, 2020.

[35] Dimitrios Kollias, Anastasios Arsenos, Levon Soukissian και Stefanos Kollias. *Mia-cov19d: Covid-19 detection through 3-d chest ct image analysis*. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 537–544, 2021.

[36] Dimitrios Kollias, Anastasios Arsenos και Stefanos Kollias. *Ai-mia: Covid-19 detection & severity analysis through medical imaging*. *arXiv preprint arXiv:2206.04732*, 2022.

[37] Dimitris Kollias, Y Vlaxos, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate και S Kollias. *Transparent adaptation in deep medical image diagnosis*. *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, pages 251–267. Springer, 2020.

[38] Athanasios Tagaris, Dimitrios Kollias, Andreas Stafylopatis, Georgios Tagaris και Stefanos Kollias. *Machine learning for neurodegenerative disorder diagnosis—survey of practices and launch of benchmark dataset*. *International Journal on Artificial Intelligence Tools*, 27(03):1850011, 2018.

[39] Dimitrios Kollias, Athanasios Tagaris, Andreas Stafylopatis, Stefanos Kollias και Georgios Tagaris. *Deep neural architectures for prediction in healthcare*. *Complex & Intelligent Systems*, 4(2):119–131, 2018.

[40] Dimitrios Kollias, Miao Yu, Athanasios Tagaris, Georgios Leontidis, Andreas Stafylopatis και Stefanos Kollias. *Adaptation and contextualization of deep neural network models*. *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–8. IEEE.

[41] Francesco Caliva, Fabio Sousa De Ribeiro, Antonios Mylonakis, Christophe Demazi'ere, Paolo Vinai, Georgios Leontidis και Stefanos Kollias. *A deep learning approach to anomaly detection in nuclear reactors*. *2018 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2018.

[42] Nabil Ibtehaz και M. Sohel Rahman. *MultiResUNet : Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation*. 2019.

[43] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy και Alan L. Yuille. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*, 2016.

[44] Jie Hu, Li Shen, Samuel Albanie, Gang Sun και Enhua Wu. *Squeeze-and-Excitation Networks*, 2017.

[45] Towaki Takikawa, David Acuna, Varun Jampani και Sanja Fidler. *Gated-SCNN: Gated Shape CNNs for Semantic Segmentation*. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019.

[46] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu και Bin Xiao. *Deep High-Resolution Representation Learning for Visual Recognition*, 2019.

[47] Fabio De Sousa Ribeiro, Francesco Caliv&39;a, Mark Swainson, Kjartan Gudmundsson, Georgios Leontidis και Stefanos Kollias. *Deep bayesian self-training*. *Neural Computing and Applications*, 32(9):4275–4291, 2020.

[48] Adrian Rosebrock. *Introduction to neural networks*, 2021.

[49] Phivos Mylonas, Evaggelos Spyrou, Yannis Avrithis και Stefanos Kollias. *Using visual context and region semantics for high-level concept detection*. *IEEE Transactions on Multimedia*, 11(2):229–243, 2009.

[50] Pranoy Radhakrishnan. *Why transformers are slowly replacing cnns in Computer Vision?*, 2021.

[51] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan και Mubarak Shah. *Transformers in Vision: A Survey*. *ACM Computing Surveys*, 2022.

[52] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli και Levent Sagun. *ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases*, 2021.

[53] Hack A BIT. *Semantic segmentation*, 2019.

[54] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel και Yoshua Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*, 2015.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser και Illia Polosukhin. *Attention Is All You Need*, 2017.

[56] Lilian Weng. *Attention? Attention!* *lilianweng.github.io*, 2018.

[57] Wanshun Wong. *What is residual connection?*, 2021.

[58] Bashar Alhnaity, Stefanos Kollias, Georgios Leontidis, Shouyong Jiang, Bert Schamp και Simon Pearson. *An autoencoder wavelet based deep neural network with attention mechanism for multi-step prediction of plant growth*. *Information Sciences*, 560:35–50, 2021.

[59] Andreas Psaroudakis και Dimitrios Kollias. *MixAugment & Mixup: Augmentation Methods for Facial Expression Recognition*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2375, 2022.

[60] Debesh Jha, Pia H. Smedsrud, Dag Johansen, Thomasde Lange, Havard D. Johansen, Pal Halvorsen και Michael A. Riegler. *A Comprehensive Study on Colorectal Polyp Segmentation With ResUNet, Conditional Random Field and Test-Time Augmentation*. *IEEE Journal of Biomedical and Health Informatics*, 25(6):2029–2040, 2021.

[61] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomasde Lange, Dag Johansen, Concetto Spampinato, Duc Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler και Pl Halvorsen. *KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection*. *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, pages 164–169, New York, NY, USA, 2017. ACM.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep Residual Learning for Image Recognition*, 2015.

[63] Ming Liang και Xiaolin Hu. *Recurrent Convolutional Neural Network for Object Recognition*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[64] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang και Jiaya Jia. *Pyramid Scene Parsing Network*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[65] D Kollias, N Bouas, Y Vlaxos, V Brillakis, M Seferis, I Kollia, L Sukissian, J Wingate και S Kollias. *Deep Transparent Prediction through Latent Representation Analysis*. *arXiv preprint arXiv:2009.07044*, 2020.

[66] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin και Baining Guo. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*, 2021.

[67] Kronland Martinet R. Morlet J. Tchamitchian P. Holschneider, M. *A real-timealgorithm for signal analysis with the help of the wavelet transform*. pages 289–297, 1989.

[68] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus και Yann LeCun. *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*, 2013.

[69] Alessandro Giusti, Dan C. Cireşan, Jonathan Masci, Luca M. Gambardella και Jürgen Schmidhuber. *Fast Image Scanning with Deep Max-Pooling Convolutional Neural Networks*. 2013.

[70] George Papandreou, Iasonas Kokkinos και Pierre Andre Savalle. *Modeling local and global deformations in Deep Learning: Epitomic convolution, Multiple Instance Learning, and sliding window detection*. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[71] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*, 2016.

[72] Laurent SIfre και Stéphane Mallat. *Rigid-Motion Scattering for Texture Classification*, 2014.

[73] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto και Hartwig Adam. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, 2017.

[74] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li και Li Fei-Fei. *ImageNet: A large-scale hierarchical image database*. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[75] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah και Kyoung Mu Lee. *Enhanced Deep Residual Networks for Single Image Super-Resolution*, 2017.

[76] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke και Alex Alemi. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, 2016.

[77] Towaki Takikawa, David Acuna, Varun Jampani και Sanja Fidler. *Gated-SCNN: Gated Shape CNNs for Semantic Segmentation*, 2019.

[78] Daniel Godoy. *Understanding binary cross-entropy / log loss: A visual explanation*, 2019.

[79] Ekin Tiu. *Metrics to evaluate your semantic segmentation model*, 2020.

[80] Adam Shafi. *How to learn the definitions of precision and recall*, 2022.

[81] Diederik P. Kingma και Jimmy Ba. *Adam: A Method for Stochastic Optimization*, 2014.

[82] GRNET S.A. *ARIS Documentation - Hardware overview*.

[83] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li και Li Fei-Fei. *Imagenet: A large-scale hierarchical image database*. *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[84] Qing Xu, Wenting Duan και Na He. *DCSAU-Net: A Deeper and More Compact Split-Attention U-Net for Medical Image Segmentation*, 2022.

[85] Maryam Asadi-Aghbolaghi, Reza Azad, Mahmood Fathy και Sergio Escalera. *Multi-level Context Gating of Embedded Collective Knowledge for Medical Image Segmentation*, 2020.

[86] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh και Jianming Liang. *UNet: A Nested U-Net Architecture for Medical Image Segmentation*. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. Springer International Publishing, 2018.

[87] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane και Martin Jagersand. *U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection*. 2020.

[88] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen Wei Chen και Jian Wu. *UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation*, 2020.

[89] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan Yen Lo και Ross Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2, 2019.

[90] Danqing Liu. *A practical guide to relu*, 2017.