POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Few-shot Learning in Vision Transformers for Skin Cancer Semantic Segmentation

Supervisors

Prof.ssa Tatiana TOMMASI

Prof. Marco ORTOLANI

Candidate

Francesco DI GANGI

October 2024

Abstract

Skin cancer is one of the most prevalent and potentially life-threatening diseases, characterized by aberrant skin cell proliferation, mostly caused by DNA damage due to exposure to ultraviolet (UV) radiation from the sun or other sources, like tanning beds. There are several types of skin cancer, such as melanoma, squamous cell carcinoma, and basal cell carcinoma, each with unique traits and implications for treatment and diagnosis.

Timely detection of skin cancer is fundamental to maximize the probability of successful treatment. In this regard, computer-assisted diagnosis plays a crucial role and is supported by the automated analysis of images through segmentation. Segmentation effectively recognizes and outlines regions of interest in dermoscopic images, aiding healthcare practitioners in precisely identifying and assessing lesions. This enables early diagnosis and the tracking of skin changes over time, ultimately facilitating prompt medical intervention.

In the context of skin cancer, the ultimate goal of segmentation is to offer dermatologists and other medical professionals accurate and sophisticated tools to help them quickly identify and treat concerning skin lesions.

The use of deep learning for image segmentation has been widely documented in the recent literature. In particular, Convolutional Neural Networks (CNNs) have been shown to be able to accurately separate skin lesions by autonomously learning from a training dataset and automatically recognizing significant characteristics in dermoscopic pictures. Although convolutional operators have been an essential component of image processing, Vision Transformers have become more popular due to their lack of ability to handle spatial variations and capture global context. ViTs are motivated by the success of Transformers in natural language processing and use a global attention mechanism that enables them to analyze images quickly by taking into account the relationships between all of the image's components, regardless of how far apart they are. As a result, ViTs are able to outperform CNNs in a variety of visual tasks and get around some of their shortcomings.

The main limitation consists in the lack of the large amount of labelled data needed to train algorithms like Vision Transformers.

This thesis focuses on exploring the use of Vision Transformer for image segmentation within a *Few-shot learning* paradigm, where a (potentially pre-trained) model can be further refined to make accurate predictions or perform specific tasks using only a very small amount of labeled training data, which is particularly appropriate in the described scenario.

The experimental assessment will rely on publicly available datasets, such as the ISIC dataset, which contains dermoscopic images for research and development in

dermatology, particularly for skin cancer detection and diagnosis.
Due to the lack of sufficient medical data available in real-world scenarios, Cross Domain Segmentation methodologies combined with Few-shot Learning techniques are then implemented and studied.
Experimental results will be presented and compared with most recent works, revealing that segmentation could potentially prove beneficial in both current and forthcoming medical endeavors.

Acknowledgements

I would like to thank Prof. Marco Ortolani and Keele University, who allowed me to do this thesis and who welcomed me as if I were one of their students, allowing me to interface in a completely new world that made me passionate about what I did.

I would also like to thank Prof. Tatiana Tommasi, who had faith in me and agreed to be my internal supervisor even before I was her student, and the Politecnico di Torino, which allowed me to have an adequate education for what I did and what I am going to do in the future.

Finally, I want to thank some people, who I know have believed in me since the bachelor's degree, the *Kyssene* group, a group that I found and embraced me as if we were brothers and who I hope will continue to be so. I also want to thank those who were, are and always will be there: Gabriele, Fabio, Simone, Vincenzo. I also want to thank Furio: a person who was there, and who unfortunately is no longer with us. I want to thank Dr. Guccione, who helped me for years, and finally I want to thank the most important people in my life: my parents and my sister. It is thanks to you that I have become what I am today.

"A mia madre, che mi ha insegnato cos'è l'amore e cosa significa essere amati; a mio padre, che mi ha insegnato cosa significa sacrificarsi per il bene altrui; a mia sorella, il mio faro in mezzo a una tempesta."

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The skin, which is the largest organ in the human body, is a sophisticated structure consisting of multiple layers that work together to regulate body temperature, enable touch sensations, and provide protection. It is essential to have a thorough understanding of the intricate morphology and functions of the skin in order to comprehend different dermatological conditions and enhance diagnostic techniques. This thesis specifically focus on examine the fundamental aspects of skin structure, with a special emphasis on the interactions between its primary layers—the dermis, hypodermis, and epidermis. In order to understand the underlying causes of skin lesions and cancers, this thesis will closely examine the cellular makeup and roles of these layers. Skin cancer is the most prevalent form of cancer worldwide,

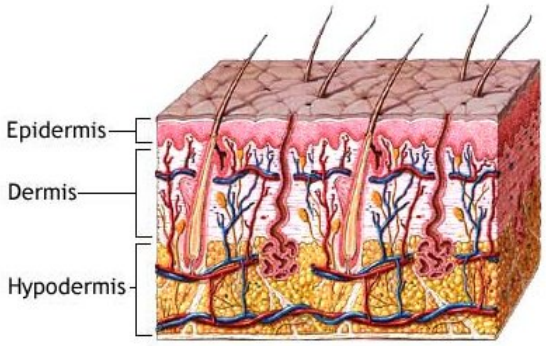


Figure 1.1: Layer of the skin.

and cases of melanoma, its most dangerous type, have increased by nearly 50% in the last decade [1]. Skin cancer occurs when cells in the epidermis undergo abnormal proliferation due to unrepaired DNA damage and resulting mutations. Extensive research highlights the connection between an individual's melanin levels, cumulative UV radiation exposure, and the risk of developing skin cancer.

In dermatology, early detection and treatment of cancers are of utmost importance. The use of advanced imaging techniques plays a crucial role in identifying specific characteristics of lesions and monitoring structural changes over time, enabling timely interventions. This thesis explores a variety of non-invasive imaging methods for comprehensive evaluation of lesions. Additionally, the study incorporates innovative technologies such as Vision Transformers (ViT), Few-Shot Learning, and semantic segmentation to enhance diagnostic accuracy and improve the detection rates of different skin cancers.

1.1 Structure of the work

The following paragraph illustrates how this thesis is composed:

- Chapter 1, *Introduction* which introduces to the goal of the thesis, focusing on Few-shot learning and Vision Transformers in computer vision.
- Chapter 2, *Background* where the goal is to introduce the reader to the basics concepts in the field of Machine Learning. Important subjects covered include deep learning, attention mechanism, and the foundations of image processing.
- Chapter 3, *Related works* introduces the reader to the advanced technologies involved in this work. The existing researches on few-shot learning, Vision Transformers (ViT), and their uses in computer vision is examined in depth in this chapter.
- Chapter 4, *Methods and Materials* describes the research procedures used in the study, including as the particular CNN and ViT architectures used, the datasets used, the few-shot learning techniques applied, and the assessment measures selected for evaluation performance.
- Chapter 5, *Experiments* presents the experiments that have been conducted. It explains the various experimental configurations, the pre-processing steps for the data, and the training methods that were employed.
- Chapter 6, *Results* discusses the results obtained after the experiments. It talks about trends that have been noticed, highlights important results, and offers statistical evidence to back up the conclusions.
- Chapter 7, *Conclusions and future work* presents the conclusion, with an open discussion on future works and possible implementations. The main conclusions of the thesis are outlined in this chapter, with special attention to the advancements made in the field of few-shot learning.

Chapter 2

Background

This chapter introduces the reader to some of the basic concepts of Machine Learning, Artificial Intelligence and Skin Cancer. In this chapter the basis of this work are described.

2.1 Skin Cancer: Overview

The aberrant proliferation of cells in the epidermis due to unrepaired DNA damage that results in mutations is known as *skin cancer*. The overwhelming body of research indicates that the relative quantity of melanin in an individual and their cumulative exposure to UV radiation are related to skin cancer. Skin malignancies are classified into two categories: non-melanoma and melanoma.

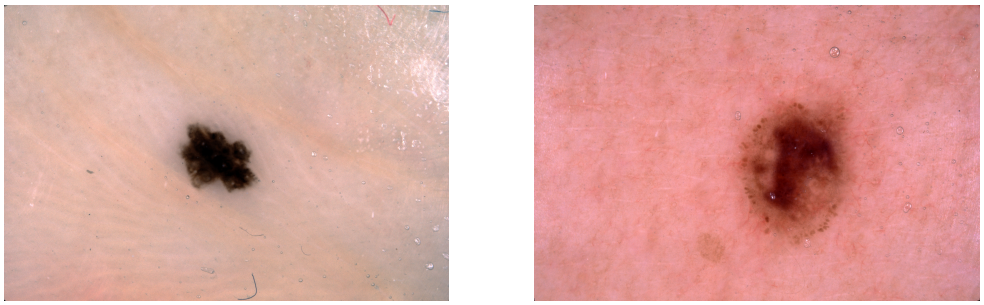
- **Melanoma**: the most fatal type of skin cancer, is a malignant tumorous development of melanocytes. This is because it has the ability to metastasize early, or quickly spread to other parts of the body. Although melanoma is heterogeneous, meaning it can take many different forms, the three most prevalent types are *nodular melanoma, lentigo maligna melanoma, and superfcial spreading melanoma.* Of those with melanoma diagnoses, 60% to 70% have Superficial Spreading Melanoma (SSM), the most prevalent kind. Fat lesions that first spread outward over the epidermis' upper layer rather than below are indicative of SSM. These lesions can be ignored in the early stages because they frequently initially mimic freckles or melanocytic nevi.

- **Non-melanoma (NMSC)**: is the most prevalent type of skin cancer in the United Kingdom, accounting for 155,985 new occurrences between 2016 and 2018. Squamous cell carcinoma (SCC) and basal cell carcinoma (BCC) are two of these types of skin cancer. BCC makes up 75% of yearly NMSC instances, with SCC making up the remaining 20%. Because these NMSC cases rarely

metastasize, their mortality rates are lower than those of more aggressive melanoma malignancy.

Other common types of skin cancer are:

- **Basal Cell Carcinoma (BCC)**: refers to keratinocyte cell tumors that typically develop on skin regions like the head and neck that have received significant amounts of UV exposure over time. BCC are often pearly, translucent, pink nodules with a cratered, ulcerated center. Nonetheless, BCC can be found in several subtypes, each with unique traits including pigmented, superfcial, and nodularcystic.
- **Squamous Cell Carcinoma (SCC)**: is a keratinocytic tumor and is the second most frequent type of skin cancer. Nevertheless, compared to BCC, this type of cancer has a greater death rate and the ability to spread. The frequent presentation of these lesions is either pink or skin-colored papules or plaques.



(a) Nevus. **(b)** Melanoma.

Figure 2.1: Example of nevus and melanoma malignancies from ISIC2017 dataset.

2.1.1 Skin Lesion Imaging

Imaging techniques can be used to analyze structural changes in a lesion over time and uncover characteristics that may help in the early detection and treatment of suspected malignancies. Non-invasive procedures enable investigation of the entire lesion architecture in addition to conventional physical biopsies of lesion tissue.

- **Clinical Images**: any picture of a lesion captured with a camera is referred to as a clinical image. This easy method can be used to control a lesion's evolution over time or to validate an inspection.
- **Dermoscopic Images**: can be used to look at a lesion's inner structures, which are apparent to the naked eye, and discover aspects that are hidden

from view. The portable equipment is made up of a microscope and a strong light source that enable deeper skin structures to be examined through skin surface microscopy. It has been demonstrated that using this method will raise the rate of BCC identification from 60% to 90%.

2.2 Machine Learning and Deep Learning

Machine Learning (ML) is a branch of *Artificial Intelligence* which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.
Deep Learning (DL) is a branch of machine learning that uses that replicates the elaborate decision-making capabilities of the human brain using multi-layered neural networks, or deep neural networks.

A key component of the expanding discipline of data science is machine learning. Algorithms are trained to produce predictions or classifications and identify important information in data mining projects using statistical techniques. Afterwards, these insights influence business and application decisions, which ideally affect important growth metrics.
There are different types of Machine Learning:

Supervised Learning: These algorithms are provided with data that includes the solutions of the problem, known as *labels*. The goal of the model is to identify a function that can map input values to the corresponding target label. To achieve this, these systems adjust their parameters during the training phase so that they can accurately associate each input value with the correct target label once the training is complete. It is mainly used in classification and/or regression problems.
Unsupervised Learning: Unsupervised learning models are trained on *unlabeled* data, meaning that the data does not have any associated labels. Consequently, these models do not attempt to assign labels to each input data point. Instead, their objective is to identify recurring patterns and uncover latent information within the dataset. It is mainly involved in clustering and/or association projects.
Semi-Supervised Learning: Type of learning algorithm that falls in between Unsupervised and Supervised Learning. These algorithms make use of *both* labeled and unlabeled data, with a small amount of labeled data and a larger amount of unlabeled data. Typically, these models are initially trained in an unsupervised way, and once all the labels have been assigned, they are further trained in a supervised way.
Reinforcement Learning: In contrast to the three preceding approaches, reinforcement learning algorithms involve the learning system, referred to as the

agent, attempting to acquire the knowledge of accomplishing a specific task by maximizing its *reward*. The agent achieves this by utilizing feedback received from the external environment, where positive feedback results in an augmentation of the reward, while negative feedback leads to a reduction in the reward.

2.2.1 Neural Networks

Neural Networks (NN) are a collection of algorithms designed to imitate the function of the human brain. A Neural Network consist of two main components:

- *Neurons:* they are the computational units.
- *Synapses:* they are the connections between neurons.

A *neuron* receives a set of input parameters, which are then multiplied by corresponding weights, combined ans subjected to a non-linear activation function, as shown in figure 2.2, composed by an input layer, an hidden layer and the output layer. The number of neurons in each layer might vary, except for the input and output layers. The number of the neurons in the input layer depends on the number of inputs that the model receives, the number of the neurons in the output layer depends on the desired number and type of outputs (in classification, for example, the number of neurons in the output layer would be equal to the number of classes or labels aimed to classify).

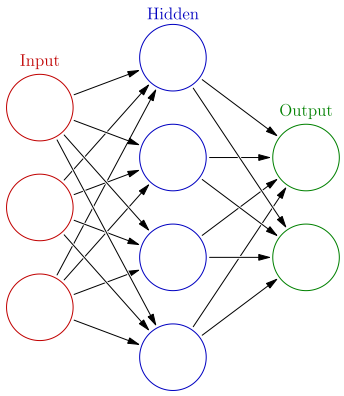


Figure 2.2: A simple Neural Network. [2]

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are widely utilized in the computer vision field, due to ther effectiveness. A key component of CNNs is the convolutional layer, that is where the convolution operations aids in reducing the number of features

in the model, retaining only the most important information. According to [3], this process enables the network to focus on small low-level features in the initial hidden layer and then combine this features to form more complex higher-level features in the subsequent layers. Using a hierarchical approach, allows to mirror the structure commonly found in real-world images, contributing to the superior performance of CNNs in computer vision applications.

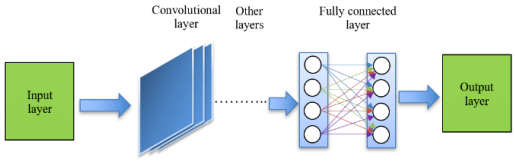


Figure 2.3: Generic architecture of CNN.

Convolutional Layer

The primary function of the *convolutional layer* in a Convolutional Neural Network is to extract high-level features from input images. In the initial hidden layers, the focus is on identifying low-level features such as edges and variations in intensity. As the network progresses to deeper layers, the objective shifts towards detecting more complex, high-level features that are of interest. To achieve this, the input image undergoes convolution with a *kernel* or *filter*, which is typically smaller in size compared to the input dimension. The kernel slides across the image, generating an activation map that indicates the presence and intensity of a specific feature within the input.

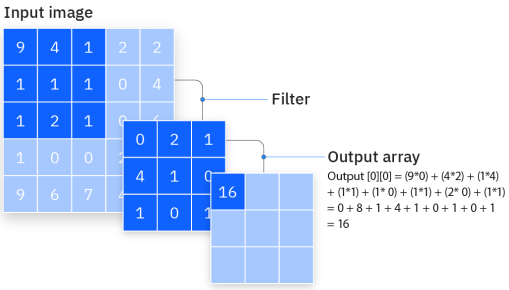


Figure 2.4: A convolutional layer. [4]

Pooling Layer

The primary purpose of the pooling layer is to reduce the spatial dimensions of the feature maps by summarizing their content. Consequently, the model does

not need to learn features at precise positions, thereby establishing the translation invariance of CNNs. The two prevailing strategies employed in pooling are:

- *Max Pooling*: it returns the maximum value of each block of the feature map.
- *Average Pooling*: it returns the mean of each block's values of the feature map.

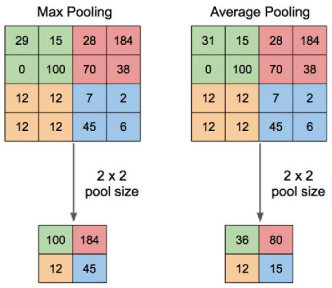


Figure 2.5: Example of max and average pooling. [5]

Fully Connected Layer

Fully connected layers establish connections between all inputs from a preceding layer to each activation unit in the subsequent layer. In this type of layer, weights are not shared and are typically positioned at the conclusion of a model. These layers are commonly employed to acquire non-linear combinations of the advanced features.

Activation function

The activation function, a crucial component, is applied following every convolutional layer to assist the model in capturing intricate patterns from the input data. In order to achieve this, we enhance our neural network with a non-linear function. Various activation functions are needed for both hidden layers and output layers, with the prevailing best practices being:

- *Rectified Linear Unit* (RELU): The hidden layers make use of the function $a(x) = max(0, x)$ for their operations. This function is defined as the maximum value between 0 and x. Its purpose is to address the issue of gradient saturation that was observed in the sigmoid activation function, which was used before the current function.
- *Sigmoid*: The activation function employed on the output layers for binary classification problems is chosen due to its computational efficiency. Specifically,

it is defined as $\sigma(x) = \frac{1}{1+e^{-x}}$, effectively restricting the output values within the range of 0 to 1.

- *Softmax*: The activation function employed in the output layer is utilized to enhance the performance of multi-class classification problems. In fact, it is regarded as the extension of the sigmoid function for scenarios involving multiple classes. This function is defined as $a(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$, where i represents the current class and $j = 1..n$ denotes all the potential classes.

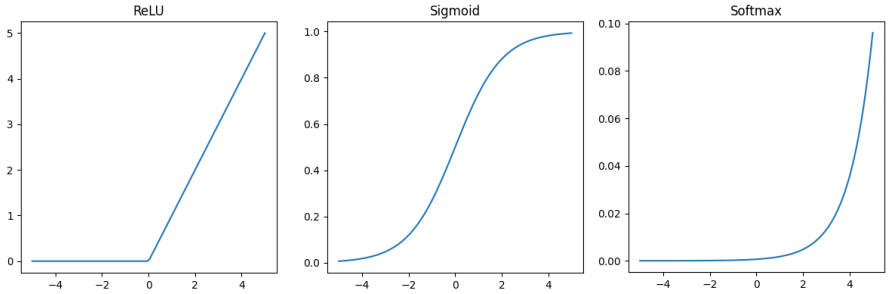


Figure 2.6: Activation functions.

2.3 Transformers and Vision Transformers

Vision Transformers, shown in figure 2.7, have been introduced in [6] and their mechanism was proposed in [7]. Transformers became one of the most used architectures in Natural Language Processing (NLP) field, in particular BERT, introduced in [8], became state-of-art. The Transformer is a model that uses attention 2.4 to boost the speed which these models can be trained, yet the most significant benefit comes from how a transformer lends itself to parallelization. After this, [6] created the Vision Transformers (ViTs) that are Transformer-based architectures intended to prove that convolutions are not strictly necessary to obtain optimal results. The ViT also requires less computational resources than a Convolutional Neural Network (CNN). The following is a short overview of what a Vision Transformer (ViT) architecture is:

1. Split the input image into patches of fixed-size
2. Flatten all the patches
3. From the flattened patches, create lower-dimensional linear embedding
4. The sequence if fed to a Transformer encoder

5. There is a pre-training phase for the ViT model using the image labels

The Vision Transformer (ViT) treats image patches as a sequence of tokens, similar to how words are processed in natural language. Unlike the original Transformer, which uses 6 encoder layers, ViT typically has at least 12 encoder layers and does not include a decoder. However, one of the problems that characterize the ViT is the computational *bottleneck* caused by its global self-attention mechanism, which requires every image patch to attend to every other patch. This is eventually solved by Swin Transformer [9], which introduces a more efficient approach to self-attention.

2.3.1 Swin Transformer

Swin Transformer, whose architecture is illustrated in figure 2.8 overcome the issue of computational complexity of self-attention, that depends on the image size, making it suitable also for the case of large images or dense prediction such as *segmentation* problems.
The key feature of the Swin Transformer is that the architecture computes self-attention via non overlapping windows, shifting between consecutive layers. The results is a more efficient implementation.

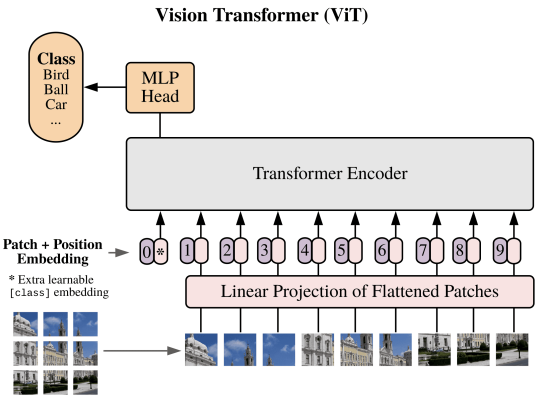


Figure 2.7: Vision Transformer architecture. [6]

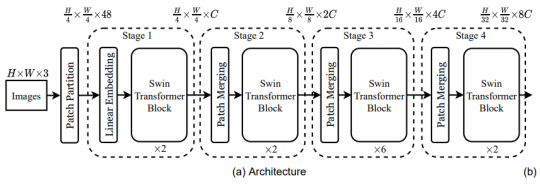


Figure 2.8: Swin Transformer architecture. [9]

The main component of a Swin Transformer are:

- **Patch Partition**, that is the component that creates non-overlapping patches.
- **Linear Embedding**, applied to raw valued features after the patch partitioning phase.
- **Patch Merging**, to reduce the number of tokens.
- **Swin Transformer Block**, that is basically the most important part since it replaces the standard multi-head self-attention (MSA) module of a Transformer. The block idea is to divide the image into windows of a fixed-size patches and perform self-attention between patches inside of every window. The attention windows are then shifted with respect to previous layer, similar to what happens in strided convolutions.

2.4 Attention Mechanism

The attention mechanism is introduced in [7]. The attention mechanism is now behind all transformer models and it is a technique that allows the model to focuses on the relevant parts of the input sequence as needed.
Sequence-to-sequence model and attention model differs in two main points:

1. First, the encoder passes many more data to the decoder. Instead of passing the last hidden state of the encoding stage, the encoder passes all the hidden states to the decoder.
2. An attention decoder performs an extra step before producing its output. In order to focus on the parts that are relevant to this decoding time step, the decoder carries on the following operations:
   (a) Examine the collection of encoder hidden states that were received; the majority of each encoder hidden state are connected to a certain part in the input text.
   (b) Assign a score to every hidden state, usually with a *softmax* function, to produce softmax scores.
   (c) Each hidden state should be multiplied by its softmaxed score in order to magnify high-scoring hidden states and obscure low-scoring hidden states.

In order to have a complete overview, this is what happens in the most relevant part of the architecture: the decoder.

1. The $< END >$ token embedding and the initial decoder hidden state are fed into the attention decoder.

2. After processing its inputs, the CNN generates an output along with a fresh hidden state vector. The result is thrown away.

3. **Attention Step**: A context vector is computed for this time step using the previous hidden state vector and the encoder hidden states.

4. The context vector and the hidden state vector are concatenated into one vector.

5. This vector is fed into a feedforward neural network that was trained in tandem with the model.

6. The output word for this time step is shown by the feedforward neural networks' output.

7. Repeat for the next time steps.

Different types of attention, have been presented in the literature:

- **Self-Attention**: this type of attention mechanism considers various positions of the same input sequence to generate a representation of it, contrasting with the traditional additive attention which primarily focuses on the alignment between input and output positions in a sequence. This form of attention is extensively utilized not just in natural language processing tasks, but also in computer vision, and forms the fundamental component of the Transformer architecture

- **Hard vs Soft Attention**: two distinct types of attention mechanisms, known as Hard and Soft Attention, were introduced in [10] for the purpose of image captioning. These mechanisms aim to determine which parts of the image to focus on in order to generate meaningful captions. The key difference between hard and soft attention lies in the level of access the attention block has to the image. In *soft attention*, the alignment weights are learned by considering the entire image, resulting in smooth and differentiable computations. However, a drawback of soft attention is that it becomes computationally intensive when dealing with large inputs. On the other hand, *hard attention* processes one patch of the image at a time, allowing for faster computations. However, due to its non-differentiable nature, hard attention requires more sophisticated training methods.

- **Global vs Local Attention**: another distinction in attention mechanisms is between Global and Local attention. In *Global attention*, similar to soft attention, the output of the attention block is computed by considering all source states from the encoder and all decoder states prior to the current state.

In contrast, *Local attention*, resembling hard attention, only utilizes a window of positions from the encoder to compute the output. The advantage of local attention over hard attention is that it is differentiable.

2.5 Semantic Segmentation

Semantic segmentation is the process to assign predefined semantic labels to each pixel in an image. After this process, the input is used to semantically group pixels and analyse data such as 2D, 3D and video [11]. Semantic segmentation is associated with popular computer vision tasks such as image classification, object detection, instance segmentation, and panoptic segmentation, which all allow the identification of individuals, objects, etc. within the input data.

There have been numerous advancements achieved in this time- and resource-consuming activity throughout the years. Because it enables the automatic or semiautomatic extraction of the lesion's annotation, this task is crucial to skin cancer imaging. This task is not limited to tumor segmentation; it may also be used for more complicated cases such as organ segmentation, where more labels and tighter borders need to be handled.
Boundary extraction, region-based segmentation, threshold-based segmentation, and other techniques are among the principal early traditional medical picture segmentation techniques [12]. A new generation of image segmentation models, including *FCN, U-Net*, and their variants, were created as deep learning networks advanced, and their segmentation performance was much enhanced.

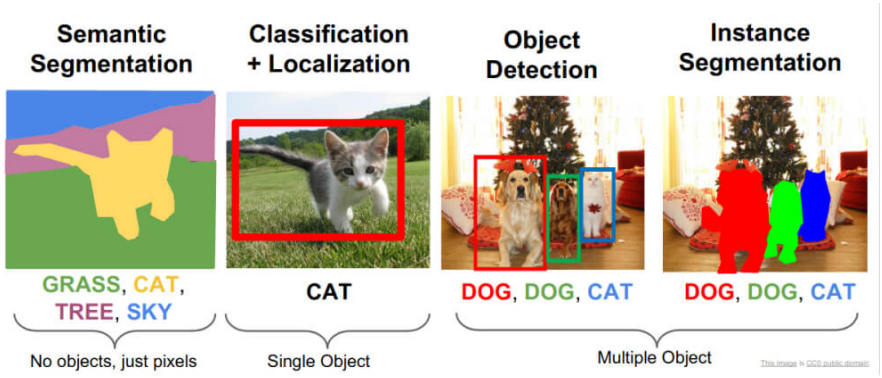


Figure 2.9: Difference between different computer vision tasks. [13]

Using known sample data, the deep learning-based image segmentation approach establishes the mapping between individual pixels and their corresponding categories. This means that the deep learning model analyzes each pixel in the image and

assigns it to a specific class or instance, such as "cat", "dog", "background" or "car". This type of approach makes use of deep learning's potent nonlinear fitting capability and trains on a sizable amount of sample data.

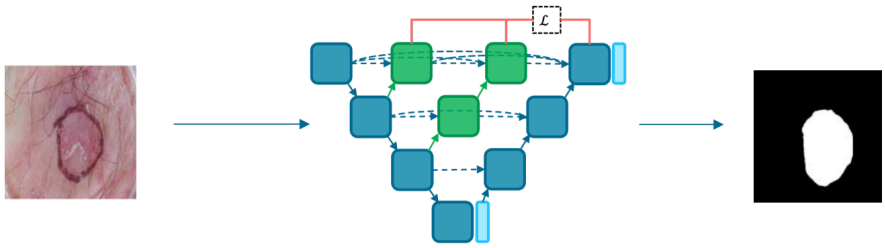


Figure 2.10: Segmentation pipeline. The image is acquired from ISIC-2017 dataset and is given as input to a semantic segmentation architecture, to obtain the lesion mask.

Chapter 3

Related work

This section presents recent researches that are relevant to this thesis, with an emphasis on the particular methods and approaches applied in Few-Shot Learning for semantic segmentation of skin cancer. It highlights the function of vision transformers in this context and discusses developments in Few-Shot Learning, its applications, and deep learning techniques for skin lesion analysis.

3.1 Few-shot Learning

Few-shot Learning (FSL) is a Machine Learning framework that enables a pre-trained model to generalize over new categories of data, so new type of data that the pre-trained model has not seen during training, using only a few labeled samples per class. It falls under the paradigm of **meta-learning**, which means that the model learns how to learn more efficiently from few examples, acquiring a kind of experience on how to tackle new tasks or classes with a limited number of examples. The model is trained on a small training set, with the goal to know the similarities and differences among the given objects, which differentiate from the typical goal, i.e to classify a certain animal species, for example.
Eventually the model will be able to identify the similarities between two images. The main goal is then to understand how similar images are, and at the end of the training, the model will be able to tell if two images are the same or not.
There are several variants of few-shot learning, including:

- **One-shot learning**: The model is trained with only one example per class.
- **Few-shot learning**: The model is trained with only a few examples per class, usually a very small number, such as 5 or 10 examples per class.
- **Zero-shot learning**: The model can generalize and make predictions on classes not seen during training.

A peculiarity of this approach is certainly the fact that the model is trained with a significantly smaller number of samples than normal, this is possible thanks to the use of different techniques:

- **Transfer Learning**: is a technique that employs the acquired knowledge of a trained model to learn a different dataset. The primary objective of transfer learning is to enhance the learning process in the target domain by harnessing the knowledge gained from the source domain and learning task.
- **Siamese Networks**: These models are designed to compare pairs of images and learn to measure their similarity. They are trained on pairs of images to learn to distinguish between different classes. Such architectures can be adapted for Few-shot learning, evolving into few-shot-based support classifiers.
- **Prototypical Networks**: This technique creates prototypical representations for each class using the extracted features from training images. It calculates the prototype for each class and then uses the distance between the prototype and new images to make predictions for unseen classes.
- **Attention Mechanisms or Progressive Update**: Some approaches utilize attention mechanisms or progressive updating to focus on relevant parts of images or gradually adapt the model to new data, enabling better adaptation with a few training examples.
- **Augmentation and Generative Models**: Leveraging data augmentation techniques to generate more examples from limited data or using generative models such as Generative Adversarial Networks (GANs) to increase the number of training instances.

These strategies enable models to learn from a limited number of training examples, employing intelligent techniques to maximize information extraction and generalization from the available data.
The available data need to be grouped into specific sets for this type of approach, although a training set and a test set are still used.

- **Support set**: few labeled samples per novel category of data on which the model needs to generalize using previous knowledge information and information gained from the support set
- **Query set**: both new and old data on which the model needs to generalize using previous knowledge and information gained from the support set
- **N-way K-shot learning scheme**: "N-way" indicates that there are N numbers of novel categories on which a pre-trained model needs to generalize

over. A higher N value means a more difficult task. "K"-shot defines the number of labeled samples available in the support set for each of the N novel classes. The few-shot task becomes more difficult (that is, lower accuracy) with lower values of K because less supporting information is available to draw an inference. K usually is in the range of one to five.

There is a difference between training set and support set:

- Each class in the training set has many examples.
- The training set is large enough to learn a deep neural network.

The support set, instead, is small and can only provide additional information at test time. The most important reasons to use Few-Shot-Learning is to avoid domain shift in the test set.
Few-shot learning mitigates the above problems in the following way:

1. Training a model does not require large amounts of costly labelled data because, as the name implies, the goal is to generalise with only a few labelled samples.
2. Since a pre-trained model (trained on a large dataset, e.g. ImageNet) is extended to new data categories, a model does not need to be trained from scratch, which saves a lot of computing power.
3. With the help of FSL, models can also learn about rare data categories for which only limited prior information is available. For example, data on endangered or newly identified animal/plant species is rare and this is sufficient to train the FSL model.
4. Even if the model was previously trained with a statistically different data distribution, it can be extended to other data ranges as long as the data in the support and query sets are coherent.

There are also different approaches for Few-Shot Learning:

- **Data level**: If the training of an FSL model stalls due to lack of training data (and to prevent overfitting or underfitting), more data can be added - which can be structured or unstructured. Suppose there are two labelled samples per class in the support set, which may not be enough. So it is possible to try extending the samples using different techniques.
- **Parameter-level**: Parameter-level FSL approaches involve the use of meta-learning, which guides the use of model parameters to intelligently determine which features are important for the task at hand.

- **Metric-level**: Metric-level FSL approaches aim to learn a distance function between data points. The features are extracted from the images and the distance between the images is computed in the embedding space. This distance function can be Euclidean distance, ground motion distance, cosine similarity based distance, etc.

- **Gradient-based meta-learning**: Gradient-based meta-learning approaches use two learners - a teacher model (base learner) and a student model (meta-learner) using knowledge distillation. The teacher model guides the student model through the high-dimensional parameter space.

- **One-shot learning**: One-shot learning is a task in which the support set consists of only one data example per class. You can imagine that the task is more complicated with less supporting information. The facial recognition technology used in modern smartphones utilises one-shot learning.

3.2 Examples of Few-shot Learning application

A first approach with Few-shot learning comes from the paper Few-shot Classification of Skin Lesions from Dermoscopic Images by Meta-Learning Representative Embeddings [14], which presents a new machine learning method to classify skin lesions by means of meta-learning technologies. The main problem, as in many cases in medical imaging, is that of data availability: few annotated images exist for not only new but also existing diseases. The approach used in their case is to transfer new learning tasks to different epochs, this is referred to as meta-training. Once the meta-training phase is over, it is possible to move forward to the meta-testing phase, i.e., testing all the tasks with which the model has been trained to evaluate its performance.
A new addition to the meta-learning introduced is Model-Agnostic Meta-Learning (MAML) [15], designed to train models that can quickly adapt to different tasks using a small and limited amount of data. The idea behind MAML is the optimisation of initial model parameters, such as gradient steps. This basically means that the model is trained to be easy to fine-tune.
Few-shot Learning is used in multiple applications and in different fields. In this section, some works concerning medical imaging in particular will be analysed. The first to be introduced is Prototype-based Incremental Few-shot Semantic Segmentation [16]. This introduces an approach called incremental few-shot learning, redirecting it into a semantic segmentation context. In particular, the authors propose a method called Prototype-based Incremental Few-shot Segmentation (PIFS) that consists of combining prototype learning and knowledge distillation. By doing so, the model is able to learn new classes from limited data with the

knowledge of previously learned information.
In Semi-supervised few-shot learning for medical image segmentation [17] a new approach to Few-shot semantic segmentation, specifically in the medical imaging domain, is presented. The proposed solution integrates unlabeled data with what are called *surrogate tasks*, a technique inspired by self-supervised learning that involves the use of many unlabeled medical images to improve the model's performance in learning more generalizable features. It is also involved episodic training, which is a method commonly used in few-shot learning where each episode simulates a Few-shot learning scenario and then the model weights are updated and then repeated.

3.3 Deep Learning methods for Skin Lesion analysis

In [18] the authors introduce two networks designed to aid in melanoma detection using dermoscopy images. The first one is called Lesion Indexing Network (LIN) and it is dedicated to segmentation and classification, using two fully convolutional residual networks to produce both output, from segmentation and classification. The LIN uses a novel technique called Lesion Indexing Calculation Unit (LICU) that refines the classification assigning weight to each pixel based on its distance from the lesion border.
The second network is called Lesion Feature Network (LFN) designed for dermoscopic feature extraction, involving convolutional neural networks (CNNs) to identify four key dermoscopic features: Pigment Network, Negative Network, Streaks, and Milia-like Cysts. It also uses a superpixel segmentation method to divide the image intro smaller region to classify better, based on the presence of these features.
Both LIN and LFN were evaluated on the ISIC 2017 dataset, demonstrating promising results in both lesion segmentation and dermoscopic feature extraction. It also introduces a different pair of values for mean and standard deviation, values that will be tested in this thesis.

Chapter 4

Methods and Materials

The datasets that were used and the changes made to the two papers [19] and [20] will be described in this chapter. These relate to the data augmentation done on the images, the adjustments made to the data loader, and the fine-tuning of the settings. The details of each modification will subsequently be covered in depth in the experiments chapter. The aim of this thesis is to demonstrate how a model, using methodologies such as Vision Transformers and Few Shot Learning techniques can be effective in detecting skin cancer, through segmentation techniques. In particular, two architectures were considered: one involving the combined use of CNN and ViT, the other involving the use of a dedicated Few-shot architecture. Both architectures will be analysed as well as their changes, and the work will be divided into two parts: the first in the training of three models with three different datasets of medical images only, the second consists of testing these models with an extremely small number of images per class, thus adapting a real scenario.

4.1 Analyzing Convolutional Neural Networks and Vision Transformers

One of the challenges in this work is overcoming the obstacle of the restricted availability of medical images by utilizing contemporary methods that may result in the development of a model that must be built from the ground up or that already exists.
Even though semantic segmentation is a difficult task in and of itself, there are contemporary methods that can help, such as vision transformers. But it's important to weigh the benefits and disadvantages of this approach.
Vision Transformers (ViT), described in 2.3, seem to offer significant improvements in terms of efficiency and accuracy metrics, but they also come with a risk of losing local features, which architectures like Convolutional Neural Networks, explained

in 2.2.2, are more efficient at capturing. On the other hand, because of their self-attention mechanism, these architectures are particularly good at capturing long-range relationships between different parts of an image. In contrast to convolutional neural networks, this enables ViT to have a more global knowledge of the image.

To combine the strengths of CNNs and Transformers, hybrid architectures like TransUNet and DS-TransUNet emerged, incorporating CNNs for spatial features and Transformers for global information encoding. TransFuse [21] proposed a BiFusion module to combine CNN and Transformer features.
The paper [19] introduces CoTrFuse, which leverages Swin Transformer and EfficientNet as dual encoders. It introduces the Swin Transformer and CNN Fusion module (STCF) to fuse global and local semantic information effectively before the skip connections, enhancing segmentation performance. Experimental results on skin lesion and COVID-19 infection segmentation datasets demonstrate that CoTrFuse outperforms state-of-the-art segmentation methods, making it a promising approach for medical image segmentation.
The main strenght points of this architecture are:

- **Overall Architecture Design**: CoTrFuse's architecture includes two parallel branches, namely the Transformer Branch and the CNN Branch, for feature extraction. The extracted features from these branches are fused using a module called Swin Transformer and CNN Fusion (STCF). After fusion, the multi-level feature maps are passed through the skip connection and Decoder block for segmentation.

- **EfficientNet Block**: The EfficientNet block is used for feature extraction in the CNN Branch. It comprises multiple MBConvBlocks, including convolution, batch normalization, dropout, Swish activation, and a Squeeze and Excitation block for advanced feature capture. The compound scaling method is employed to optimize model parameters effectively.

- **Swin Transformer Block**: Swin Transformer is introduced as an efficient alternative to the standard multi-head self-attention module used in Transformers. It includes the windows multi-head self-attention module (W-MSA) and the shifted windows multi-head self-attention module (SW-MSA) to enhance performance. These modules facilitate information exchange between different windows within a feature map.

- **Swin Transformer and CNN Fusion Module (STCF):** To effectively fuse features from the Transformer and CNN branches, a novel module called STCF is proposed. It leverages spatial attention mechanisms (CBAM blocks) to exploit spatial relationships between feature maps. The STCF module

consists of spatial attention, channel attention, and feature recalibration processes, ultimately combining features from both branches to achieve improved segmentation performance.

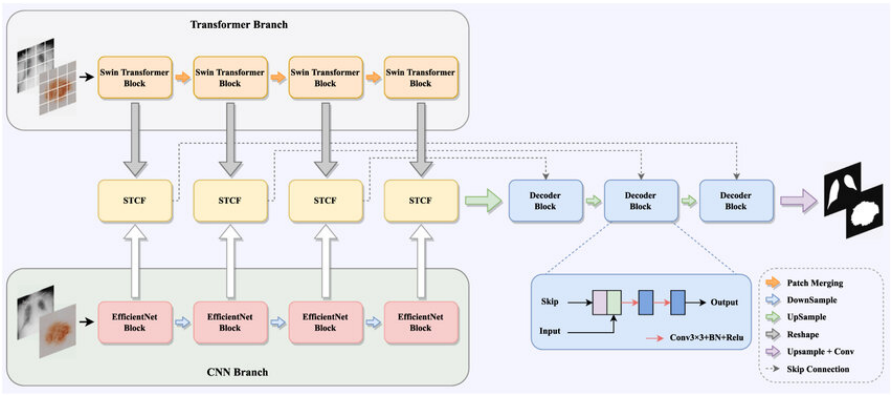


Figure 4.1: The proposed CoTrFuse architecture. [19]

4.1.1 Specifics of the changes and implementations

The dataloader has undergone the first modification in this effort in order to optimize it. The initial optimization was to construct a "on demand" dataloader, which only takes the stated amount of photos (for local test purposes) and only when it is needed. The original dataloader used to collect all the images when the training file was called.

Furthermore, the dataset fitting script needs to be improved to incorporate necessary features for efficient training and evaluation steps. The ultimate version seeks to ensure effective data loading straight into the GPU. Additionally, by concentrating only on CUDA devices, this updated version optimises image loading and reduces dependence on the CPU to improve consistency and performance.

By making this change, the images' time to load could be shortened, as seen above. Please note that loading time may be different in other set-ups. Adding plots of the values for *train loss*, *validation loss*, and *intersection over unit* was the second, very little change.
After that, more backbones might be added in addition to the first one (EfficientNet B0). In particular, it was possible to determine that *ResNet-50* was the ideal backbone for this work after conducting several experiments, that will be discussed in the next chapter.
Once several models were found to be compatible, they could be added as a command line before training began, making it simple to switch among models.

Finally, it was possible to adapt this model also to the ISIC2018 (4.4.2) dataset.

Dataset	Dataloader	Time (seconds)
ISIC 2017	Original	2756.45
	Modified	**2452.9**
ISIC 2018	Original	N/A
	Modified	2659.23
COVID-QU-Ex	Original	351.7
	Modified	**258.47**

Table 4.1: Results demonstrating faster images loading times.

The final goal of this paper was to train the model with COVID-QU-Ex dataset and then use the very same model for Few-Shot Learning with the two ISIC dataset, and then finally a comparison with and without Few-Shot Learning techniques. Each of the changes discussed will will be explored further in Chapter 5.

4.2 Integration of Few-shot Segmentation

What has been described so far, however, does not solve one of the first problems posed: the solution to the lack of data for medical image segmentation.
To overcome this issue, Few-shot Segmentation techniques are a good fit, since with an extremely small amount of data still attempt to segment correctly thanks to an architecture that allows the model to undergo meta-training, in which the model is trained with images that may or may not even be medical and then used with medical datasets, doing what is called *Cross Domain*. The con in this case is found in the previous architecture, as it is not prepared for Few-shot Segmentation.

To introduce then Few-shot Segmentation, it was possible to use an architecture called DMTNet [20]. A cross-domain experiment was conducted using a trained model for Few Shot Segmentation using non-medical images but with medical images through meta-training and meta-testing.
The previously mentioned research presents a novel architecture composed of three modules:

- Self-Matching Transformation (SMT)
- Dual Hypercorrelation Construction (DHC)
- Test-time Self-Finetuning (TSF)

To transform independent domain features, the Self-Matching Transformation module learns a transformation matrix for each background and query image. However, correlations between the query, the background, and foreground are created in the Dual Hypercorrelation Construction section. On the other hand, in meta training, a CNN processes the support and query images to extract multi-level features, after which SMT learns a transformation matrix, DHC performs the correlations, and an encoder and decoder are used to generate the mask. In contrast, Test-time Self-Finetuning (TSF) is employed in meta testing to fine-tune the parameters while the test is running, and the final mask is obtained by refining the meta-training's raw mask.
Going into the details, this is what each step does.

4.2.1 Self-Matching Transformation

This module reduces dependency on support features in situations where SMT additionally makes use of query data to prevent overfitting or in cases where the support set is excessively small or differs from the query
Produces a rough mask using query-first-background similarity in the manner described below:

1. Calculates foreground and background prototypes on a global and local scale.
2. Divides support feature maps into several local feature maps to improve prediction accuracy.
3. Creates an initial query mask by fusing background and foreground data.
4. Computes correlation maps between local support prototypes and query.
5. Makes use of a Binary Cross Loss Entropy

Finally, it transforms the features in an adaptive way, that is, to make the features independent of the domain, both for the query set and the support set.

4.2.2 Dual Hypercorrelation Construction

The first thing to examine is the background correlations, which are based in part on the hypothesis that objects in the same category tend to find themselves in similar environments; as a result, the query background and image support may be useful; then the module constructs the dual correlations between the query images dense features and the support image's foreground and background features.
Then, using the cosine similarity, calculate the correlation between the primary characteristics of the support image and the query images characteristics. Finally,

compute the correlation between the secondary features of the support image and the query image's features, computed in a similar manner.
As last step, it generates the query's prediction mask, foreground, and backdrop.

4.2.3 Test-time Self-Finetuning

It is divided into two parts:

1. The model creates the previous support masks and updates the network using a loss function based on the Binary Cross Entropy (BCE) between the previous support masks and the ground truth.

2. The entire network is frozen and the final prediction for the query image is executed in an attempt to increase only a few encoder parameters.

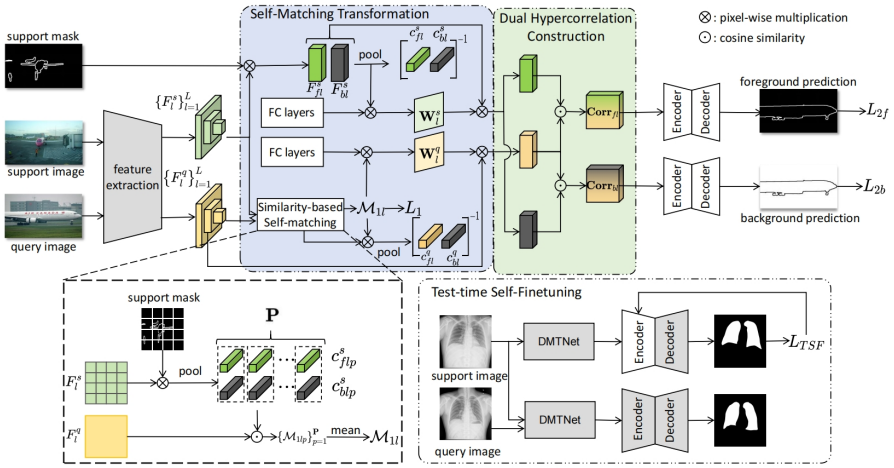


Figure 4.2: Architecture of DMTNet. [20]

4.2.4 Specifics of the changes and implementations

In this case, the ISIC2018 dataset's preprocessing was the first modification applied. This is because the ISIC2018 dataset contains three classes, whereas the original technique only allowed for splitting into two. As a result, the class of seborrheic keratosis was introduced in the new splitting process.

After that, the model was adjusted to conduct experiments with ISIC2017 as well. This was made possible by the similarities with ISIC2018, but in this case, the nevus class was absent, which is meanwhile melanoma and seborrheic keratosis were present. However, the modification has been feasible and has taken a short

period of time.

The data augmentation was completely changed for both the ISIC2017 and ISIC2018 datasets. The chapter on experiments - precisely in 5.2.1 - will provide a detailed explanation of the data augmentation process.

4.3 Metrics and evaluation

4.3.1 Semantic Segmentation

To evaluate the segmentation (2.5) performance, the metric used are:

- Dice coefficient (Dice)
- Mean Intersection over Unit (mIoU)
- Precision
- Recall
- F1-Score
- Pixel Accuracy (PA)

The calculation of the metrics described was done using a confusion matrix that provides:

- *True Positive (TP)*: the sample was predicted as positive and belongs to the class.
- *True Negative (TN)*: the sample was predicted as negative and belongs to the class.
- *False Positive (FP)*: the sample was predicted as positive but it does not belong to the class.
- *False Negative (FN)*: the sample was predicted as negative but belongs to the class.

Dice Coefficient

The Dice coefficient in semantic segmentation calculates the overlap between the ground truth and expected segmentations. It's preferred because of how well it handles class imbalance and is easy to understand. Nevertheless, it overlooks complex spatial linkages and can be sensitive to little variations in broad segmentations.

Notwithstanding its drawbacks, it is nevertheless a useful instrument for testing and refining segmentation models, which advances the comprehension of images with more precision and significance.

$$Dice = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{4.1}$$

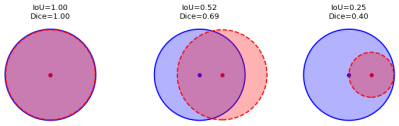


Figure 4.3: Example of a Dice Coefficient. [22]

Mean Intersection over Unit

A critical statistic in medical image segmentation, Mean Intersection over Unit (mIoU) measures the average overlap between predicted and ground truth masks across all classes, and is used to assess model accuracy. It is useful for medical applications because of its interpretability and capacity for handling multi-class segmentation, but it can also be sensitive to class imbalance and miss other important elements of segmentation quality.

$$mIoU = \frac{TP}{TP + FN + FP} \tag{4.2}$$

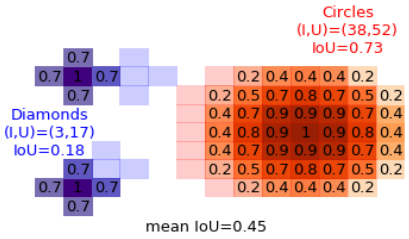


Figure 4.4: Example of mIoU. [22]

Precision

Precision in medical image segmentation refers to the proportion of actually positive pixels that a model classifies as positive. Reduced false positives from high precision

imaging are essential for preventing misdiagnosis in medicine. For best results, though, recall and precision must be balanced.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

Recall

Recall, or sensitivity, is a term used to describe how successfully a model recognizes all true positive instances in medical image segmentation. In medical imaging, high recall reduces false negatives, which is essential to prevent missed diagnosis. But concentrating only on memory may lead to more false positives. For segmentation to be effective and dependable, recall and precision must be balanced so that the model identifies the majority of true positives while minimizing the number of false positive predictions.

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

F1-Score

The F1 score is a crucial parameter in the field of medical image segmentation that balances the sometimes conflicting goals of recall and accuracy. It functions as a thorough evaluation of a model's correctness, encompassing both precision—the ability to recognize positive cases—and recall—the ability to collect all real positive examples.
The F1 score provides a fair evaluation that does not prioritize one metric over another. It is computed as the harmonic mean of accuracy and recall. This delicate balance is especially important in medical imaging, where misdiagnosing healthy tissue as sick (false positives) and failing to detect real illnesses (false negatives) can have devastating effects.

$$F1 - score = 2 \times \frac{Recall \times Precision}{Precision + Recall} \tag{4.5}$$

A high F1 score shows that the model achieves the best possible balance between recall and precision, exhibiting both comprehensiveness in catching actual positive cases and accuracy in positive predictions. This is especially important for medical applications, as precise and trustworthy segmentation is essential for patient outcomes, treatment planning, and diagnosis.
Essentially, the F1 score serves as a complete indicator of a model's efficacy in medical picture segmentation, capturing its capacity to precisely and thoroughly detect regions of interest.

Pixel Accuracy

Pixel accuracy measures the percentage of correctly categorized pixels and offers a basic evaluation of a model's overall accuracy in the field of medical imaging. It provides a fast indication of how closely ground truth and projections match. For a thorough assessment of segmentation models in the critical medical setting, additional metrics must be used due to its shortcomings, which include sensitivity to class imbalance and disregard for spatial linkages.

$$PA = \frac{TP + TN}{TP + TN + FN + FP} \tag{4.6}$$

4.3.2 Few-shot Segmentation

To evaluate the Few-shot segmentation, the metric used are:

- Mean Intersection Over Unit (mIoU), described in 4.3.1.
- Foreground/Background Intersection Over Unit.

Foreground/Background Intersection Over Unit

Foreground/Background Intersection Over Unit (FBIoU) is a measure that's used to assess how well models of semantic segmentation work, especially when few-shot learning is involved. It calculates the intersection, divided by the union of the expected foreground and background regions and the associated ground truth regions. It is calculated independently for the foreground and background regions, providing information on the model's ability to accurately segment both the target and the surrounding area.
Regarding the use in detail, it is possible to say that:

- Foreground: the pixels in an image that belong to the object of interest.

$$FIoU = \frac{TP}{TP + FP + FN}$$

- Background: the pixels in an image that *do not* belong to the object of interest.

$$BIoU = \frac{TN}{TN + FP + FN}$$

- Foreground/Background: combination of the two metrics above, and can be calculated as the simple average.

$$FBIoU = \frac{FIoU + BIoU}{2}$$

4.4 Datasets

In this section, the public datasets used for this work are described. The datasets shown are:

- ISIC 2017
- ISIC 2018
- COVID-QU-Ex-Dataset
- PASCAL Visual Object Classes

The ISIC and COVID-QU-Ex datasets were first used in CoTrFuse architecture during both train and test phases, producing three models. These models will be then used in DMTNet architecture to test the Few-shot Segmentation and do a comparison with the original DMTNet model, trained on PASCAL Visual Object Classes. In the following subsections all the datasets are described.
To simulate extremely limited availability of even these data, the models tested after training with the datasets will contain no more than five examples per class. Please be aware that the datasets that will be presented contain a large number of images that serve as examples for what will be done during the segmentation phases.

4.4.1 ISIC2017

The *International Skin Imaging Collaboration (ISIC)* published the ISIC 2017 dataset, a sizable collection of dermoscopy images. 2,000 pictures for training using ground truth segmentations (2000 binary mask images) represent the Task 1 challenge dataset for lesion segmentation.
The dataset was released in 2017 by the IBM T. J. Watson Research Center, Emory University, University of Central Arkansas, Kitware, Memorial Sloan-Kettering Cancer Center, Missouri University of Science and Technology, USA, and Medical University of Vienna, Austria.

This dataset will be used to train and test on CoTrFuse and will then be used in the DMTNet architecture to carry out tests with the few shot segmentation, using the ISIC2017 dataset and the model trained on ISIC2017.

Training Data
The training data are divided in two sections:

- Dermoscopy Image Data

- Ground Truth Segmentations

Dermoscopy Image Data
The images provided are 2000 and are used as training data; particularly this part of the dataset contains dermoscopic lesion images in JPEG format named used the scheme **ISIC_<image_id>.jpg**, where the image id represents a 7-digit unique identifier.
All the images are of the specific class of "lesion", since there is only one class in the dataset.

Ground Truth Segmentations
The ground truth is composed by 2000 binary mask images named **ISIC_<image_id>_segmentation.png** where the image id matches the corresponding Training Data image for the mask.
All mask images have the exact same dimensions as their corresponding training image and are encoded as single-channel 8-bit PNGs where each pixel is:

- 0: that would be the background of the image or the areas outside the lesion
- 255: that would be the foreground of the images or the areas inside the lesion

The masks are created by an expert clinician using a semi-automated process or a manual process. The difference is that in the first process is used an algorithm called flood-fill algorithm, on the other hand the masks are from a series of user-provided polyline points.

Evaluation and selection of the images
The segmented images have been compared to following variety of metrics:

- Sensitivity
- Specificity
- Accuracy
- Jaccard index
- Dice coefficient

In this work there are three sets of data utilized that would be *training set*, *validation set*, and *test set*, composed as follows:

- Training set: 2000 images (and 2000 labels)
- Validation set: 150 images (and 150 labels)

- Test set: 600 images (and 600 labels).

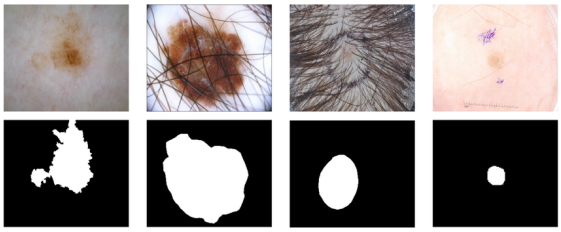


Figure 4.5: Some samples of the training set (dermoscopy image) and their respective mask (ground truth) from the dataset. [23]

4.4.2 ISIC2018

The *International Skin Imaging Collaboration (ISIC)* published the ISIC 2018 dataset, a sizable collection of dermoscopy images. The lesion segmentation task is shown in this Task 1 dataset. There are 2594 pictures in it.
As for ISIC2017, this dataset will be used to train and test on CoTrFuse and will then be used in the DMTNet architecture to carry out tests with the few shot segmentation, using the ISIC2018 dataset and the model trained on ISIC2018, and then compared with the baseline.

Input data
The input data are dermoscopic lesion images, such as in 4.4.1. Also in this case, the images are named **ISIC_<image_id>.jpg** where image id is a 7-digit unique identifier.
Lesion images from all anatomic sites (except from the mucosa and nails) have been collected using several types of dermatoscopes from a historical sample of patients who were screened for skin cancer at multiple institutions. There is only one primary lesion visible in each lesion imaging; additional pigmented areas, smaller secondary lesions, or other fiducial indicators may be overlooked.
The distribution of disease states is representative of a modified "real world" situation in which malignancies are over-represented but benign lesions outnumber malignant lesions.

Response data
These are, instead, the binary mask images, that indicate the location of the primary skin lesion within each input lesion image.
They are named **ISIC_<image_id>_segmentation.png** where the image id matches the corresponding input data image for the mask.
Also here the codification of the pixel is as seen in 4.4.1:

- 0: that would be the background of the image or the areas outside the lesion
- 255: that would be the foreground of the images or the areas inside the lesion

Mask images should only show a single contiguous foreground region, without any gaps or detached components, assuming the primary skin lesion is a single uninterrupted region. The foreground area may border the image's edges and can be any size, even the full image.
The masks are created by an expert clinician in one of the following three methods:

1. Fully-automated algorithm, reviewed and accepted by a human expert.
2. A semi-automated flood-fill algorithm, with parameters chosen by a human expert.
3. Manual polygon tracing by a human expert.

Evaluation and selection of the images
The response are scored using a threshold Jaccard index metric.

- For each image, a pixel-wise comparison of each predicted segmentation with the corresponding ground truth segmentation is made using the Jaccard index.
- The final score for each image is computed as a threshold of the Jaccard according to the following:
  - $score = 0$, if the Jaccard index is less than 0.65
  - $score =$ the Jaccard index otherwise
- The mean of all per-image scores is taken as the final metric value for the entire dataset

In this work there are three sets of data utilized that would be *training set*, *validation set*, and *test set*, composed as follows:

- Training set: 2594 images (and 2594 labels)
- Validation set: 100 images (and 100 labels)
- Test set: 1000 images (and 1000 labels).

A subset of 100 images was used to measure the lowest Jaccard agreement between three independent expert annotators in order to establish the threshold. The 0.65 value threshold (with added error tolerance) that denotes segmentation failure on an image is based on this empirically determined value (near 0.74).

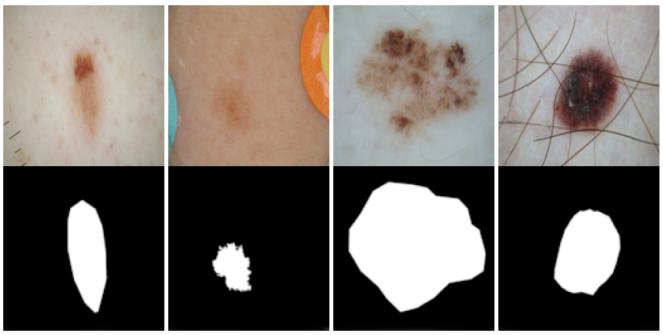


Figure 4.6: Some samples of the input data (dermoscopy image) and their response data (ground truth) from the dataset. [23]

4.4.3 COVID-QU-Ex Dataset

Compiled by the Qatar University, it is the first dataset that utilizes both lung and infection segmentation to detect, localize and quantify COVID-19 infection from X-ray images. Therefore, it can assist the medical doctors to better diagnose the severity of COVID-19 pneumonia and follow up the progression of the disease easily.

This dataset will be used to train a model on the basis of the CoTrFuse architecture, since given the size of the dataset, for the experiments that will be conducted with DMTNet, the intention is to understand how well a model trained on this type of X-ray image can then be reused and how well it is able to generate.

Input data
The input data are chest X-ray images that are divided as follows:

- 11,956 COVID-19
- 11,263 Non-COVID Infections (Viral or Bacterial Pneumonia)
- 10,701 Normal

Response data
Ground-truth lung segmentation masks are provided for the entire dataset.

Evaluation and selection of the images
The experiments were conducted on two CXR sets, where each set is divided into train, validation and test sets:

1. *Lung Segmentation Data*: entire COVID-QU-Ex dataset (33,920 CXR images with corresponding ground-truth lung masks), composed by training set (21715 images), validation set (5417 images), test set (6788 images).

2. *COVID-19 Infection Segmentation Data*: a subset of COVID-QU-Ex dataset (1,456 Normal and 1,457 Non-COVID-19 CXRs with corresponding lung mask, plus 2,913 COVID-19 CXRs with corresponding lung mask from COVID-QU-Ex dataset and corresponding infections masks from QaTaCov19 dataset). It is composed by training set (3728 images), validation set (932 images), test set (1166 images).

This dataset, in details, is the dataset chose to train the model where *few-shot learning* (3.1) will be applied. The reason is that it is the biggest dataset used in this work, so this might be an optimal point to start.

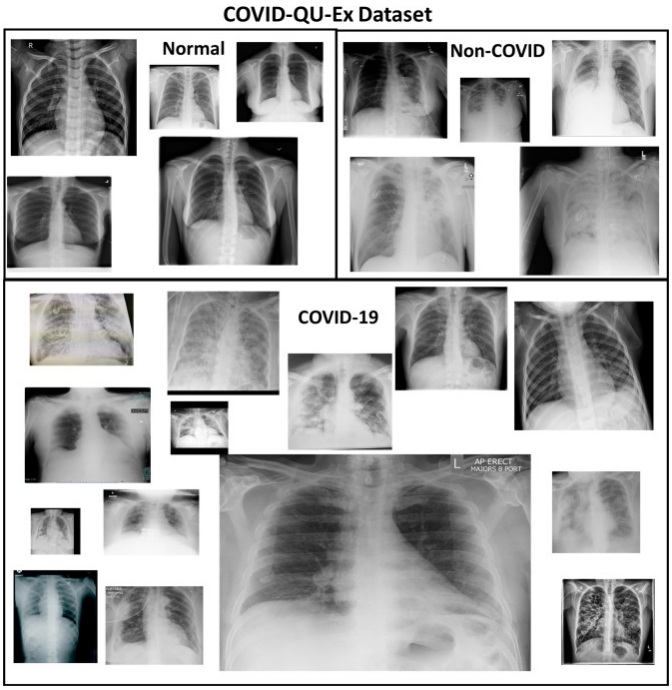


Figure 4.7: Some samples of the whole COVID-QU-Ex dataset. Credits to [24].

4.4.4 PASCAL Visual Object Classes

A well-known benchmark for computer vision object identification, segmentation, and classification is the PASCAL VOC (Visual Object Classes) [25] dataset. It has extensive annotations for 20 distinct object categories, including segmentation masks, bounding boxes, and class labels. Because it uses standardized assessment criteria like mean Average Precision (mAP), researchers frequently use it to assess the performance of models like Faster R-CNN, YOLO, and Mask R-CNN.

This dataset was used by the original authors, for the reasons listed in **??**, and will

be the subject of comparisons with experiments conducted with models previously trained with CoTrFuse.

Input data
The collection includes 7282 photos with 19694 identified items from 21 distinct categories, such as neutral, person, chair, and other. These classifications include automobiles, motorcycles, buses, boats, horses, bicycles, birds, bottles, sofas, airplanes, potted plants, trains, dining tables, and cars.

Pixel-level instance segmentation annotations are present in the images within the PASCAL VOC 2012 dataset. Owing to the nature of the job, instance segmentation can automatically be converted into object detection (bounding boxes for each item) or semantic segmentation (one mask for each class). Of the total number of photos, 1456 (20%) are unlabeled, meaning they lack annotations. Trainval (2913 pictures), train (1464 images), test (1456 images), and val (1449 images) are the four splits in the dataset. The dataset was released in 2012 by the UK joint research group.

Response data
As previously said, 20% of the data have no label, while the remaining 80% of the dataset have a label.

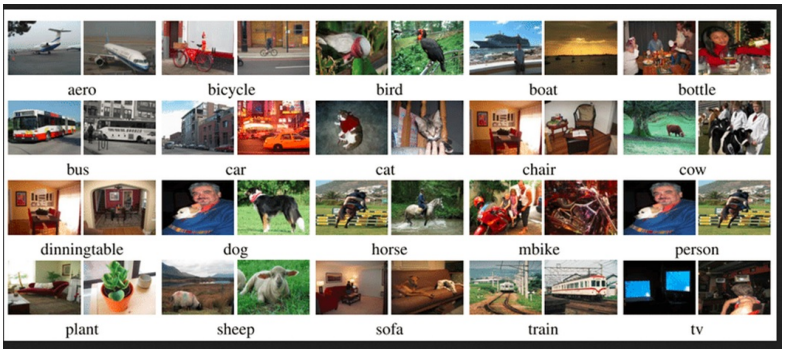


Figure 4.8: Various examples from Pascal VOC 2012.

PASCAL VOC in Medical Imaging

In many works on Few-shot Segmentation for Medical Imaging, there is the use of PASCAL VOC as a training dataset. For example in [26], the problem of data scarcity in the medical sector is explored, which is why few-shot learning techniques are used, particularly for segmentation and classification. In this case, the dataset used for training is precisely PASCAL VOC, both because of its distinct and well-defined classes, and because its benchmark is recognised in the field of

segmentation. It also offers a lot of diversity, which gives the model the possibility to adapt and be able to generalise more. It is also possible to find PASCAL VOC in [16] [20], such as those mentioned above. This indicates an excellent versatility of the dataset in question, which is why a trained architecture was chosen with this dataset.

Chapter 5

Experiments

This chapter will look in detail at the modifications made, the techniques and the relevant hyper-parameters used to conduct this work. The following subsections introduce the common changes for the three datasets. Computational resources were provided by HPC@POLITO (http://hpc.polito.it) and from Google Colab Pro.

5.1 Experiments over CNN and Transformers

Employing the COVID-QU-Ex, ISIC 2017, and ISIC 2018 datasets, respectively, three training sessions were performed out as the first experiment using CoTrFuse [19].

Pretrained model

The original work uses Swin-Transfomer model trained on ImageNet-1k introduced in [9], and then tested on ISIC2017 and COVID-QU-Ex.
In this work the same pre-trained model is used, but it is trained and tested three times with the three different datasets introduced.

This section will insight the modification and the hyper-parameters used in this first part of the presented work.

5.1.1 Experiments with COVID-QU-Ex

Data augmentation

After the dataset has been fully uploaded, data augmentation is done in accordance with the findings of the paper's original authors [19]; specifically, this means that:

- A straightforward resizing of the picture to 224x224 pixels from its original size.

- The item Rotational variance is introduced into the training data by randomly rotating the picture by either 0, 90, 180, or 270 degrees, which aids in the model's capacity to generalize to new images.

- A normalization that guarantees that the image's pixel values are normalized using the mean and standard deviation values (found in 5.1), which are commonly used for ImageNet pre-trained models like ResNet-50.

Learning Rate

A grid search was used to determine the optimal learning rate. Due to the large size of the dataset, it was only performed using a portion of the dataset and a lower number of epochs (30), in order to obtain an overall trend and choose the optimal value.
For the grid search, the values of the Learning Rates were $[10^{-1}, 10^{-3}, 10^{-4}, 10^{-5}]$. The best outcome for this case was 10^{-4} even if $10^{-}3$ was also a valid alternative, as it is possible to see in the graphs below:

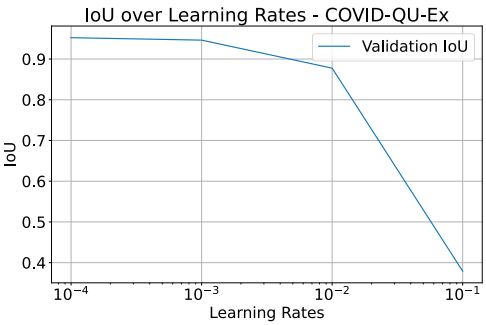


Figure 5.1: IoU with different learning rates for COVID-QU-Ex dataset.

Hyper-parameters

A table including every hyper-parameter for COVID-QU-Ex is provided below.

Hyper-parameter	Value
Batch size	8
Workers	16
Learning Rate	0.0001
Image size	224x224
Seed	2021
Epoch	50
Datasets	Infection Segment Data, Lung Segment Data
Flip probability	0.5
Mean	[0.485, 0.456, 0.406]
Standard deviation	[0.229, 0.224, 0.225]
Max Pixel Value	255.0
Probability of the transformation	1

Table 5.1: Hyper-parameters for COVID-QU-Ex.

5.1.2 Experiments with ISIC 2017 and ISIC 2018

Since the two datasets are comparable to one another and maintain similar modifications even when they are not utilized together, this section displays the modifications for both ISIC2017 and ISIC2018. When there is a difference between two optimization choices for a dataset, it is clearly indicated.

Data augmentation

Similar to the COVID-QU-Ex dataset, data augmentation is done after the images are loaded and is based largely on the original authors' work of [19], but with a few modifications:

- The image size is set to 512x512.
- To assist the model learn to detect features regardless of their orientation in the picture, rotational variance is introduced by randomly rotating the image by 0, 90, 180, or 270 degrees.
- An image's probability of being flipped horizontally in order to diversify the training set and maybe improve generalization.
- To make the model more resistant to changes in object location and size inside the image, there is a random scale up or scale down and a rotation limit.
- To assist the model manage photographs captured under varying light circumstances, such as hue, saturation, and a brightness limit, the image brightness and contrast are randomly modified.

- To resemble potential real-world noise in photos, Gaussian noise is applied to the picture.

- The picture undergoes a random alteration using either lens distortion, grid effect, or distortion.

- The pixel values are normalized using the mean and standard deviation.

All the values can be found in the table 5.2

Learning Rate

For both ISIC2017 and ISIC2018, the optimal learning rate was determined using the same grid search parameters as the COVID-QU-Ex dataset, yielding the same optimal learning rate of 0.0001. Given that the datasets were also selected for their resemblance and their application to activities that are essentially the same, if not more so, this outcome seems reasonable.

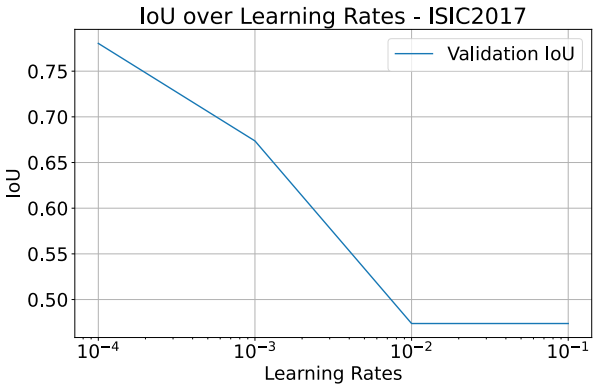


Figure 5.2: IoU with different learning rates for ISIC 2017 dataset.

Given that the two datasets are similar, the search for the learning rate for ISIC2018 may be identical to and so could select the same as for ISIC2017; however, thanks to the computational resources obtained, it was possible to conduct a test that was safe.

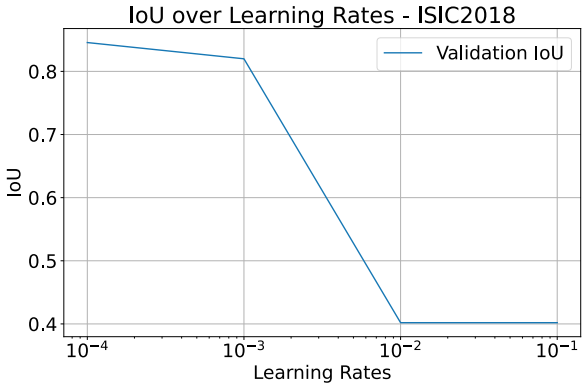


Figure 5.3: IoU with different learning rates for ISIC 2018 dataset.

Hyper-parameters

A comprehensive table detailing all the hyper-parameters used for both the ISIC 2017 and ISIC 2018 datasets is provided below, offering a clear comparison of the settings applied during experimentation. In cases where there are differences between ISIC 2017 and ISIC 2018, these distinctions are explicitly noted.

Hyper-parameter	**Value**
Batch size	8 (ISIC 2017), 16 (ISIC 2018)
Workers	16
Learning Rate	0.0001
Image size	512x512
Seed	2021
Epoch	50
Datasets	ISIC 2017, ISIC 2018
Flip probability	0.5
Mean	[0.485, 0.456, 0.406]
Standard deviation	[0.229, 0.224, 0.225]
Max Pixel Value	255.0
Probability of the transformation	1

Table 5.2: Hyper-parameters for ISIC 2017 and ISIC 2018.

5.2 Experiments over Cross-Domain and Few-shot Segmentataion

Pretrained models

A portion of the experiments were conducted using the ResNet-50 model provided by the authors of [20], which had previously been pre-trained for 8 hours on PASCAL VOC 2012 (4.4.4).

5.2.1 Experiments with ISIC2017 and ISIC2018

Data pre-processing

The first adjustment made was in the pre-processing phase, as the dataset was not correctly divided into the three defined classes: nevus, melanoma, and seborrheic keratosis. The modifications included changing how the images were processed. Instead of reading the images in batches as done in the original work, the new approach reads each image individually. Additionally, to prevent redundancy, the images are moved directly to their respective locations in the modified version, whereas in the original pre-processing, the images were copied, leading to unnecessary duplication.
Additionally, in order to prevent redundancy, the images in the modified version are moved immediately, whereas in the original pre-processing, they had to be copied.

Data Augmentation

The original data augmentation strategy is described below:

- The image is downsized from its original size to 400x400 to create consistency across all images.
- Normalize the image tensor by measuring the mean and the standard deviation for each channel (R, G, B). This preprocessing step normalizes pixel values to have a median of zero and a standard deviation of one, improving convergence and model performance.

However, to come as close as possible to the date augmentation of the original model, it was modified as follows:

- The image size is set 300x300 pixels.
- A random crop of 256x256 is applied to the previous resized image.
- A random horizontal flip is applied with a probability of 50%, this is a common data augmentation technique for tasks like segmentation.

- A random vertical flip is applied with a probability of 30%.
- The image is flipped by an angle between -15 an 15 degrees to help the model to become invariant to small rotations, which can occur due to variations in the way images are captured.
- A color jitter transformation is applied, to change the brightness, contrast, saturation and hue, all this to simulate real-world images.
- A Gaussian blur, with a probability of 40%, is applied to the image with a probability of 40%, to help the model to be more robust to noise and minor variations.
- A random probability of 30% is applied between adjusting the gamma of the images to simulate different exposure levels and adjusting the sharpness of the image.
- The image is normalized with ImageNet mean and standard deviation values.

Selection of query image and support images

Initially, the current episode's class is determined by calculating a class ID based on the element's index in the experiment. Next, a casual selection of one image — called *query image* — is made from the collection of images associated with the identified class. Once the choice is made, it is ensured that the same image is not chosen more than once in the same episode.

At this point, an iterative procedure for choosing the support images is implemented. Every iteration, a random image from the same *query image* class is selected. But first, a check is made to ensure that this image is distinct from *query image* before adding it to the collection of support images. This loop continues until the desired number of support images is not reached.

Fundamentally, *query image* is the reference image for the current episode, while *support images* are a collection of additional images belonging to the same class that aid the model in learning the discriminating characteristics of the same class. The random image selection, in conjunction with the guarantee of non-duplication between *query image* and *support images*, helps to enhance the diversity of the examples provided in the model, promoting more reliable and effective learning.

Hyper-parameters

Following, a table that shows all the hyper-parameters used for the Few Shot Segmentation.

Hyper-parameter	**Value**
Batch size	30
Workers	0
Learning Rate	1e-6
Image size	400x400
Datasets	ISIC 2017, ISIC 2018
Horizontal Flip probability	0.5
Vertical Flip probability	0.3
Random Rotation	[-15,15]
Gaussian blur probability	0.4
Size and intensity of the blur	(3,3) and (0.1, 2.5)
Gamma and sharpness probability	0.3
Gamma and sharpness values	(0.7, 1.3) and (0.8, 1.2)
Max pixel value	255.0
Mean	[0.485, 0.456, 0.406]
Standard deviation	[0.229, 0.224, 0.225]

Table 5.3: Hyper-parameters for both ISIC2017 and ISIC2018 in FSS experiments.

Please note that in the results chapter (Chapter 6) different parameters will be shown when needed, since the two architectures (CoTrFuse [19] and DMTNet [20]) differ significantly each other.

Chapter 6

Results

The results of training and testing the ISIC 2017, ISIC 2018, and COVID-QU-Ex datasets will be shown in the upcoming chapter. Although multiple experiments were conducted, only the most significant ones will be presented and discussed.

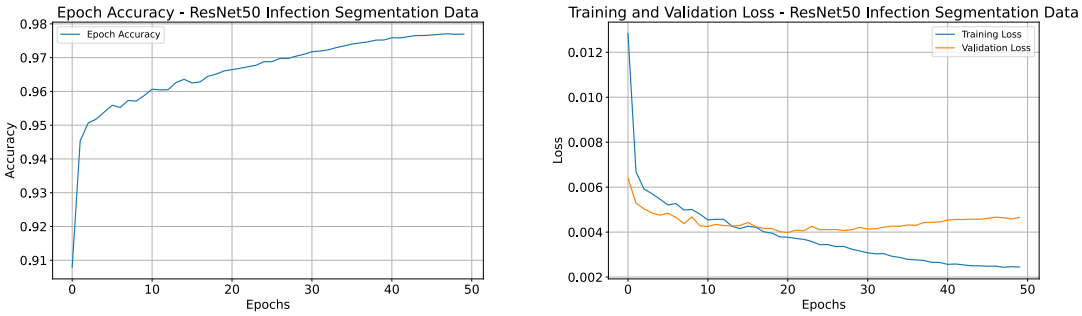
6.1 COVID-QU-Ex Training and Test results

The model that was created using the COVID-QU-Ex dataset is the first one to be shown. Few-Shot Learning methods will be applied to this model. The training is split into two sections: the first contains the dataset's Infection Segment Data, as shown in 4.4.3; the second part contains the dataset's missing Lung Infection Data.

6.1.1 Training and test with Infection Segment Data

Training

To begin the experiments with the COVID-QU-Ex dataset, which is divided in two parts, as explained in 4.4.3, it was possible to star with the part of the dataset comprising the Infection Segment Data. With 50 epochs in all, the training took place over the course of 12 hours. Training and validation loss, epoch accuracy, and validation IoU—the latter being the most important—are the formats in which the findings are shown.



(a) Accuracy. **(b)** Train and Validation Loss.

Figure 6.1: Accuracy and Losses for COVID-QU-Ex - only Infection Segment Data.

To follow the IoU Validation graph, and it can be seen that, in addition to being quite high, the values tend to stabilize around the 40th epoch, following a linear pattern with intervals when the IoU increases.

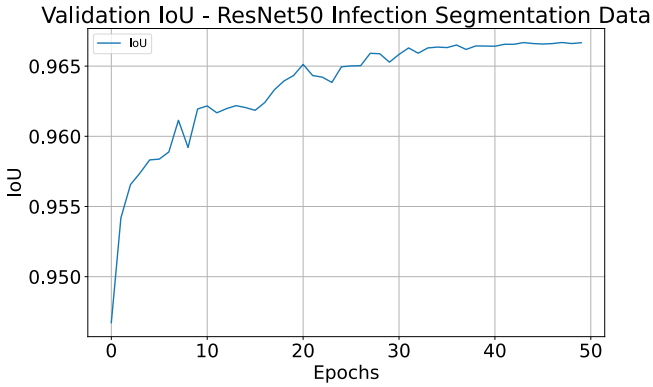


Figure 6.2: Validation IoUs for COVID-QU-Ex - only Infection Segment Data.

All the hyper-parameters and configuration can be seen in table 5.1.

Test

Regarding the test set, a test was conducted to assess the work completed thus far before proceeding with the full training of the dataset. Presented below are the test results using the specified metrics 4.3, using only the Infection Segment Data.

Method	Dice (%)	mIoU (%)	Precision (%)	Recall (%)	F1-Score (%)	Pixel Accuracy (%)
Tweaked CoTrFuse	98.7	96.3	97.37	96.84	97	98.66

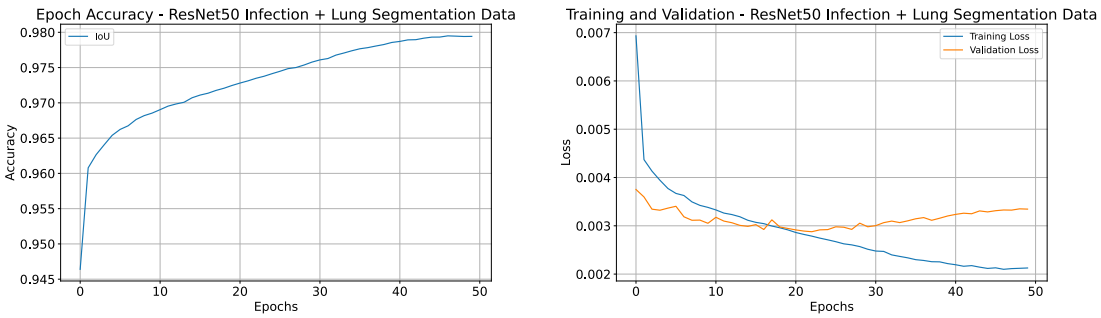
Table 6.1: Results of tests performed on the COVID-QU-Ex portion of dataset.

As a consequence of the positive results, training proceeded with the dataset's remaining portion, which contained the lung infection data.

6.1.2 Training and test with Infection Segment Data and Lung Segment Data

Training

The other dataset, the Lung Segmentation Data, could be used for training when the results were collected and it was determined that the model was operating effectively. Due to the computing resources, a one-time training was not feasible. Once more, there were fifty epochs, and the training took around eighteen hours. Since the two sets are comparable and originate from the same dataset, the hyper-parameters remained unchanged and are, thus, identical to those in the table 5.1.



(a) Accuracy. **(b)** Train and Validation Loss.

Figure 6.3: Accuracy and Losses for COVID-QU-Ex complete dataset.

Additionally, a graph showing the trend of the Validation IoU begins to fully stabilize from the 40th epoch onwards, similar to the prior scenario, suggesting that a lengthier training might further improve and stabilise the model.

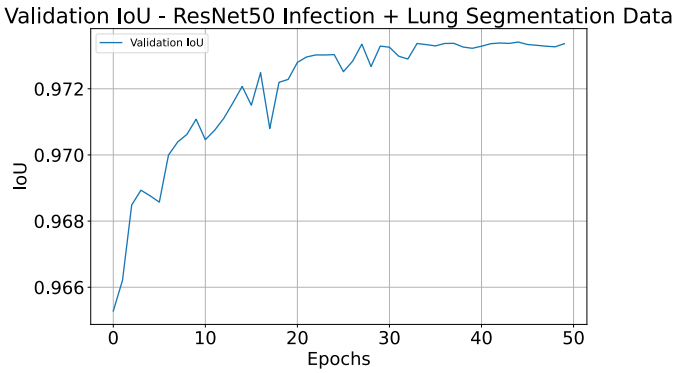


Figure 6.4: Validation IoUs for COVID-QU-Ex complete dataset.

Test

After training and validation using the whole dataset presented in section 4.4.3, testing was at last feasible, allowing for the initial comparison with the project baseline, as seen below:

Method	Dice (%)	mIoU (%)	Precision (%)	Recall (%)	F1-Score (%)	Pixel Accuracy (%)
Baseline	93.63	90.51	90.35	90.42	88.45	98.1
Tweaked CoTrFuse	**98.6**	**97.28**	**98.00**	**97.68**	**97.83**	**99.05**

Table 6.2: Results of tests performed on the whole COVID-QU-Ex dataset.

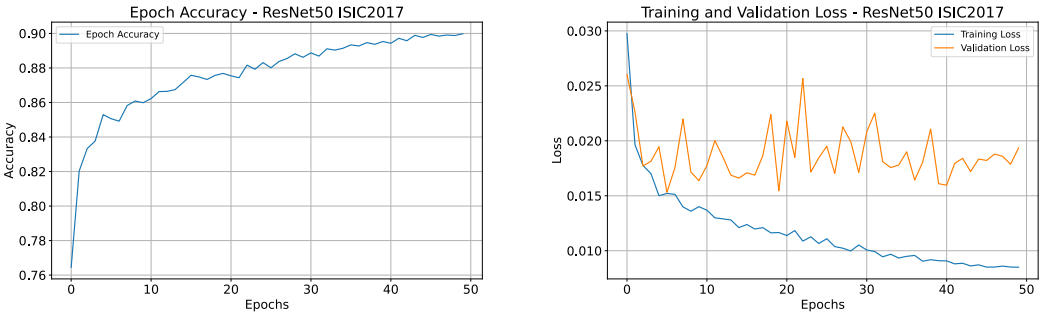
Following testing, it was discovered that CoTrFuse, a great baseline with results that are already optimum, is still optimisable with data augmentation approaches and a smaller batch size—the baseline's batch size is 16, not 8 as in this study.

6.2 ISIC2017 and ISIC2018 Training and Test results

6.2.1 Training and test with ISIC2017

Training

With respect to the ISIC2017 training, it was feasible to complete it in its entirety in one sitting. The training, which lasted roughly 16 hours and 50 epochs, was conducted on the Politecnico di Torino's Legion cluster, much like for COVID-QU-Ex. The Train Loss, Validation Loss, Epoch Accuracy, and Validation IoU values were then extracted at the conclusion of this.



(a) Accuracy. **(b)** Train and Validation Loss.

Figure 6.5: Accuracy and Losses for ISIC2017 dataset.

Epoch accuracy and train loss seem to be steady or very stable in this case. However, when it comes to validation loss, it may seem a little unstable given that we are talking about a factor that was considered but did not raise any specific concerns—roughly 0.005 to 0.01 at most. The most concerning aspect of the graph is really the latter section, where it seems as though the loss is increasing. This suggests that, as can be shown starting about epoch 42, less training may have produced a more stable loss.
However, the Validation IoU tends to trend more steadily towards the end of the training, suggesting that additional training might have further stabilised the IoU.

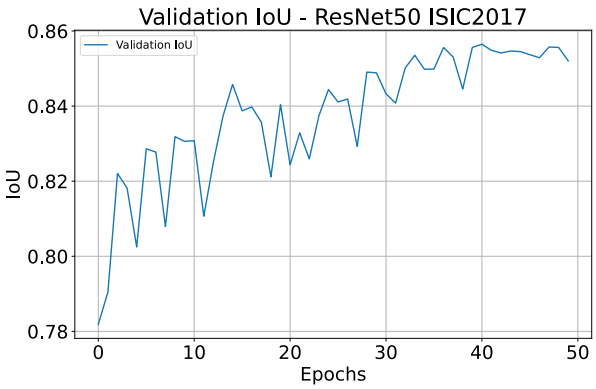


Figure 6.6: Validation IoUs for ISIC2017 dataset.

Test

Although the Intersection over Union began to stabilize towards the end of training, the test set using the newly developed model was selected. The results were comparable to the baseline, and in some cases, even better.

Method	Dice (%)	mIoU (%)	Precision (%)	Recall (%)	F1-Score (%)	Pixel Accuracy (%)
Baseline	**90.86**	**85.24**	93.08	**85.13**	**86.75**	**94.50**
Tweaked CoTrFuse	89.51	83.50	**95.45**	80.59	84.91	93.42

Table 6.3: Baseline and Tweaked versions compared.

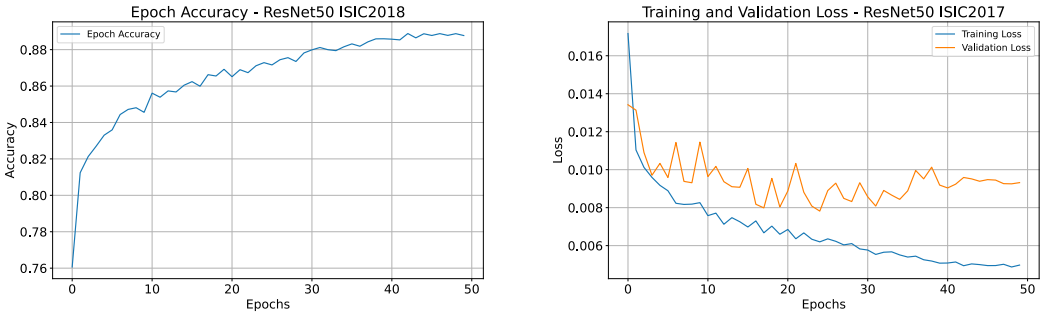
Unfortunately, in the case of ISIC, the modified version was only better in one area —precision— despite the results being consistent with the baseline. This suggests that while the adjustments were undoubtedly beneficial to the COVID-QU-Ex dataset, they were not particularly beneficial to ISIC2017. As was previously mentioned, an option would be to do a longer training.

The full list of hyper-parameters setting was reported in table 5.2.

6.2.2 Training and test with ISIC2018

Training

The next stage was to apply ISIC2018 rather than ISIC2017 using the pre-trained model. In this case, a total of 50 epochs and 24 hours of training were needed. The hyper-parameters remained unchanged because of how comparable the data were. Despite certain instabilities in the validation loss, as can be observed, they are less than in ISIC2017 in terms of both quantity (measured as the number of ups and downs) and quality (measured as the accuracy difference across epochs, which is between 0.002 and 0.003).
Additionally, during the epochs after the 40th, the validation loss starts to stabilize, but with an increase of around 0.002, indicating that additional gains may have been obtained from a longer training period.
Conversely, in terms of precision, its increase is linear and rather constant.



(a) Accuracy. **(b)** Train and Validation Loss.

Figure 6.7: Accuracy and Losses for ISIC2018 dataset.

For the Validation IoU, a separate case must undoubtedly be made because it exhibits instability up until around epoch 40, at which point it starts to rise steadily, suggesting the likely need for more training.

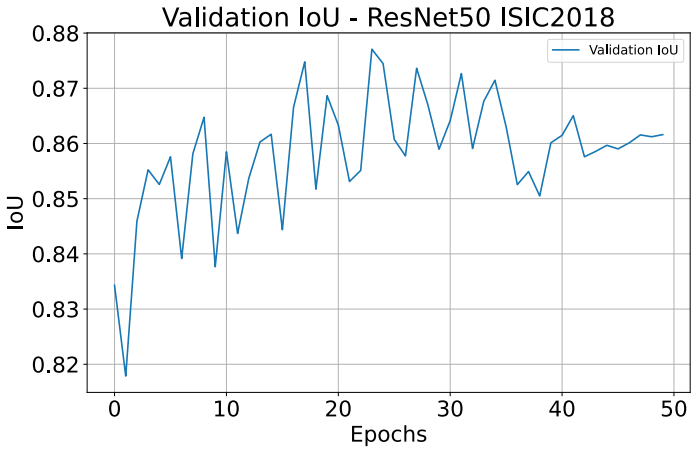


Figure 6.8: Validation IoUs for ISIC2018 dataset.

Test

Testing the model trained using ISIC2018 reveals values that are significantly higher than those in ISIC2017, frequently surpassing 90% and consistently above 84%. This is a good improvement over ISIC2017.
Again, as it was never evaluated with ISIC2018, a direct comparison with the baseline is not feasible. Nonetheless, because the two datasets are comparable, a quick comparison with the ISIC2017 baseline may be established.

Method	Dataset	Dice (%)	mIoU (%)	Precision (%)	Recall (%)	F1-Score (%)	Pixel Accuracy (%)
Tweaked CoTrFuse	ISIC2017	89.51	83.50	**95.45**	80.59	84.91	**93.42**
Tweaked CoTrFuse	ISIC2018	**91.09**	**85.77**	87.41	**94.12**	**89.40**	93.91

Table 6.4: Comparison between ISIC 2017 and ISIC 2018 tests.

Once again it is possible to see the whole hyper-parameters list in table 5.2.

6.3 Few-shot Segmentation

This section will present the outcomes of experiments conducted using two models:

- ResNet-50 pre-trained with PASCAL VOL (4.4.4), provided from the authors of [20] and based on DMTNet.

- ResNet-50 trained with the whole COVID-QU-Ex dataset, based on CoTrFuse (4.4.3, 6.1.2).
- ResNet-50 trained with the whole ISIC2017 dataset, based on CoTrFuse (4.4.1, 6.2.1).
- ResNet-50 trained with the whole ISIC2018 dataset, based on CoTrFuse (4.4.2, 6.2.2).

ISIC2017 and ISIC2018 are the two datasets that were used for the testing, and there will not be a training part. The results that are displayed try to clarify the circumstances in which Few-shot learning can influence such a process.

6.3.1 Tests with original model

Since it was appropriate to confirm the results even on a setting that differed significantly from the authors', the first two tests were conducted using the original model and without modifying the parameters. Nonetheless, the outcomes validated the findings of the writers, which is why moving on was feasible.

Methods	**Backbone**	**Dataset**	**1 shot**	**5 shot**
Original setup	ResNet-50 (pretrained)	ISIC2018	43.55	52.30
Alternative setup	ResNet-50 (pretrained)	ISIC2018	**43.68**	**52.33**

Table 6.5: Results showing 1 shot on the two different versions. The value of mIoU represents the average.

6.3.2 Tests with COVID-QU-Ex model on Few Shot Segmentation

This section presents all of the test results that were collected utilizing the previously trained model in the CoTrFuse architecture, executed on the DMTNet architecture.
Because the model trained on the COVID-QU-Ex datasets lacks a precise class separation, testing it using Few Shot Segmentation techniques was only possible with the ISIC2017 and ISIC2018 datasets.

After training the model with CoTrFuse 4.1, and testing with the named architecture, the decision was made to test the model on an architecture that could support Few-shot segmentation to see how it behaved with few data provided. Also in this case, the data augmentation applied to the dataset is the one reported in

table 5.3.
Tests were conducted in two ways:

- 1 shot (1 sample per class)
- 5 shot (5 samples per class)

The first tests were conducted without any changes to the reported hyper-parameters in 5.3. Although it is not possible to make a direct comparison, it can be seen that there is an improvement from 1 shot to 5 shots, indicating that the model performs better with more data.

Model	**Backbone**	**Dataset**	**Metric (%)**	**1-shot**	**5-shot**
COVID-QU-Ex	ResNet-50	ISIC2017	mIoU	25.51	27.76
COVID-QU-Ex	ResNet-50	ISIC2018	mIoU	22.91	26.51

Table 6.6: Results for mIoU 1-shot and 5-shot with Learning Rate 0.000001.

The mIoU turns out to be slightly lower than the models trained with the proposed architecture and a similar dataset (Chest X-Ray), showing that the model trained with CoTrFuse, with more data, could reach - if not exceed - the results originally reported in the paper 4.2.
Another metric measured, but which does not appear in the paper, is the FBIoU 4.3.2 and shown below:

Model	**Backbone**	**Dataset**	**Metric (%)**	**1-shot**	**5-shot**
COVID-QU-Ex	ResNet-50	ISIC2017	FB-IoU	12.97	45.10
COVID-QU-Ex	ResNet-50	ISIC2018	FB-IoU	12.88	42.83

Table 6.7: Results for FBIoU with 1-shot and 5-shot with Learning Rate 0.000001.

The results show that there is a moderate overlap between the model's prediction and the correct mask: the model is segmenting the object or the background, in which case 45% and 42% are interpreted as optimal values as the data presented is extremely small and the model was drawn in a very different architecture. Again, the presence of more shots per class would increase the FBIoU.

After these tests, it was possible to conduce further tests taking into consideration the following paper [18]. This in fact indicate new tuples of values for mean and standard deviation, despite the fact that they are not works focusing on segmentation.
The values in question were tested by the original authors exclusively on ISIC2017,

which is why they were not also carried out completely on the ISIC2018 dataset - despite the similarity - but only in part.

Model	Backbone	Dataset	Metric (%)	Mean	Standard deviation	1-shot	5-shot
COVID-QU-Ex	ResNet-50	ISIC2017	mIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	25.74	30.02
COVID-QU-Ex	ResNet-50	ISIC2018	mIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	21.94	27.23

Table 6.8: Results mIoU with different mean and standard deviation.

Meanwhile the following results shows the FBIoU:

Model	Backbone	Dataset	Metric (%)	Mean	Standard deviation	1-shot	5-shot
COVID-QU-Ex	ResNet-50	ISIC2017	FBIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	36.56	41.10
COVID-QU-Ex	ResNet-50	ISIC2018	FBIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	20.64	22.74

Table 6.9: Results FBIoU with different mean and standard deviation.

It is possible to see a clear improvement in the results of ISIC2017, but a deterioration in the data of ISIC2018. This is due to the fact that the normalisation values are designed specifically for ISIC2017, but nevertheless given the similarity between datasets, it was decided to apply these tests to ISIC2018 as well in order to understand the trend of the results.
The results show that the FBIoU improves noticeably, especially in the 1-shot; while in the 5-shot there is a slight deterioration but remains in line with the values obtained.

Given the improvement especially in the 5-shot, it was decided to perform new tests using different means and standard deviations, which are summarised in two separate sections.

ISIC2017

Below it is possible to see new values for mean and standard deviation, one taken completely from the reference paper [18] while the other has been mixed with values typically used for pre-trained models with ImageNet.

Model	Backbone	Metric (%)	Mean	Standard deviation	1-shot	5-shot
COVID-QU-Ex	ResNet-50	mIoU	[0.763, 0.545, 0.570]	[0.140, 0.152, 0.169]	**25.51**	**30.05**
COVID-QU-Ex	ResNet-50	mIoU	[0.684, 0.483, 0.519]	[0.185, 0.186, 0.199]	24.06	26.66

Table 6.10: ISIC2017 mIoUs with different means and standard deviations.

Model	Backbone	Metric (%)	Mean	Standard deviation	1-shot	5-shot
COVID-QU-Ex	ResNet-50	FBIoU	[0.763, 0.545, 0.570]	[0.140, 0.152, 0.169]	23.74	41.64
COVID-QU-Ex	ResNet-50	FBIoU	[0.684, 0.483, 0.519]	[0.185, 0.186, 0.199]	15.11	16.89

Table 6.11: ISIC2017 FBIoUs with different means and standard deviations.

It is possible to see a big improvement from 1 shot to 5 shots for ISIC2017 in both mIoU and FBIoU while using [0.763, 0.545, 0.570], [0.140, 0.152, 0.169], indicating that different mean and standard deviation values can lead to very good results. And yet, although the mIoU remains optimal, the same cannot be said for the FBIoU of the other value pair used.

ISIC2018

After viewing the results of ISIC2017, it is possible to move on to the results of ISIC2018.
It can be seen that the values of the mean and standard deviations are different. This is due to having carried out tests - which turned out to be inconclusive - with the values also used previously. Consequently, it was decided to adapt the ImageNet mean and standard deviation values with those used for ISIC2017, considering the similarity between the crucial datasets in this case.

Model	Backbone	Metric (%)	Mean	Standard deviation	1-shot	5-shot
COVID-QU-Ex	ResNet-50	mIoU	[0.763, 0.545, 0.570]	[0.229, 0.224, 0.225]	18.55	25.65
COVID-QU-Ex	ResNet-50	mIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	**21.94**	**27.23**

Table 6.12: ISIC2018 mIoUs with different means and standard deviations.

Model	Backbone	Metric (%)	Mean	Standard deviation	1-shot	5-shot
COVID-QU-Ex	ResNet-50	FBIoU	[0.763, 0.545, 0.570]	[0.229, 0.224, 0.225]	20.53	25.44
COVID-QU-Ex	ResNet-50	FBIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	36.27	37.70

Table 6.13: ISIC2018 FBIoUs with different means and standard deviations.

In this case, the mIoU and FBIoU in the case of [0.684, 0.483, 0.519], [0.229, 0.224, 0.225] are found to respond positively, reporting results that are not only average and higher than some of the results seen above, but also increase linearly and do not deteriorate in the transition from 1 shot to 5 shots. The results of the other value pair also remain in linear growth, but nevertheless lower than the other pair.

6.3.3 Tests with ISIC2017 and ISIC2018 models on Few Shot Segmentation

This section presents all of the test results that were collected utilizing the previously trained model in the CoTrFuse architecture with both ISIC2017 and ISIC2018 dataset, executed on the DMTNet architecture.
It is important to note that the ISIC2017 model is only used with the ISIC2017 dataset, just as the ISIC2018 model is only used with the ISIC2018 dataset.

ISIC2017 model

The results of ISIC2017 using a different model that was developed on this exact dataset are displayed in this section. The table 5.2 displays specifics about the instruction.

Model	Backbone	Dataset	Metric (%)	1-shot	5-shot
ISIC2017	ResNet-50	ISIC2017	mIoU	28.00	22.68
ISIC2017	ResNet-50	ISIC2017	FBIoU	38.83	43.31

Table 6.14: Results for ISIC2017 dataset with ISIC2017 model using mean and standard deviation ImageNet values.

The FBIoU clearly improved from 1 shot to 5 images, suggesting that the model is capable of distinguishing foreground and background at its best. Despite this, the mIoU degrades from 1 shot to 5 shots, which is why more testing was chosen to get a better outcome.

Due to the quantity of studies, tests were conducted to show how the mean and standard deviation varied. As a result, two distinct tables—one for mIoU and one for FBIoU—will be displayed.

Model	Backbone	Dataset	Metric (%)	Mean	Standard deviation	1-shot	5-shot
ISIC2017	ResNet-50	ISIC2017	mIoU	[0.763, 0.545, 0.570]	[0.140, 0.152, 0.169]	25.51	26.00
ISIC2017	ResNet-50	ISIC2017	mIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	**26.01**	**30.05**
ISIC2017	ResNet-50	ISIC2017	mIoU	[0.684, 0.483, 0.519]	[0.185, 0.186, 0.199]	25.51	29.00

Table 6.15: mIoU results for ISIC2017 dataset with ISIC2017 model using different mean and standard deviation values.

As can be seen, although the 1-shot is inferior to the model with the mean and standard values from ImageNet, it can be seen that they are actually better in terms of growth, as it is linear from 1-shot to 5-shot, with the 5-shot outperforming, in two cases, the previous 1-shot result visible here 6.14.

Model	Backbone	Dataset	Metric (%)	Mean	Standard deviation	1-shot	5-shot
ISIC2017	ResNet-50	ISIC2017	FBIoU	[0.763, 0.545, 0.570]	[0.140, 0.152, 0.169]	12.76	13.20
ISIC2017	ResNet-50	ISIC2017	FBIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	15.44	15.60
ISIC2017	ResNet-50	ISIC2017	FBIoU	[0.684, 0.483, 0.519]	[0.185, 0.186, 0.199]	43.31	27.00

Table 6.16: FBIoU results for ISIC2017 dataset with ISIC2017 model using different mean and standard deviation values.

In terms of the FBIoU, it is a different scenario. Actually, though to a lesser extent, the first two models expand linearly. In contrast, the third model's 5-shot decreases significantly even when the 1-shot produces a very good outcome. The model thus reveals a possible problem in discriminating foreground and background, which may be driven by the class imbalance of the dataset.

ISIC2018 model

This section will display the model's training outcomes using ISIC2018 and CoTr-Fuse, similar to ISIC2017. The training specifics are still displayed in the same table as in ISIC2017: table 5.2.

Model	Backbone	Dataset	Metric (%)	1-shot	5-shot
ISIC2018	ResNet-50	ISIC2018	mIoU	25.41	21.74
ISIC2018	ResNet-50	ISIC2018	FBIoU	16.71	26.65

Table 6.17: Results for ISIC2018 dataset with ISIC2018 model using ImageNet mean and standard deviation values.

The table above displays the findings produced using ImageNet mean and standard deviation values. The FBIoU findings are rising and on par. Nevertheless, the transition from one shot to five shots is less even with the ideal FBIoU values.

This time, the mIoU tends to increase linearly and has respectable values in both scenarios. On the other hand, the performance of the former type of adjustment applied to the mean and standard deviation is clearly superior to that of the latter.

Model	Backbone	Dataset	Metric (%)	Mean	Standard deviation	1-shot	5-shot
ISIC2018	ResNet-50	ISIC2018	mIoU	[0.763, 0.545, 0.570]	[0.140, 0.152, 0.169]	**25.18**	**26.46**
ISIC2018	ResNet-50	ISIC2018	mIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	24.57	25.79

Table 6.18: mIoU results for ISIC2018 dataset with ISIC2018 model using different mean and standard deviation values.

Once more, the first case shows to be better than the later. However, even if the FBIoU case grows linearly, it is not better than the model using the ImageNet base values.

Model	Backbone	Dataset	Metric (%)	Mean	Standard deviation	1-shot	5-shot
ISIC2018	ResNet-50	ISIC2018	FBIoU	[0.763, 0.545, 0.570]	[0.140, 0.152, 0.169]	20.64	22.74
ISIC2018	ResNet-50	ISIC2018	FBIoU	[0.684, 0.483, 0.519]	[0.229, 0.224, 0.225]	18.06	12.52

Table 6.19: FBIoU results for ISIC2018 dataset with ISIC2018 model using different mean and standard deviation values.

Reiterating that there are not three changes to the mean and standard deviation values in ISIC2018 as there were in ISIC2017, it is important to note that the third modification, with values [0.684, 0.483, 0.519] and [0.185, 0.186, 0.199], produced inconsistent results, which is why it was excluded.

6.3.4 Comparison with baseline DMTNet

These experiments aim to investigate the behaviour of a pre-trained model, whether or not medical images are used, on an architecture different from those used for few-shot segmentation, in order to determine whether or not existing models can be updated and applied to real-world scenarios. As a result, a straight comparison with the ISIC2018 dataset's baseline is possible. The data used come from the previously discussed paper [20] as well as the best model that was taken into consideration in earlier tests.

For a comprehensive comparison, the outcomes of a portion of the work in the original paper will also be presented in addition to the baseline. Furthermore, since the FBIoU of all other models is absent, just the mIoU will be reported. Nonetheless, the outcomes of the FBIoU are seen in the phases that came before.

Since ISIC2018 was evaluated by the original authors, a direct comparison of the models may be done with it; nevertheless, the ISIC2017 dataset will also be included for completeness and similarity. The models present instead will be notes those based on CoTrFuse and trained with COVID-QU-Ex, ISIC2017 and ISIC2018. ResNet-50 serves as the backbon for all models, including the original pre-trained model, the three models previously discussed, and every other model in the table.

Method	Dataset	1-shot	5-shot
Ft-last-1 [Deeplab]	ISIC2018	11.08	16.57
Ft-last-2 [Deeplab]	ISIC2018	10.22	17.56
Linear [Deeplab]	ISIC2018	19.42	30.04
PGNet [Zhang et al. 2019a]	ISIC2018	21.86	21.25
RPMMs [Yang et al. 2020]	ISIC2018	18.02	20.04
PATNet [Lei et al. 2022]	ISIC2018	41.16	**53.58**
DMTNet	ISIC2018	**43.55**	52.30
COVID-QU-Ex model*	ISIC2018	21.94	27.23
ISIC2018 model*	ISIC2018	25.18	26.46
COVID-QU-Ex model*	*ISIC2017*	25.74	30.02
*ISIC2017 model**	*ISIC2017*	26.01	30.05

Table 6.20: Results for ISIC2018 (and ISIC2017) dataset with all the models.

It is evident that the models based on PATNet, specifically DMTNet and PATNet, outperform all other models, even the ones indicated with an asterisk (i.e., those based on CoTrFuse).
The models that have been highlighted are, nonetheless, fully averaged and, in many instances, outperform models trained in architectures appropriate for Few-shot Segmentation. It follows that even if a model has not been trained in an architecture designed for that purpose, it can nevertheless assist and support Few-shot Segmentation approaches if it has been trained especially on medical imaging, and in this case, in an architecture containing Vision Transformers.

6.4 Analysis of results

In the previous sections, it is possible to see an improvement in performance with the use of CoTrFuse with the three datasets used. However, the same cannot be said of the results with the use of DMTNet and Few-shot Segmentation. Although the results are average with the other models, they do not exceed the state of the art.

In order to investigate why this is not the case, experiments were conducted on the datasets themselves to first of all understand the usefulness of cross-domain but particularly to understand how far apart the two datasets, i.e. PASCAL VOC and ISIC, were. An experiment was then conducted to measure the distances between the two datasets, using a non-symmetrical measure of the difference between the two distributions, which is the Kullback-Leibler Divergence (KL-Divergence),

defined below.

$$D_{KL}(P||Q) = \sum_i P(i) \log_2(\frac{P(i)}{Q(i)}) \tag{6.1}$$

Where P denotes the 'true' distribution of the data, while Q represents the approximation of P.

The KL-Divergence showed a marked difference between the two datasets, shown in the graph below after applying a dimensionality reduction algorithm, namely t-distributed stochastic neighbour embedding (t-SNE). The experiment was performed using the ISIC2018 and PASCAL VOC datasets, i.e. the two also used in the original work of the DMTNet authors.

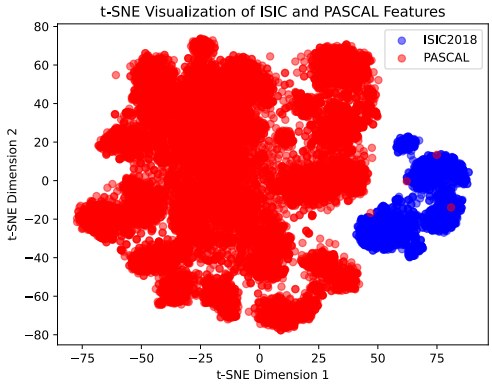


Figure 6.9: Difference between PASCAL VOC and ISIC2018 features.

It can be seen that the features are almost in all cases markedly different, thus demonstrating that the more distant the datasets used during training and few-shot are, the more performance decreases.

As counter-evidence, an experiment was also carried out by analysing the two datasets of ISIC, that are ISIC2017 and ISIC2018, which showed a higher similarity than the one shown earlier, thus justifying the results appearing similar to each other.

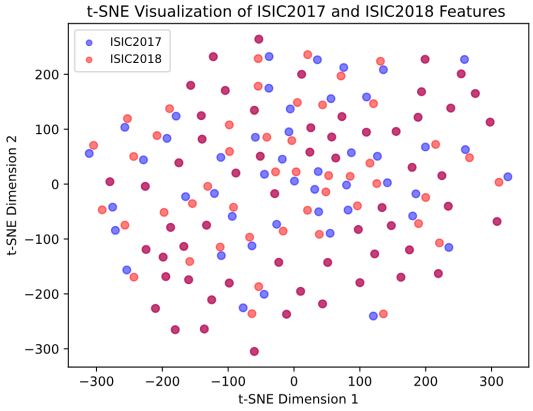


Figure 6.10: Difference between ISIC2017 and ISIC2018 features.

Chapter 7

Conclusions and future work

The primary goal of this thesis was to use Vision Transformers, one of the most promising developments in medical imaging, to address the critical problem of early skin cancer detection in the field of medical segmentation. The availability of labeled medical data is a major impediment that frequently arises in the creation of real-world artificial intelligence systems. Reason why, the second goal was to show how novel meta-learning techniques, such few-shot learning, can be beneficial in many situations.

The architecture utilized in the first part of this study was called CoTrFuse, which combines vision transformers and convolutional neural networks for image segmentation. CoTrFuse demonstrated its efficacy in combining multi-scale and multi-modal data to improve the Vision Transformers' segmentation performance. With all three datasets utilized, the adjustments made to the baseline produced satisfactory results that nearly always outperformed. Specifically, the outcomes on the ISIC2018 dataset are quite good; comparisons were possible, and all measures were well over 95%, with the exception of one instance when the pixel accuracy was marginally over 99%. Although the COVID-QU-Ex dataset yielded ideal findings as well, a comparison with the baseline could not be made. However, with ISIC2017, the results are very close to the baseline and, in one instance, even better by roughly 2%, never dropping below the 80% threshold.

The second part of this study is focused on Few-shot Segmentation architecture and technique, particularly the architecture took in consideration is DMTNet, that focuses on Cross Domain Few-shot Segmentation. Regretfully, the models trained on CoTrFuse, reached among 30% and 32% — while the baseline was between 45% and 53%. To understand why these results were obtained, a KL-divergence study was then carried out to measure the distance of the features of the datasets to see when their choice in the training phase could have an influence, and it was

shown that the PASCAL VOC dataset, used for training the DMTNet model, and ISIC2018, taken as a sample, were very different. Therefore, the choice of dataset and training methods used is also crucial. The models' restricted capacity to generalise across fewer shots may be the cause of the inferior outcomes when compared to the baseline. Despite these obstacles; such as the class division and the two different architectures, the outcomes frequently outperform those of alternative methods in the field, demonstrating the efficacy of the technique used. It is evident that the models might perform even better and possibly outperform the baseline in more consistent ways with improvements to the shot-per-class distribution and training method. The results show that further development is necessary, but they also validate the approach's promise for segmentation tasks involving other datasets.

It is therefore believed to have been demonstrated that models that are already in use and do not require training in Few-shot architecture can still be helpful in the described technology by simply updating the parameters or using data augmentation similar to that seen in this thesis or, even better, applying Cross Domain techniques. Additionally, it is possible to use more recent versions of ResNet, which are not used here due to compatibility, and which may therefore produce results that are more stable and lightweight. Therefore, this thesis is seen as a great basis for future works and a more powerful implementation.

Bibliography

[1] MelanomaUK. «2020 MELANOMA SKIN CANCER REPORT». In: (2020) (cit. on p. 1).

[2] Wikipedia. «Neural network (machine learning)». In: (2003) (cit. on p. 6).

[3] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: `1511.08458 [cs.NE]` (cit. on p. 7).

[4] IBM. «What are convolutional neural networks?» In: (2024) (cit. on p. 7).

[5] Muhamad Yani, S Irawan, and Casi Setianingsih. «Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail». In: *Journal of Physics: Conference Series* 1201 (May 2019), p. 012052 (cit. on p. 8).

[6] Alexey Dosovitskiy et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». In: *International Conference on Learning Representations*. 2021 (cit. on pp. 9, 10).

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. «Attention is All You Need». In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964 (cit. on pp. 9, 11).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186 (cit. on p. 9).

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: 2103.14030 [cs.CV] (cit. on pp. 10, 38).

[10] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. *Show and Tell: A Neural Image Caption Generator*. 2015. arXiv: 1411.4555 [cs.CV] (cit. on p. 12).

[11] Busra Emek Soylu, Mehmet Serdar Guzel, Gazi Erkan Bostanci, Fatih Ekinci, Tunc Asuroglu, and Koray Acici. «Deep-Learning-Based Approaches for Semantic Segmentation of Natural Scene Images: A Review». In: *Electronics* 12.12 (2023). ISSN: 2079-9292 (cit. on p. 13).

[12] Xiaoyan Jiang, Zuojin Hu, Shuihua Wang, and Yudong Zhang. «Deep Learning for Medical Image-Based Cancer Diagnosis». In: *Cancers* 15.14 (2023). ISSN: 2072-6694 (cit. on p. 13).

[13] Fei-Fei Li, Justin Johnson, and Serena Young. «Lecture 11: Detection and Segmentation». In: California, USA: Stanford University, 2011 (cit. on p. 13).

[14] Karthik Desingu, Mirunalini P., and Aravindan Chandrabose. *Few-Shot Classification of Skin Lesions from Dermoscopic Images by Meta-Learning Representative Embeddings*. 2022. arXiv: 2210.16954 [cs.CV] (cit. on p. 18).

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. «Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks». In: *CoRR* abs/1703.03400 (2017). arXiv: 1703.03400 (cit. on p. 18).

[16] Fabio Cermelli, Massimiliano Mancini, Yongqin Xian, Zeynep Akata, and Barbara Caputo. *Prototype-based Incremental Few-Shot Semantic Segmentation*. 2021. arXiv: 2012.01415 [cs.CV] (cit. on pp. 18, 37).

[17] Abdur R Feyjie, Reza Azad, Marco Pedersoli, Claude Kauffman, Ismail Ben Ayed, and Jose Dolz. *Semi-supervised few-shot learning for medical image segmentation*. 2020. arXiv: 2003.08462 [cs.CV] (cit. on p. 19).

[18] Yuexiang Li and Linlin Shen. «Skin Lesion Analysis towards Melanoma Detection Using Deep Learning Network». In: *Sensors* 18.2 (2018). ISSN: 1424-8220 (cit. on pp. 19, 54, 55).

[19] Yuanbin Chen, Tao Wang, Hui Tang, Longxuan Zhao, Xinlin Zhang, Tao Tan, Qinquan Gao, Min Du, and Tong Tong. «CoTrFuse: A novel framework by fusing CNN and transformer for medical image segmentation». English. In: *Physics in Medicine and Biology* 68.17 (Sept. 2023). Publisher Copyright: © 2023 The Author(s). Published on behalf of Institute of Physics and Engineering in Medicine by IOP Publishing Ltd. ISSN: 0031-9155 (cit. on pp. 20–22, 38, 40, 45).

[20] Jiayi Chen, Rong Quan, and Jie Qin. «Cross-Domain Few-Shot Semantic Segmentation via Doubly Matching Transformation». In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. Ed. by Kate Larson. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2024, pp. 641–649 (cit. on pp. 20, 23, 25, 37, 43, 45, 52, 59).

[21] Yundong Zhang, Huiye Liu, and Qiang Hu. *TransFuse: Fusing Transformers and CNNs for Medical Image Segmentation*. 2021. arXiv: 2102.08005 [cs.CV] (cit. on p. 21).

[22] «Metrics for semantic segmentation». In: (May 2019) (cit. on p. 27).

[23] Guanzhong Zhang and Shengsheng Wang. «Dense and shuffle attention U-Net for automatic skin lesion segmentation». In: *International Journal of Imaging Systems and Technology* 32 (June 2022) (cit. on pp. 32, 34).

[24] Anas M. Tahir et al. «COVID-19 infection localization and severity grading from chest X-ray images». In: *Computers in Biology and Medicine* 139 (2021), p. 105002. ISSN: 0010-4825 (cit. on p. 35).

[25] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (cit. on p. 35).

[26] Jannatul Nayem, Sayed Sahriar Hasan, Noshin Amina, Bristy Das, Md Shahin Ali, Md Manjurul Ahsan, and Shivakumar Raman. *Few Shot Learning for Medical Imaging: A Comparative Analysis of Methodologies and Formal Mathematical Framework*. 2023. arXiv: 2305.04401 [eess.IV] (cit. on p. 36).