FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

CNN-Based Refinement for Image Segmentation

José Soares Rebelo

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Jaime dos Santos Cardoso, PhD

Second Supervisor: Kelwin Fernandes

July 9, 2018

CNN-Based Refinement for Image Segmentation

José Soares Rebelo

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Alexandre Valle de Carvalho, PhD
External Examiner: Adrian Galdran, PhD
Supervisor: Jaime dos Santos Cardoso, PhD

July 9, 2018

Abstract

Image segmentation is the area of computer vision that tries to partition an image into multiple parts, according to some semantic meaning. This is one of the core computer vision problems, and crucial to many applications. Medical imaging is one of such applications that can benefit from automatic image segmentation, due to the large amounts of experience necessary to properly evaluate the images and conditions, and even then facing ambiguity and sometimes leading to disagreement, even between medical professionals.

Traditional image segmentation algorithms operate by iteratively working over an image, as if refining a segmentation until a stopping criterion is met, as the used algorithm determines.

Deep learning is currently playing a crucial role in computer vision, replacing the traditional approaches of feature engineering with learned representations of data. The technology breakthroughs in processing power have allowed for the creation of deep neural networks that achieve state-of-the-art performance in many problems, one of them being image segmentation. However, the concept of segmentation refinement is not present anymore, since usually the images are segmented in a single step.

This work focuses on the refinement of image segmentations using deep convolutional neural networks, first by exploring improvements for a method for image segmentation by quality inference. This approach tries to segment an image by first predicting the quality of an existing segmentation and then refining it through gradient descent, by modifying the input segmentation mask while trying to maximize the expected quality. Possible improvements include data augmentation, siamese networks, different quality metrics and possible stopping criteria. After that, a network for direct segmentation refinement with an extra quality output is introduced.

We show that data augmentation tuned to the base model and the application of siamese networks can be used to improve the quality inference performance, despite not improving the segmentation refinement process, and that the quality concept can be used as a regularizer while training a network for direct segmentation refinement, achieving better performance results.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

CAD	Computer Aided Diagnosis
CPU	Central Processing Unit
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DCNN	Deep Convolutional Neural Network
FN	False Negative
FP	False Positive
GPU	Graphical Processing Unit
MAE	Mean Absolute Error
MSE	Mean Squared Error
RNN	Recursive Neural Network
ROI	Region of Interest
TN	True Negative
TP	True Positive

Chapter 1

Introduction

Contents

This first chapter gives an overview of the dissertation, by introducing its context, motivation and goals. Finally, it presents the structure of the remaining chapters.

1.1 Context

Image segmentation consists of partitioning an image in multiple parts, without information about what exactly each part represents [48]. However, this expression is normally used when referring to semantic image segmentation, which also tries to partition an image into multiple parts, but those should have some semantic meaning, *i.e.* belong to one of multiple classes. This represents one of the great challenges for computer vision, since image segmentation is at the base of many computer vision problems.

Traditional image segmentation algorithms, like region-growing methods, usually operate by iteratively working over the image, optimizing some sort of function or performing some operation until a stopping criterion is reached, and the image is considered segmented.

Medical imaging plays a crucial role in the diagnosis and treatment of patients. The medical images are, however, dependent on the visual interpretation of the medical professionals, which is time consuming and prone to error, since some areas require many years of experience in order to properly analyze such medical images. Furthermore, the subjective nature of the interpretation sometimes leads to disagreement, even between experient medical professionals. This motivates the research into Computer Aided Diagnostic (CAD) systems, which should assist the medical professionals and allow for a faster and more accurate diagnosis, as well as reducing the associated costs. At the base of such systems is the image analysis, particularly the semantic segmentation.

1.2 Motivation

In the past years, deep learning has overshadowed traditional algorithms in many areas, by achieving state-of-the-art performance over the old approaches, and sometimes even outperforming humans. One of those areas is image segmentation, where deep learning has shown promising results [17].

While deep learning has also been applied to medical image analysis [47], the requirement for huge amounts of data has limited its applications in the field. In the past years some new datasets have been made available, with enough images to allow for the training of deep neural networks.

Deep learning architectures vary, but most apply the same single-step paradigm to image segmentation where one image is given to the network as input, and the segmentation is obtained as output. This contrasts with the traditional iterative segmentation methods which needed multiple iterations, usually starting from scratch or from a coarse segmentation and progressing towards a more fine result.

Fernandes et al. [11] have developed a novel segmentation paradigm that tries to segment an image indirectly, by first learning the concept of segmentation quality, and then applying back-propagation over an initial segmentation multiple times in order to iteratively refine it.

1.3 Goals

This research focuses on the study of the architecture presented by Fernandes et al. [11], the identification of some possible improvements, their implementation and testing. This includes the exploration of methodologies such as multitask, transfer learning, siamese networks and regularization, as well as research into the possible integration of iterative refinement techniques into deep learning architectures.

The developed and implemented solutions will be evaluated on multiple medical imaging datasets with image segmentation metrics and compared to the base results obtained with the original architecture.

Specifically, the main goals can be elaborated as follows:

1. Research into an alternative data augmentation technique, by introducing segmentations that better prepare the network for the segmentation refinement process.
2. Research into possible stopping criteria for the segmentation refinement process.
3. Research into the application of triplet loss and siamese networks to the quality inference, comparison and segmentation refinement.
4. Research into the usage of different and/or multiple quality metrics for quality inference and segmentation refinement.
5. Research into the direct refinement of a segmentation using an encoder-decoder architecture, possibly extended with a quality output.

1.4 Dissertation Structure

The remainder of this dissertation is structured as follows:

- Chapter 2, "Literature Review" (p. 5), provides a literature review on the areas of image segmentation and deep learning, ending with the main work upon which this dissertation builds.
- Chapter 3, "Image Segmentation" (p. 17), addresses the architecture for segmentation by quality inference at a deeper level, exposing the possible areas of improvement focused by this work, as well as suggested solutions to be explored.
- Chapter 4, "Implementation and Results" (p. 25), describes the implementation of the solutions proposed in Chapter 3, the difficulties faced while executing them and the obtained results.
- Chapter 5, "Conclusion" (p. 39), concludes this dissertation with an overview of the accomplished results. Additionally, it includes suggestions for further work.

Chapter 2

Literature Review

Contents

This chapter gives an overview into image segmentation, starting with an outline on the classic image segmentation approaches before moving to deep learning and its complementary techniques. Finally, it introduces the concept of image segmentation by quality inference, which will be the main base of this dissertation.

2.1 Image Segmentation

Image segmentation consists of classifying each pixel in an image according to its semantic meaning. The raw images usually contain some portions that are not relevant to the problem at hand, making it necessary to find the region of interest (ROI), before starting the actual segmentation process.

Each pixel in an image can belong to one of the classes to be considered for segmentation, which can be classified on a broader scale as foreground / background or, on a more fine basis, as each of the more specific classes to be considered for segmentation.

2.2 Traditional Approaches

Traditional image segmentation methods (those which do not use neural networks) make heavy use of domain knowledge in order to properly segment the images, in a process known as feature engineering. Feature engineering is a crucial part of many traditional machine learning approaches,

by using human knowledge and intuition to modify the feature space of the data being analyzed in order to highlight the desired features for further processing [28]. This consists of a lengthy trial and error process that requires large amounts of human involvement and expertise. There is no universally defined method to determine what constitutes a feature, since that is heavily dependent on the problem and type of application.

Depending on what kind of features and how they are being analysed, the traditional approaches for image segmentation can be divided in 4 main techniques [41]: thresholding, clustering, region-based and edge-based approaches.

2.2.1 Thresholding methods

Thresholding consists of one of the simplest image segmentation methods, where the pixels are partitioned depending on their intensity value [3]. Thresholding can be further divided into two main types: global and local thresholding.

Global thresholding

Global thresholding uses a single threshold value for the whole image. It can be used only when the pixels from the background and foreground of an image belong to sufficiently distinct distributions. For each pixel $f(x,y)$, the resulting value $t(x,y)$ is decided according to a threshold T.

$$t(x,y) = \begin{cases} 1, & \text{if } f(x,y) > \text{T} \\ 0, & \text{if } f(x,y) \leq \text{T} \end{cases} \tag{2.1}$$

There are multiple techniques that can be used to determine the threshold. One of the most used ones is Otsu's method [40]. Otsu's method determines the the threshold value that optimally separates the two classes, by minimizing their combined spread (intra-class variance).

Local adaptive thresholding

A single threshold will not perform well for many types of images, since most will have uneven illumination. Local thresholding works by applying a different threshold value $T(x,y)$ to each pixel, which depends on the local statistics around it, such as range or variance [39].

$$t(x,y) = \begin{cases} 1, & \text{if } f(x,y) > T(x,y) \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

2.2.2 Clustering methods

Clustering algorithms aim to group a set of objects in such a way that objects in the same group (a **cluster**) show similarities when compared to the remaining objects in the cluster [44]. In image segmentation, the objects to cluster are the pixels belonging to the image to be segmented, and the similarity to be considered is one or more predefined features which depend on the problem being

tackled. They can range from just using the color of the pixels to more advanced feature vectors that take into consideration extra information from the surrounding area.

The K-means clustering approach aims to partition n observations into k clusters, in which the observations belong to the cluster with the nearest mean [5]. This is an NP-hard problem, but there are some efficient heuristic algorithms that can iteratively converge to a local optimum.

Fuzzy C-means [8] is an adaptation of the K-means algorithm for fuzzy clustering, *i.e.* each data point can now belong to more than one cluster.

Gaussian mixture models assume that each data point is generated from a mixture of a number of Gaussian distributions with unknown parameters [13]. This can be seen as a generalization of the K-means clustering, by incorporating information about the covariance of the data as well as the centers of the latent Gaussian distributions.

2.2.3 Region-based methods

Region-based methods attempt to determine the segmentation region directly, according to a set of predefined criteria [41]. They can be split into two main types: region growing, and region splitting and merging.

Region growing

In region growing methods, the segmentation regions start from a group of seed pixels, which are iteratively increased by appending to each region neighboring pixels that are similar, according to the defined rules [37]. The region growing stops when no more pixels can be added according to some stopping criterion, like the size or shape of the region, or the non-existence of any more candidate pixels.

The watershed algorithm [49] is based on the simulation of a flooding process. It takes a gray-scale image as input and interprets it as a height map, where the intensity values and the height are directly proportional. The algorithm starts by placing a water source at the regional minimums, and a watershed is found when two bodies of water become connected, marking a segmentation border.

Region splitting and merging

Instead of choosing seed points like the previous approach, region splitting and merging divides the image into a set of unconnected regions and then those regions are merged.

2.2.4 Edge-detection methods

Edges in images consist of groups of pixels that present a rapid transition in intensity, when compared to their neighbors. Region boundaries and edges are closely related. There is often a sharp change in intensity at the region boundaries [41] and therefore edge detection techniques have also been used as another segmentation approach.

Active contour models [27] try to segment images along the edges, while also trying to keep a smooth segmentation border. This is achieved by using an energy function which will be minimized.

A deformable spline referred to as *snake* placed on an image. An energy function is defined, consisting of the sum of the snake's internal and external energy. The internal energy controls the deformations made to the snake, while the external energy consists of a combination of the forces caused by the image and possible constraints introduced by the user.

Given an initial position for the snake, the energy function is then iteratively optimized, using for example gradient descent.

2.3 Deep Learning

Contrary to the conventional techniques, deep learning allows for the automatic learning of the necessary features for the task at hand, eliminating the manual feature engineering step required before [47]. Its booming usage and success in the last few years can be attributed to the technology advances in processing power, both central processing units (CPUs) and graphical processing units (GPUs), to the availability of large amounts of data and to the advancements in the algorithms.

Deep Convolutional Neural Networks (DCNNs) have demonstrated large success in image-related machine learning tasks. Their training is, however, limited by the amount of training data that is available for their specific task.

In 2009, Jia Deng et al. [26] introduced the ImageNet dataset, which contained 3.2 million individually annotated images. The first big deep learning leap happened in 2012, when Krizhevsky et al. [32] successfully trained a deep convolutional neural network on the ImageNet dataset, winning the ImageNet Large Scale Visual Recognition Challenge, achieving an error rate of 15.3%, compared to the 26.2% achieved by the second place candidate. This caused a massive increase in interest and research in the field of deep learning, which has seen many different improvements, techniques and architectures since then.

2.3.1 Encoder-Decoder Architectures

The most used technique for segmentation tasks using deep neural networks consists of encoder-decoder architectures [1]. Encoder-decoder architectures operate in two phases: first, an input image is encoded into a smaller representation, which contains some semantic meaning. On the second step, the image is then decoded into the final segmentation. The encoding step is made of a sequence of convolution and max-pooling layers. The decoding step uses upsampling and convolutions, in a similar symmetric fashion. SegNet [1] was one of the first models to use such an implementation.

Encoder-decoder models have difficulties in avoiding the so-called checkerboard problem caused by the upsampling process. During the encoding step, some information is inevitably lost, which prevents the decoder from producing a refined segmentation, since adjacent pixels in the up-sampled feature map lack relationship information.

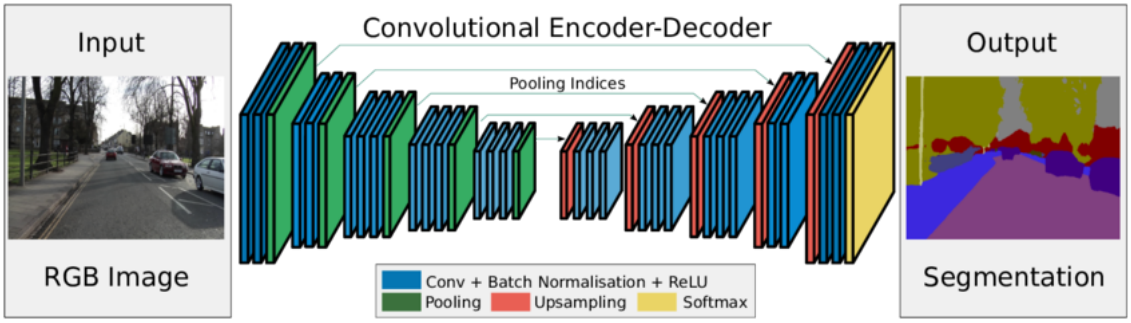


Figure 2.1: SegNet architecture [1]

RefineNet [34] tackles this problem by introducing long-range residual connections to the deeper layers, propagating the earlier captured features, allowing for a higher-resolution segmentation.

Pixel Deconvolutional Networks [14] introduce direct relationships between intermediate feature maps, overcoming the checkerboard problem and showing more accurate segmentation results.

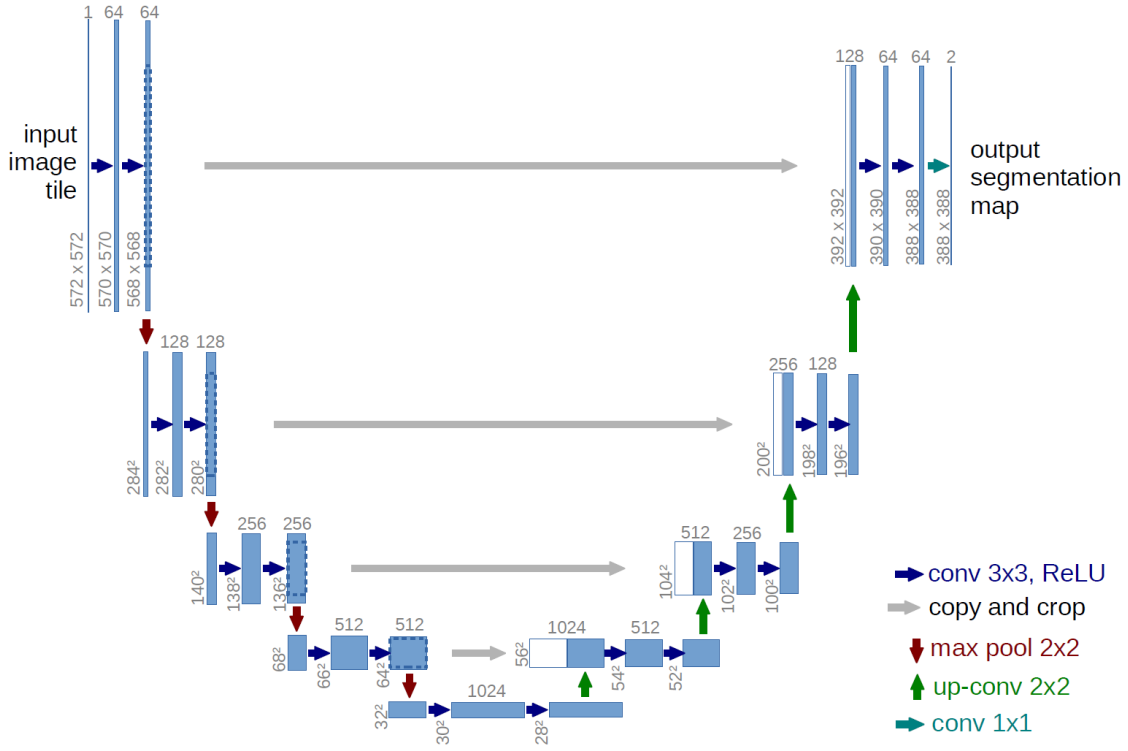


Figure 2.2: U-Net architecture [45]

One very successful evolution of the SegNet is the U-Net [45], which won the ISBI cell tracking challenge 2015, by a large margin. The U-Net uses an encoder-decoder architecture, but before

each downsampling step in the encoding path, the resulting feature map is concatenated to the corresponding level in the decoding path, as shown in Figure 2.2. This allows for high resolution features to be combined with the upsampled output, leading to a more precise output avoiding the previously mentioned checkerboard problem.

Outside of the encoder-decoder models, Jegou et al. [25] have achieved state-of-the-art results with a network based on DenseNet [20], which connects each layer to every other layer, in feed-forward fashion.

To improve the DCNNs localization, some modules can be introduced to broaden the context understanding, such as Conditional Random Fields (CRFs) [6], Recurrent layers [51] and Dilated Convolutions [54].

Recurrent layers [51] are composed of 4 RNNs (Recursive Neural Networks) coupled together in a way that captures the local and global spacial structure from the input data. This is done by first sweeping the image vertically with two RNNs (one from bottom to top, the other from top to bottom), using non-overlapping patches. Then, the resulting projections from both RNNs are concatenated, creating a composite feature map which is then swept horizontally by a new pair of RNNs, in the same manner, but without using patches.

Dilated convolutions [54] use the same convolution filter parameters in a dilated form, managing to enlarge the receptive field and thus incorporating more context without introducing extra parameters or computation cost.

Besides architectures, there are training techniques that can lead to an easier training of deep neural networks without a large amount of training data. One such technique is transfer learning [53]. Deep neural networks trained on natural images exhibit similar features on the first layers, not tied to a specific dataset, like edge detectors. This makes it possible to train a network first on a large dataset such as ImageNet, and then locking the first layers, training just the next ones on the desired dataset.

Iglovikov and Shvets [22] have shown that using the weights from a VGG11 network pre-trained on the ImageNet dataset as the encoding path of a U-Net leads to better results even with a small training dataset, having won the Kaggle's "Carvana Image Masking Challenge".

2.3.2 Regularization

Regularization is a key strategy to avoid overfitting [15]. Overfitting occurs when the network starts to learn the actual training data and possibly the noise associated with it, instead of the features that lead to the expected output, and thus failing to generalize for new examples. One of the simplest examples can be seen when trying to fit a polynomial curve to sampled data (see Figure 2.3). With the right fit, the curve will miss some points but it will be smooth and close enough to the original function. When it overfits, it will display an erratic behavior and be completely off, despite fitting almost perfectly to the training samples. In neural networks, overfitting occurs when the training error is still decreasing, but the validation error starts getting worse, indicating that the network is losing the capacity to generalize.

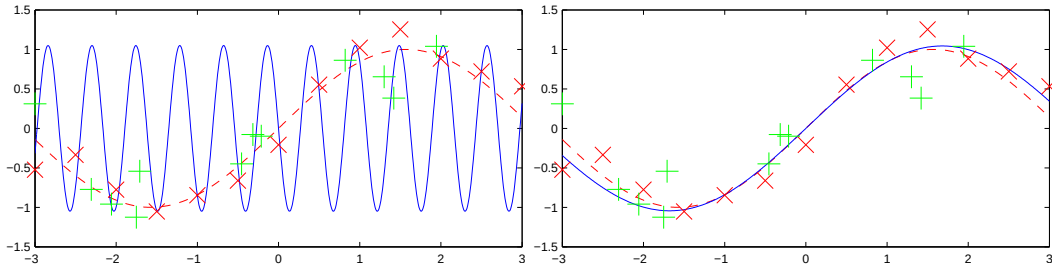


Figure 2.3: Regularization on a polynomial function fit. Training samples in red, validation examples in green, original function represented by the red dashed line. **(a)** The unregularized fit. **(b)** The regularized fit [2]

L1 and L2 regularization are the most common types of regularization. These penalize very large weights by including them as a regularization term in the cost function that is being optimized. Smaller weights lead to simpler models, which reduces overfitting.

L1 Regularization

In L1 regularization, the weights affect the cost function linearly with the sum of their absolute values, scaled by a factor of λ, that scales the regularization strength.

$$CostFunction = Loss + \lambda \times \sum |w| \tag{2.3}$$

L2 Regularization

In L2 regularization, the weights affect the cost function quadratically with the sum of their squared values, also scaled by a factor of λ.

$$CostFunction = Loss + \lambda \times \sum w^2 \tag{2.4}$$

Weight Decay [16] acts directly on the weight updates, multiplying them by a factor slightly smaller than 1, preventing them from growing too large. This is equivalent to L2 Regularization for stochastic gradient descent if the weight decay factor if reparametrized based on the learning rate, but not for adaptative gradient algorithms such as Adam [36].

Another cause of overfitting is the co-adaptation between neurons on the training data. This consists of some network units relying on others to extract certain features, while the desire is for each unit to extract the necessary features by itself.

Dropout [19] is another regularization technique that prevents co-adaptations by randomly disabling units throughout the network. This effectively causes a drop in the network capacity, forcing it to learn multiple models at the same time and averaging them.

DropConnect [52] acts in a similar way, but instead of fully disabling a unit it just disables some of the connections to it. DropConnect is a basically generalization of Dropout, since it can lead to even more models, in addition to the ones that Dropout already makes possible.

Stochastic depth [21] drops entire layers, bypassing them with identity functions, and this way managing to train very deep networks beyond 1200 layers, while still getting improvements in test error.

Early-stopping [15] is one type of cross-validation strategy, where the network training is stopped when the performance on a validation set stops improving. Then, the obtained network is tested on a third set called the test set, to verify that no overfitting occurred on the validation set.

Batch Normalization [24] makes normalization a direct part of the network, by performing it for each training batch between layers. This is useful because as the parameters of the previous layer change, its output distribution is now different, which can saturate the non-linearities in the current layer. Using batch normalization stabilizes the network, allowing for higher learning rates and decreasing the importance of optimal weight initialization.

Most of the described regularization techniques work explicitly, by reducing the effective capacity of the network [18]. Data augmentation [42], on the other hand, has proved to be effective at improving the generalization performance of a network by introducing artificial modifications in the existing training samples, like rotations, crops, flips and deformations in the case of images. Since the network itself is not changed, its capacity remains unchanged. It has been shown that data augmentation alone can achieve the same performance or higher when compared to models trained with other regularization techniques [18], especially when facing small datasets.

2.4 Evaluation Metrics

Although the results can be visually analyzed, in order to objectively evaluate and compare the performance of any segmentation algorithm or variation in parameters, some metrics must be chosen. Fenster and Chiu [9] define and review 3 types of metrics: accuracy, precision and efficiency.

The accuracy of a segmentation determines how close it is to a reference segmentation, which can be the ground-truth or a segmentation determined by an expert in the field that is being studied. These terms are normally used interchangeably.

For each class, each segmented pixel can be then be seen as a True Positive (TP), True Negative (TN), False Positive (FP) or False Negative (FN). TP and TN represent the pixels that were correctly classified as belonging or not to the class, respectively. Similarly, FP and FN represent the pixels that were misclassified as belonging or not to that class, respectively.

The following accuracy evaluation metrics can be defined:

Sensitivity

Sensitivity or Recall, determines the true positive fraction, that is, the fraction of pixels that were correctly classified as belonging to the class.

$$Sensitivity = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|} \tag{2.5}$$

Specificity

Specificity determines the the fraction of pixels that were correctly classified as not belonging to the class.

$$Specificity = \frac{|\mathrm{TN}|}{|\mathrm{TN}| + |\mathrm{FP}|} \tag{2.6}$$

Accuracy

Accuracy determines the fraction of pixels that were correctly classified.

$$Accuracy = \frac{|\mathrm{TP}| + |\mathrm{TN}|}{|\mathrm{TP}| + |\mathrm{TN}| + |\mathrm{FP}| + |\mathrm{FN}|} \tag{2.7}$$

Dice Similarity Coefficient

Dice Similarity Coefficient, also known as F1 score, is the most used metric in medical segmentation. It determines the overlap ratio between the reference and the obtained segmentation.

$$\mathrm{DSC} = \frac{2 \times |\mathrm{TP}|}{2 \times |\mathrm{TP}| + |\mathrm{FP}| + |\mathrm{FN}|} \tag{2.8}$$

Jaccard Index

Jaccard Index determines the intersection over union between the reference and the obtained segmentation.

$$\mathrm{JAC} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FP}| + |\mathrm{FN}|} \tag{2.9}$$

The Jaccard Index is related to the Dice Similarity Coefficient, both being within a factor of 2 from each other:

$$\mathrm{DSC} = \frac{2 \times \mathrm{JAC}}{1 + \mathrm{JAC}} \tag{2.10}$$

The metrics were described in a binary classification setting (either the pixel belongs to a class or not). However, they can be used in multi-class classification via micro or macro averaging [35], where the results are averaged using all the classes to produce one final result.

Sometimes, comparing the segmented areas directly alone doesn't describe the performance accurately. Kim and Huang [29] use the Jaccard index, combined with both the Euclidean distance between the centroids of the ground truth and the obtained area and with the aspect ratio of both areas.

Choosing the proper evaluation metric for the problem at hand is also important, since metrics can have a bias towards/against some properties. Taha et al. [50] devise a method for ranking metrics according to their overall bias.

In many medical imaging applications there is the problem of class imbalance, which occurs when the training data contains many more examples for one class than for the others, making it so that the network becomes biased towards predicting that class [17]. This can be especially problematic for medical applications, since the class with fewer examples usually corresponds to lesions, for example, which will make the network report a lot of false positives. Hashemi et al. [17] developed an asymmetric loss function to mitigate this issue, achieving state-of-the-art results.

2.5 Iterative Segmentation Refinement

There is already some existing work on applying deep learning to the problem of iterative segmentation refinement.

Kim et al. [30] use an encoder-decoder network similar to the U-Net, augmented with an extra input for an already existing segmentation. The existing segmentation is subjected to convolutional layers, before concatenating the obtained feature maps with the ones from the image. A new objective function based on the Dice coefficient is also proposed, which captures the improvement in Dice coefficient between iterations.

Lessmann et al. [33] use CNNs to iteratively segment images of vertebrae. By processing the image in patches from top to bottom, the network retains information about the already segmented vertebrae and uses it to find and segment the next not yet segmented vertebra.

Segmentation methods usually work directly on obtaining output segmentation. Fernandes et al. [11] present a network that infers the quality of a segmentation given an image and segmentation pair. The segmentation quality consists of some evaluation metric like those described in Section 2.4 (dice coefficient in the original work). This allows for data augmentation through the unsupervised generation of synthetic segmentations for an image, given that the segmentation quality for the objective function can be easily determined, having the ground-truth before being augmented.

With the trained model, it is then possible to iteratively refine a segmentation through back-propagation on the input segmentation, towards a local maximum for the quality.

This architecture will be the used as a base for the dissertation, and will be described further in Section 3.1.

2.6 Conclusion

The area of image segmentation is an ever-evolving and very challenging field, showing the need for the combination of multiple techniques in a pipeline to tackle it successfully, from image pre-processing and feature analysis to the segmentation refinement. Traditional methods work, but

need to be fine-tuned for specific situations, and in some fields they lack objective evaluation. Deep learning is promising, but currently faces problems with the lack of training data, especially in the medical field.

There is some existing work into segmentation refinement using deep convolutional neural networks, with promising results. Usually the quality concept is optimized while training segmentation networks, but never directly in the network itself. The quality inference notion and subsequent application to image segmentation introduces a new concept that shows a lot of potential, albeit with some drawbacks and areas for improvement that will be the focus of this dissertation.

Chapter 3

Image Segmentation

Contents

This chapter presents the architecture for image segmentation by quality inference at a deeper level, some problems and possible improvements that will be explored during this dissertation.

3.1 Segmentation by Quality Inference

Briefly introduced in Chapter 2, the technique proposed by Fernandes et al. [11] uses a deep network that predicts the segmentation quality, allowing for the iterative refinement of a segmentation using backpropagation. The model that originates from this architecture will be referred to from now on as oracle.

Traditional deep learning segmentation techniques work directly on learning a model, depicted in Figure 3.1a which, given an image, predicts the desired segmentation by optimizing some metric of quality. Both the input and output spaces are multidimensional, giving rise to the necessity of strong techniques for data augmentation in order to facilitate learning.

(a) Traditional (b) Oracle

Figure 3.1: Traditional deep learning models / New oracle model

In this oracle architecture, the input consists of an image and mask pair, and the output is now a single number (see Figure 3.1b).

Conceptually, one possible interpretation consists of reversing the decoder from an encoder-decoder network, turning the segmentation output into an input and using both inputs to learn a

proxy function, now using the old latent dimension as a quality output, as illustrated in Figure 3.2. The inference process for the segmentation refinement can then be achieved through iterative backpropagation on the input mask, maximizing the expected quality, with an associated step size in order to avoid large segmentation changes through large gradients.

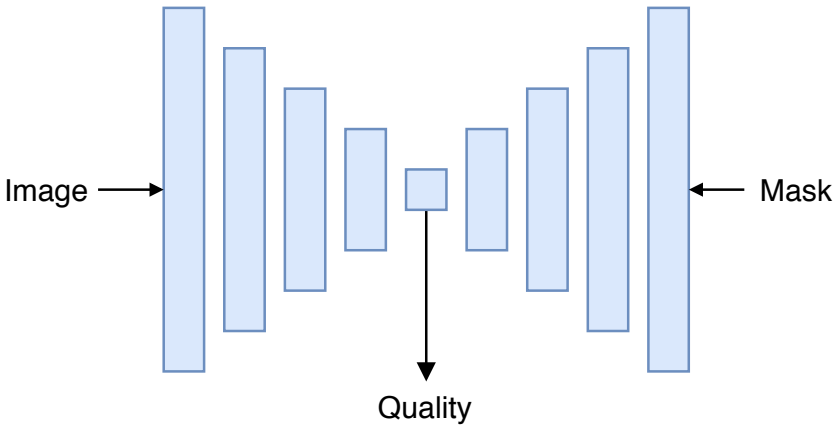
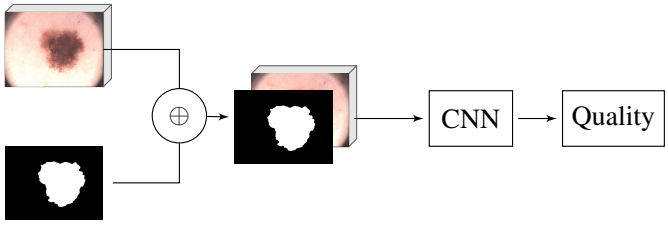


Figure 3.2: Oracle network: reversed encoder-decoder concept

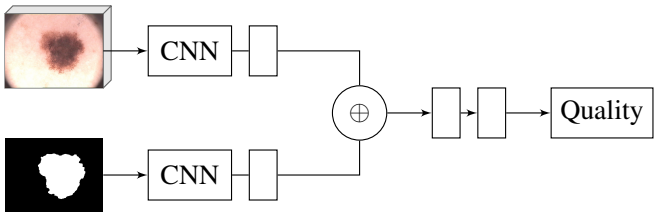
The oracle implementation can be done in one of two ways:

(a) concatenating the mask to the image as an additional channel and using a traditional CNN (see Figure 3.3a)

(b) having two separate streams, one for the input image and other for the segmentation mask and then concatenating their latent representation (see Figure 3.3b)



(a) Single stream



(b) Dual stream

Figure 3.3: Oracle network approaches: **(a)** single stream, **(b)** dual stream [11]

Both approaches have some drawbacks, because the images and segmentation masks belong to different categories (the image consists of real values, while the mask is binary), which in **(a)** may difficult learning while being handled by the same operation (convolution) and in **(b)** may result in very different latent representations.

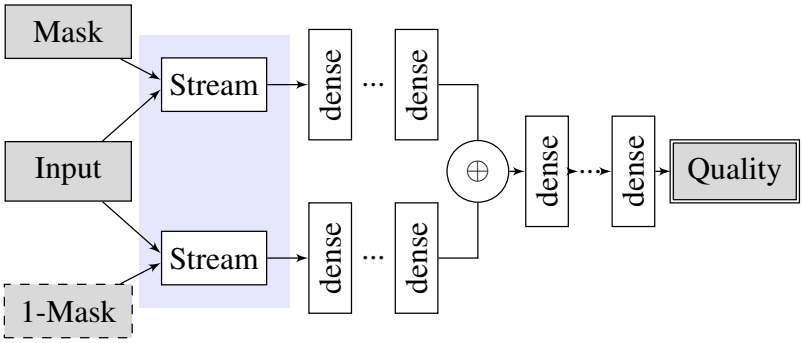


Figure 3.4: Oracle network: overview diagram [11]

This problem is tackled in the original work by having two streams that attempt to model the categories represented by the mask (one for the background, other for the foreground). The streams then communicate ("gossip") between each other, increasing/decreasing their confidence in the classification of each pixel. The gossip streams are followed by dense layers, with the final quality score as output. The full network architecture is shown in Figure 3.4.

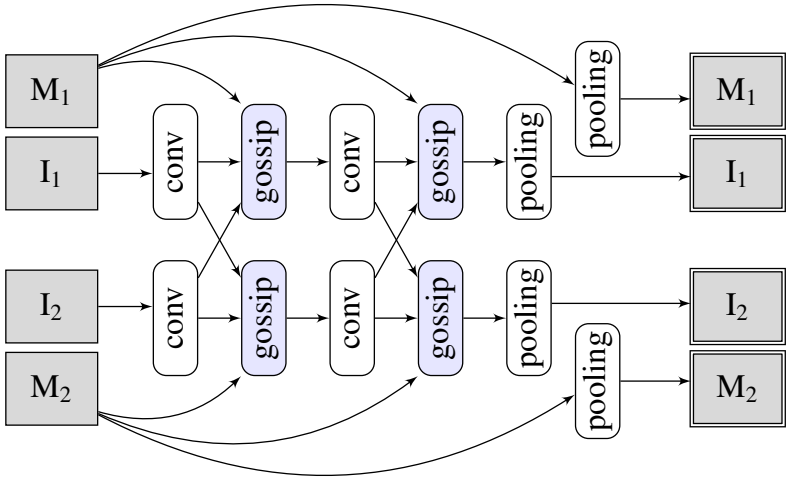


Figure 3.5: Oracle network: streams diagram [11]

Each stream receives as input the image being segmented and its corresponding foreground or background mask (which corresponds to the inverted foreground mask, in binary classification). Each stream, shown in more detail in Figure 3.5, is made up of alternating convolutional and gossip blocks, with pooling applied at the end of the stream to both the obtained feature maps and

segmentation masks. This provides interaction between the streams at each level of resolution, allowing for an early reinforcement or penalization of the respective classification. At the end of a stream, average pooling was used.

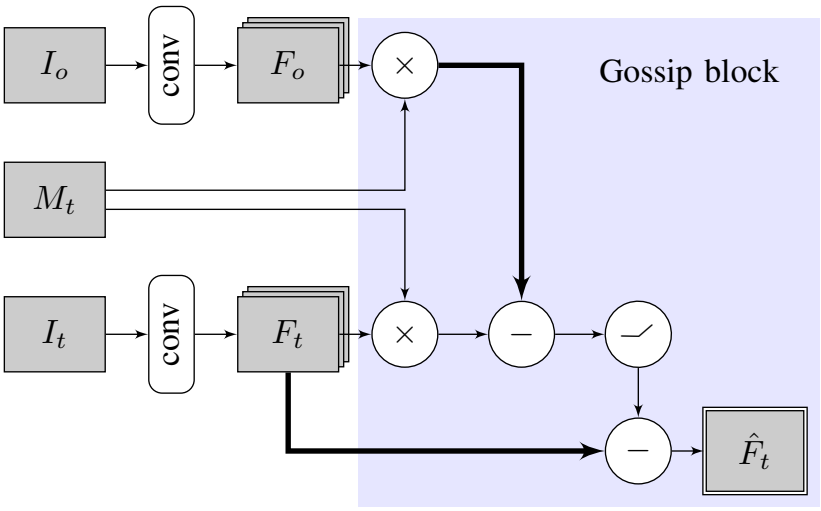


Figure 3.6: Oracle network: gossip block. Bold arrows indicate the first argument of the operation, which are not commutative [11]

The gossip blocks, shown in detail in Figure 3.6, receive as input the feature map from the previous corresponding convolutions on both streams and its own stream's segmentation mask. Then, the stream activations are penalized, if they have a stronger value in the opposite stream.

Contrary to the traditional encoder-decoder networks, where the same data augmentation transformations have to always be applied in parallel to both the image and the segmentation mask, this new technique allows for different transformations to be applied to the image and mask, since the output quality can then be calculated with the ground truth, incrementing the available training data further than it was possible before. This should allow the network to learn the impact of each type of error in a segmentation's quality, and opens up the possibility to the usage of a large number of data augmentation transformations. The default data augmentation transformations used by the reference work are enumerated in Table 3.1 with the respective parameters.

Table 3.1: Original data augmentation transformations, and the variable parameters for each transformation.

Transformation	**Parameters**
Elastic deformations	α, θ, α'
Morphological (erosion and dilation)	size
Random pixel switches	# pixels
Rotations	angle
Flip transformations	horizontal and/or vertical
Shifts	xOffset, yOffset

In order to provide the network with a balanced range of dice coefficient values, the impact of each parameter on the dice coefficient was determined empirically. For each transformation, the parameters were drawn using grid-search, and the dice-coefficient between the ground-truth and augmented masks was calculated and discretized into B bins ($B = 8$ in the original work). Stochastic transformations (elastic deformations and random pixel switches) were repeated 10 times, for each ground-truth mask. With that distribution determined, a second distribution was computed, from which parameters could be sampled while ensuring a uniform distribution of the dice coefficient across all bins.

Given an image and mask pair, the refinement process can be seen as "walking" through the solution space, by adding / removing parts of the segmentation while trying to maximize the predicted quality. This is done in practice using backpropagation over the mask, maximizing the predicted quality.

With this in mind, there are some techniques and modifications that could improve both the quality prediction and the subsequent iterative segmentation process, which will be presented in the following sections, along with some difficulties that were identified and the investigation into possible solutions.

3.1.1 Data Augmentation tuned to the base model

In the reference work, the network is trained with generic data augmentation techniques. While this covers a very wide range of transformations and gives good results for the quality prediction process, it might benefit from a more tuned data augmentation process, more directed to the final goal of segmentation refinement.

The default data augmentation produces results that, while relevant in the training for the quality evaluation, are not representative of the inputs the network will usually try to refine, obtained from the base models.

With this in mind, the network should be trained with segmentation inputs that better illustrate the refinement process, *i.e.* the masks that are seen throughout the segmentation space between the output from the base models and the ground truth, in order to better prepare it for the corrections necessary in order to properly refine the segmentations.

3.1.2 Stopping Criterion for the Refinement Process

One problem of any refinement process that doesn't have a clear finished state is determining the stopping criterion, that is one which doesn't stop too early but also doesn't continue when the segmentation is possibly getting worse. This is especially problematic for the latter situation, since the network will always try to improve a segmentation, even when that "improvement" is actually destructive. Furthermore, since we are actually using backpropagation over the mask, the network will not be able to correctly predict the performance of its own refinement since, by definition, when trying to refine the segmentation by optimizing the quality, every step it takes will improve its own perception of the segmentation quality, even when it declines (assuming the step size is

small enough not to go directly to a lower quality value). One such case is illustrated in Figure 3.7 and Figure 3.8. They show one extreme case where the image is hard to segment and the network quality prediction actually has a large error. For every refinement step, the network's predicted quality increases, while the actual quality is deteriorating. As seen in Figure 3.8, in this case the network started opening holes in the area that should actually be segmented.

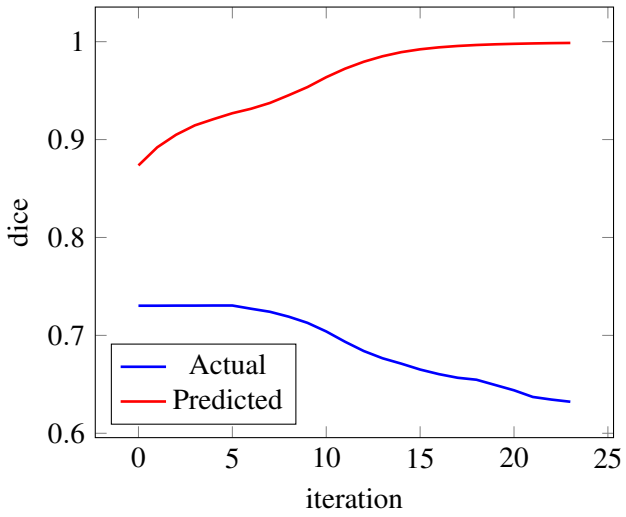


Figure 3.7: Bad refinement case: predicted and actual dice coefficient

(a) Image (b) Ground truth (c) Base (d) Refined

Figure 3.8: Bad refinement case: segmentation deterioration. The segmentation was refined for 11 iterations.

It is then necessary to try to find another stopping criterion that doesn't fully rely on the actual value of the predicted quality. This could involve the actual variation rate of the predicted quality or other metrics such as the foreground/background ratio and their variation along the refinement process.

3.1.3 Siamese Networks

Predicting the quality of a segmentation is not an easy task, especially in certain domains with more difficult boundaries and thus ambiguity on the fine details of the segmentation. Siamese Networks could be used to try to improve this distinction between segmentations of the same

image, giving the network not only one but two different segmentations per image, and thus trying to also learn directly what makes one segmentation better over another.

This quality comparison idea is inspired by the Triplet Loss [46], where a network learns to distinguish faces using 3 examples: an anchor, a positive example and a negative example (see Figure 3.9). The network then tries to minimize the distance between the anchor and the positive example (because they are from the same person) and maximize the distance between the anchor and negative example (they belong to different persons).

In the case of quality prediction, the image that is being segmented is used as an anchor of sorts, while the network must learn the concept of image quality, while at the same time reinforcing the differences between two given segmentations of different qualities.

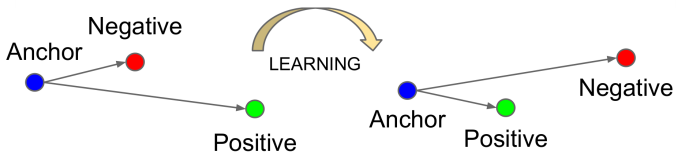


Figure 3.9: Triplet Loss [46]

3.1.4 Different Output Metrics

While the dice coefficient works for evaluating the actual overlapping areas from the reference segmentation and the output of a network, it treats every pixel in the same way, giving the network no way to easily discern areas of more importance for the segmentation. It might focus on the easier areas which provide a boost in the dice coefficient, not paying much attention to the details that would improve it further. There are metrics that might better capture different segmentation quality semantics other than the overlapping area, like the distance between the segmentations' borders.

3.1.4.1 Multiple Output Metrics

Since different metrics evaluate different segmentation concepts, after having identified other metrics a network that predicts multiple metrics for the same image pair could also be achieved, which theoretically could then use the concepts represented from each metric to refine a given segmentation, combining them into a more comprehensive concept of segmentation quality.

3.2 Direct Refinement Networks

As mentioned in Section 2.3, most of the state-of-the-art deep segmentation architectures use encoder-decoder architectures, which face some resolution and context loss during the encoding

process. Similarly, when attempting to segment an image, the network will give the same importance to any area on the entire image, making it harder for it to discern some parts that might be more important for later segmentation.

By giving the network some extra information before the encoding step in the form of a previous attempted segmentation by another network, the encoder can now learn how to use that information to focus more thoroughly on certain areas of the image that are believed to be close to the segmentation, thus refining the provided mask into a better one.

3.3 Quality Output Extension

A refinement network could also be extended with a quality output, calculated by evaluating the network's latent dimension after the encoding step, outputting the quality of the segmentation before the refinement step. By training the network with this quality output, it would be forced to not only learn how to improve a given segmentation but also its quality, which would indicate how far the input segmentation is from the desired ground-truth.

The quality output can also be used to evaluate the segmentation quality, allowing for the iterative process to continue until the quality stops improving. This time, however, we should not face the same difficulties presented in Section 3.1.2. Since there is no backpropagation involved, the network should now more easily identify a quality decrease from a bad iteration step and stop the segmentation refinement. While this is actually arguable, since it would not be a problem in the first case if the network had learned the quality perfectly, training a perfect network is not feasible for many tasks. With that in mind, it is not safe to rely fully on backpropagation using an imperfect network, with no other stopping criterion.

3.4 Summary

Segmentation quality prediction can be used to iteratively refine a mask through backpropagation, by maximizing the expected quality.

This new technique has some possible enhancements that might improve the obtained results, such as extended data augmentation through the introduction of segmentations that further illustrate the refinement process, siamese networks for quality comparison and different output metrics that capture other segmentation semantics. The stopping criterion for the refinement process is still an open problem with no clear solution, given the limitations faced by the quality prediction when using backpropagation, which does not allow for the predicted quality to decline, even when it does in reality.

Encoder-decoder architectures can be extended with an extra channel for segmentation refinement and a quality output extension, which should allow for the simultaneous direct segmentation refinement and quality prediction.

Chapter 4

Implementation and Results

Contents

This chapter presents the implementation details for the improvements proposed in the previous chapter, as well as the performance results obtained.

4.1 Introduction

All of the solutions described in this chapter were implemented in Keras, using TensorFlow as the backend, given the fast prototyping provided by Keras with its high level APIs, the author's familiarity with TensorFlow, its high performance, good documentation and support, and the already existing code-base and trained models from the reference work [11].

In order to allow reproducible results, the dataset partitions and random seeds are fixed. The base models used are the same as the ones used in the reference work.

All the images are directly used in RGB format, with all the color components normalized to the $[0, 1]$ range. There is minimal preprocessing applied, being just resized to 128×128 to conform to the original work and allow for faster training and the need for lower computational resources.

For the masks, a binary setting is considered, where pixel values of 0 indicate background and pixel values of 1 indicate foreground (the subject of interest being segmented on each dataset).

The training is done for up to a maximum of 500 epochs or 50 epochs without improvement on the validation set, in order to avoid overfitting.

Unless otherwise stated, the hyperparameter configuration for the models is the one described in Table 4.1, determined to have the best performance using cross-validation. Adam [31] was used as the algorithm for gradient optimization, with a learning rate of $1e^{-4}$.

All the dice coefficient values in the results have been multiplied by 100, being in the form of a percentage, for easier readability.

Table 4.1: Model hyperparameters

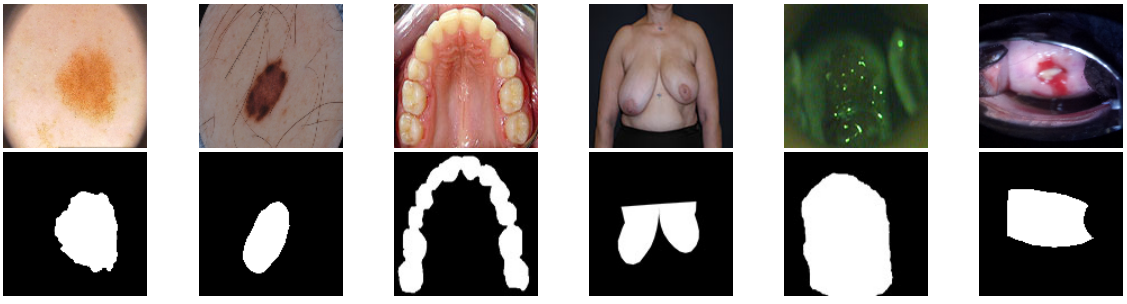
Hyperparameter	**Value**
# Convolution Levels	4
# Consecutive Convolutions	2
# Convolution Filters	32
Convolution Filter Size	3
# Dense Levels	1
Dense Stream Width	512
L2 regularization	0.001
Convolution Activation	ReLU

4.1.1 Datasets

For the training and evaluation of the proposed solutions, the datasets summarized in Table 4.2 were used. Some examples from each dataset are displayed in Figure 4.1

Table 4.2: Datasets used and partition sizes

Dataset	**# images**	**# train**	**# test**	**# validation**
PH2 [38]	200	120	40	40
ISBI 2017 [7]	2750	2000	600	150
Teeth-UCV [12]	100	60	20	20
Breast-Aesthetics [4]	120	72	24	24
Cervix-HUC [10]	287	171	58	58
Cervix-MobileODT [23]	1613	940	338	335



PH2 ISBI 2017 Teeth-UCV Breast HUC MobileODT

Figure 4.1: Datasets - image and segmentation samples

The PH2 dataset is used as a base to validate the correct implementation of the algorithms and architectures, given its small size which allows for the fast training of a network and the high-contrasting foreground/background, making segmentation learning easier for an initial validation of the correct operation of an algorithm or network.

4.2 Data Augmentation tuned to the base model

In addition to the default data augmentation transformations used in the reference work, which were already outlined in Table 4.3, an extra random interpolation step was added between the base masks and the ground-truth, introducing extra examples that illustrate the refinement progress from the base mask to the desired segmentation.

4.2.1 Interpolation between masks

The segmentations between the base models and the ground-truth are generated using the shape-based interpolation approach described by Raya and Udupa [43] and illustrated in Figure 4.2. It comprises 4 main steps:

1. For both masks, find the perimeter of the objects. A pixel belongs to the perimeter if it is non-zero and next to at least one zero-valued pixel.

2. For both masks, find the distance map which contains, for each pixel, the distance to the closest perimeter pixel. This distance map should be then converted to the signed distance map, which should be positive for every pixel inside the object and negative for values outside the object.

3. Interpolate between the corresponding pixels in both signed distance maps using linear interpolation, with $\alpha \in [0,1]$ to control the interpolation.

4. Convert the interpolated signed distance map back to a binary contour by identifying the zero-crossings: if the pixel value is greater than 0 it becomes 1, else it becomes 0.

The obtained interpolation results present an almost linear variation in the dice coefficient between the interpolated mask and the ground truth, shown in Figure 4.3. This results in an approximately uniform distribution of the resulting dice coefficients when the interpolated masks are sampled randomly. This is desirable, allowing for a uniform coverage of the space between the base segmentation and the ground truth, without bias towards one side or the other.

The interpolation can now be used in order to either train a quality network from scratch or to fine-tune a network that was already trained with the more generic data augmentation described before.

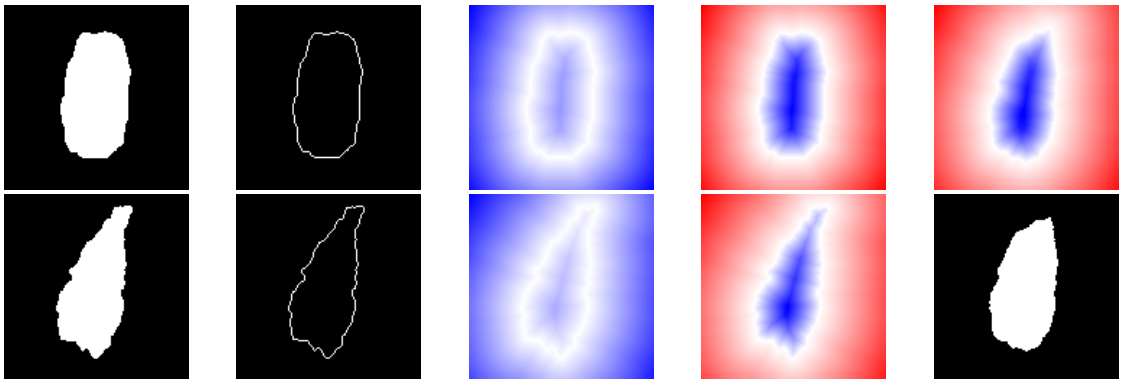


Figure 4.2: Mask interpolation shape steps. For the distance maps, white corresponds to 0, red corresponds to values greater than 0 and blue to values lower than 0. In the figure: **(a)** Segmentations A and B, **(b)** their respective perimeters and **(c)** distance maps, **(d)** their respective signed distance maps, **(e)** the interpolated signed distance map and interpolated segmentation.

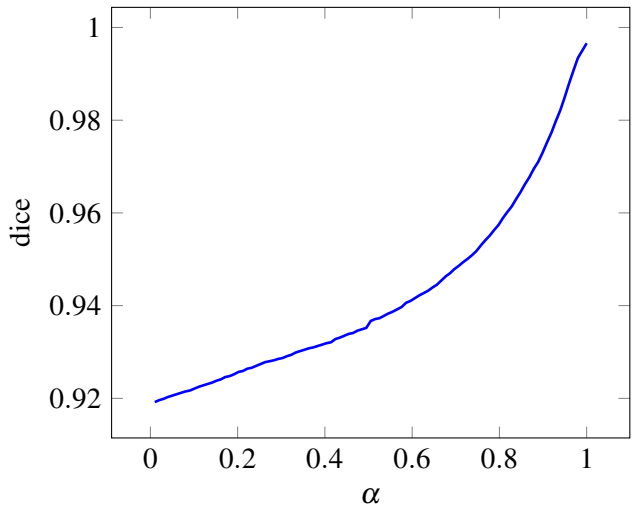


Figure 4.3: Dice variation along interpolation between base mask and ground-truth

4.2.2 Training and results

For the training, a base model was used to generate the base segmentation masks. Then, with a chance of 80%, a random interpolation between the base model output and the ground truth is generated. Otherwise, the base mask or the ground truth are augmented with a random morphological transformation (erosion or dilation), in order to also cover some segmentations outside of the interpolation range.

For the evaluation, the base masks generated from the validation set are first refined until the quality stops increasing, in order to determine the number of iterations. Then, that number of iterations is used to refine the test set and obtain the performance results.

It is not possible to train a network from scratch with just interpolated masks, since that does

not provide enough information and the network will not learn (the predicted quality would be maximum for most of the masks coming from a U-Net, leading to no gradients to backpropagate and thus no refinement). The interpolation was then incorporated in the existing learning approach in 3 different ways, with all of the results presented in Table 4.3:

- from scratch, using interpolation side by side with the already existing default data augmentation (**Int + Def**)
- interpolation alone, but starting from the weights of a pre-trained network with the default data augmentation (**Warm + Int**)
- side by side with the already existing data augmentation, but again starting from the weights of a pre-trained network, as to try to not lose the existing training progress (**Warm + Int + Def**)

Table 4.3: Interpolation for Data Augmentation: Refinement performance results, in terms of Dice Coefficient. **Base** indicates the U-Net performance, with no refinement. **Oracle** indicates the original oracle refinement performance. **Int** indicates interpolation, **Warm** indicates the pre-trained weights were used, **Def** indicates the default data augmentation. The best result for each dataset is highlighted in bold.

Dataset	Base	Oracle	Int + Def	Warm + Int	Warm + Int + Def
PH2	83.70	84.09	**85.23**	84.29	84.91
ISBI 2017	71.35	76.52	**76.89**	76.17	76.44
Teeth-UCV	85.85	**85.91**	80.50	80.60	80.53
Breast-Aesthetics	93.08	**93.31**	93.02	93.12	93.04
Cervix-HUC	77.25	**77.26**	76.85	76.85	76.86
Cervix-MobileODT	88.24	**88.25**	87.24	87.19	87.19

There is a slight improvement in the performance obtained when the network is trained from scratch with the interpolation combined with deformed masks for the PH2 and ISBI 2017 datasets, both easier to segment due to the contrasting images. For the remaining datasets the results present a slight decline in performance, with the worst case being the Teeth-UCV dataset. We theorize that the interpolated masks vary too uniformly between the base segmentation and ground-truth, still skipping many intermediate steps. For example, when interpolating between an undersegmented mask and the ground-truth, the interpolation will grow the mask uniformly towards the reference segmentation, while intermediate steps where it could first grow in only one direction or another (while still increasing the perceived quality) are skipped.

As for quality prediction performance, the results shown in Table 4.4 show improvements for 4 of the 6 datasets when interpolation was used during the training process. The 2 datasets that had better results with no interpolation consist of the two most difficult datasets to segment, due to both their small size and the heavy semantics necessary to properly segment them, especially for the Cervix-HUC dataset, which contains images from 3 different colposcopy modalities, with large differences in color and aspect between them.

Table 4.4: Interpolation for Data Augmentation: Quality prediction performance results, in terms of MSE. **Oracle** indicates the original oracle quality prediction performance. **Int** indicates interpolation, **Warm** indicates the pre-trained weights were used, **Def** indicates the default data augmentation. The best result for each dataset is highlighted in bold.

Dataset	Oracle	Int + Def	Warm + Int	Warm + Int + Def
PH2	0.0106	0.0114	0.0099	**0.0088**
ISBI 2017	0.0260	**0.0159**	0.0188	0.0178
Teeth-UCV	0.0164	**0.0139**	0.0319	0.0190
Breast-Aesthetics	**0.0015**	0.0037	0.0040	0.0037
Cervix-HUC	**0.0521**	0.0550	0.1008	0.0556
Cervix-MobileODT	0.0152	0.0133	0.0152	**0.0131**

Another alternative approach consisted of freezing the first layers of the network, effectively fine-tuning the remaining layers with the new data-augmentation technique, trying not to lose the lower-level features learned by the first layers. However, this did not improve the results.

4.3 Stopping Criterion for the Refinement Process

It was not possible to determine an acceptable stopping criterion by relying on the quality predicted by the network or its variation speed (difference in predicted quality between the refined mask from the current iteration and the previous one) during the refinement process. The difference between predicted qualities at the beginning of the refinement process and at the optimal number of iterations (determined empirically for each image in the dataset) shows no clear correlation that can be used as a stopping criterion. The same can be said about the variation speed in predicted quality, which does not show any perceivable similarity across different segmentations.

Another stopping criterion considered was the foreground/background ratio between the base and refined masks. Since the base mask that is being refined comes from a U-Net which will usually oversegment the image, theoretically this would result in a reduction in that ratio during the refinement process, if the segmentation quality was improving (and the base segmentation was, indeed, oversegmented). However, while that tendency was indeed verified for some datasets, the observed behavior was irregular, with no clear conclusion.

4.4 Siamese Network

The Siamese Network was built using the same architecture as the oracle for each sub-network, with four inputs (two for each image/mask pair) and three outputs, as illustrated in Figure 4.4. The new output consists of the subtraction between the predicted quality from each sub-network, *i.e.* how better or worse one image/mask pair's segmentation is compared to the other.

Initial experiments have shown that training the architecture with just the quality difference as an output is not feasible, since that does not convey enough information for the network to learn the quality semantics. The two alternatives were to either start training from the weights of another

oracle trained normally on just predicting the quality or having the final siamese network with 2 extra outputs for the predicted quality of each sub-network. In the end, the latter alternative was chosen.

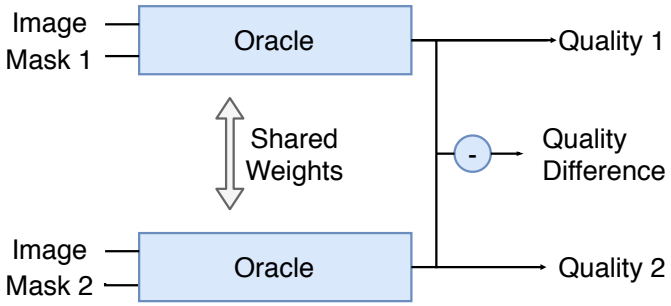


Figure 4.4: Siamese Network Implementation

4.4.1 Results

For segmentation, as shown in Table 4.5 there was a performance increment, again in the PH2 and ISBI 2017 datasets, while the remaining saw approximately the same or even worse performance. Using siamese networks does not improve the segmentation performance. However, when the quality prediction is evaluated, the MSE is actually lower when using the siamese network for most datasets, except the Breast-Aesthetics and Cervix-HUC ones, as seen in Table 4.6. This indicates that the network has improved at predicting the segmentation quality when trained a siamese network setting, despite it not really helping the segmentation through backpropagation.

Table 4.5: Siamese Network Segmentation Performance Results, in terms of dice coefficient. **Base** indicates the U-Net performance, with no refinement. **Oracle** dice indicates the original oracle refinement dice. The best result for each dataset is highlighted in bold.

Dataset	**Base**	**Oracle**	**Siamese**
PH2	83.70	84.09	**85.02**
ISBI 2017	71.35	76.52	**77.03**
Teeth-UCV	85.85	**85.91**	81.68
Breast-Aesthetics	93.08	**93.31**	93.28
Cervix-HUC	77.25	**77.26**	76.86
Cervix-MobileODT	88.24	**88.25**	86.90

Another attempt was made using a siamese network in a binary classification setting, where the goal was to just determine which segmentation was better. The results are not presented because the network would not learn, possibly because a simple binary classification was not enough for it to learn the necessary semantics, even with further outputs added to assist on learning the quality concept.

Table 4.6: Siamese Network Quality Prediction: Performance Results, in terms of MSE. Best result for each dataset highlighted in bold.

Dataset	Oracle	Siamese
PH2	0.0105	**0.0087**
ISBI 2017	0.0258	**0.0191**
Teeth-UCV	0.0127	**0.0119**
Breast-Aesthetics	**0.0014**	0.0017
Cervix-HUC	**0.0522**	0.0595
Cervix-MobileODT	0.0152	**0.0112**

4.5 Different/Multiple Output Metrics

Other than the dice coefficient, two other evaluation metrics were considered:

Cohen's Kappa

Shown by Taha et al. [50] to be one of the metrics with the lowest bias for image segmentation, Cohen's Kappa d_K measures the agreement between two raters on the classification of N items into C mutually exclusive categories:

$$d_K = \frac{p_0 - p_e}{1 - p_e} \tag{4.1}$$

Where p_0 corresponds to the relative observed agreement among raters, p_e being the hypothetical probability of chance agreement, by calculating the probability of each observer seeing each category through random sampling.

Hausdorff Distance

Given two segmentation perimeters, the Hausdorff Distance d_H can be used to determine the maximum distance from a point in one perimeter to the closest point in the other perimeter. This means that two segmentations with a low Hausdorff distance between them will have relatively close borders.

$$d_H(X,Y) = \max\{\sup\inf d(x,y), \sup\inf d(x,y)\} \tag{4.2}$$

If more than one metric is chosen, it is then possible to reuse early layers of the network, splitting the output into multiple segments, as illustrated in Figure 4.5.

For the backpropagation and refinement step, the gradients are added and backpropagation is performed using their sum.

Different metrics may have different ranges, so they have to be normalized. This is done by determining their approximate maximum empirically for each dataset and then using it to normalize the quality values, both during training and evaluation.

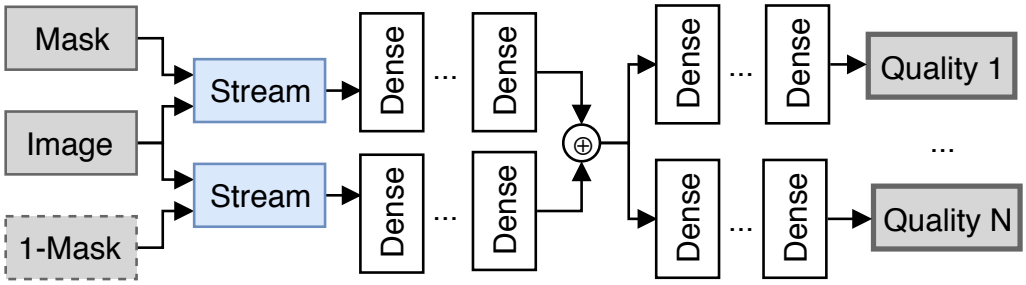


Figure 4.5: Oracle: Multiple outputs

The obtained performance results for the refinement process are shown in Table 4.7 and for the quality prediction in Table 4.8.

Table 4.7: Multiple output metrics: refinement performance results, in terms of dice coefficient. The metrics used for the quality prediction are Dice **(D)**, Cohen's Kappa **(K)** and Hausdorff Distance **(H)**.

Dataset	**D**	**K**	**H**	**D + K**	**D + H**	**H + K**	**D + H + K**
PH2	84.66	84.79	83.22	84.63	83.31	83.29	84.54
ISBI 2017	76.20	75.40	71.30	76.84	74.21	71.52	74.09
Teeth-UCV	80.53	80.55	80.50	80.54	80.51	80.50	80.51
Breast-Aesthetics	93.04	92.87	92.82	93.08	93.02	92.81	93.06
Cervix-HUC	76.85	76.86	76.85	76.86	76.85	76.86	76.85
Cervix-MobileODT	87.18	87.17	87.17	87.18	87.18	87.17	87.18

Table 4.8: Multiple output metrics quality prediction performance, in terms of MSE. The metrics used for the quality prediction are Dice **(D)**, Cohen's Kappa **(K)** and Hausdorff Distance **(H)**

Dataset	**D**	**K**	**H**	**D + K**	**D + H**	**H + K**	**D + H + K**
PH2	0.0083	0.0037	77.9838	0.0206	77.3878	77.5594	77.0769
ISBI 2017	0.0187	0.0024	292.4379	0.0354	289.5102	287.9356	291.4578
Teeth-UCV	0.0133	0.0001	160.5277	0.0166	158.6505	158.6547	158.4219
Breast-Aesthetics	0.0017	0.0002	25.9078	0.0024	25.5951	25.6057	25.4956
Cervix-HUC	0.0516	0.0064	465.3512	0.0982	457.6976	463.0834	456.3224
Cervix-MobileODT	0.0102	0.0008	132.7508	0.0165	131.7186	133.0427	131.7001

The network fails to learn the Hausdorff Distance, presenting a very high MSE for all the datasets, both alone and when combined with other metrics. This can be attributed to the irregular behavior of the metric itself, since the variation of a single pixel can cause large jumps in the output metric, as evident in Figure 4.6. Just a few outliers in the test set can cause a massive misestimation of the quality by the network.

Cohen's Kappa, however, has shown some improvement for most of the datasets when combined with the Dice Coefficient, albeit residual. This might indicate that the Dice Coefficient and Cohen's Kappa somewhat complement each other, and with further fine-tuning might achieve better results.

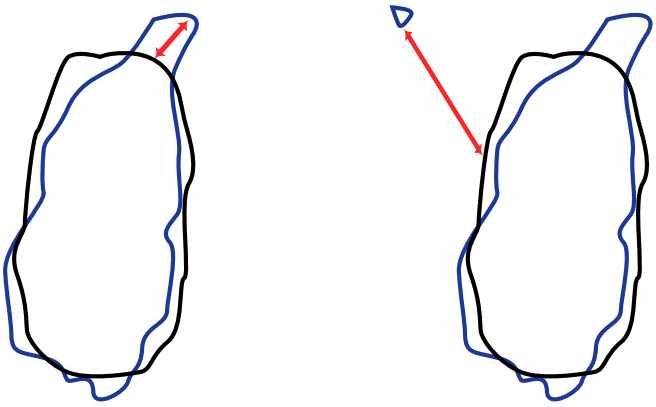


Figure 4.6: Hausdorff Distance - Irregular behavior example: the distance between two very similar segmentation settings, highlighted in red, is much larger between the segmentations on the right when compared to the ones on the left, even though they only differ by a few pixels.

4.6 Refinement U-Net and Quality Output Extension

For the direct refinement network, the U-Net was used as the base architecture. A base segmentation mask, provided by another U-Net trained on the dataset was concatenated to the input image, turning it into a 4-channel input. The network is then trained normally.

The quality output extensions was added to the U-Net latent dimension, taking advantage of all the semantic information extracted by the encoder before the upsampling/decoding process, as depicted in Figure 4.7. Global Average Pooling was used to make it independent of input image dimensions, followed by two dense layers.

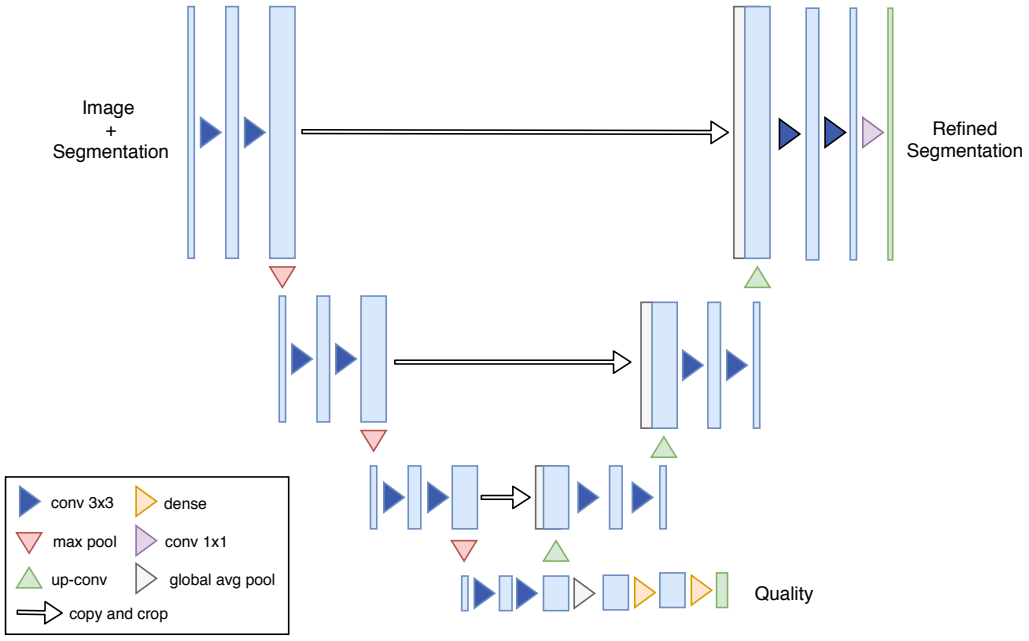


Figure 4.7: U-Net with Quality Output Extension

4.6.1 Results

The performance results for the Refinement U-Net are presented in table 4.9. They were obtained by applying one (1x) and two (2x) refinement steps to the output of the base U-Net. We can see that the first refinement step provides a big quality increase for all datasets, while a second iteration step starts to perform worse than the previous one for the network trained with no quality output.

The network was then trained with the quality output, using the dice coefficient as the prediction quality. We can see that the network performs better than the one trained with no quality output in all datasets, and manages to achieve a quality improvement even for a second iteration step. For the third iteration step the results would, again, decline. It can be theorized that the quality output extension acted as a regularizer, which improved the network's generalization when it was forced to learn the quality alongside the refinement of the segmentation.

Table 4.9: Refinement U-Net Performance, with and without quality output for 1 and 2 refinement iterations. Best result for each dataset highlighted in bold.

Dataset	**0x**	**No Quality**		**Quality**	
		1x	**2x**	**1x**	**2x**
PH2	83.21	89.10	89.14	90.40	**90.41**
ISBI 2017	71.30	80.53	79.89	81.13	**81.18**
Teeth-UCV	80.50	83.21	83.62	83.84	**84.83**
Breast-Aesthetics	92.81	93.01	92.77	**94.11**	93.98
Cervix-HUC	76.85	79.10	78.46	79.60	**79.65**
Cervix-MobileODT	87.17	87.98	87.90	88.26	**88.37**

Since the network can now try to predict its own segmentation quality directly, we can try to use that as a stopping criterion for the refinement process, in order to apply more than 1 iteration step. The results displayed in Table 4.10 however, are somewhat inconsistent and do not show a clear improvement for all of the datasets. It can be speculated that even though the quality prediction has a very low MSE, since the quality values are so close to the ground truth already, the network might have a hard time distinguishing a very good segmentation from an even slightly better one, making it stop prematurely.

Table 4.10: Refinement U-Net Performance, with quality output and automatic iterations. Best result for each dataset highlighted in bold, when compared with Table 4.9.

Dataset	**Auto iter**	**# iter**	**Quality MSE**
PH2	**90.55**	1.5	0.0147
ISBI 2017	80.82	2.4	0.0289
Teeth-UCV	83.98	1.4	0.0060
Breast-Aesthetics	94.00	1.7	0.0014
Cervix-HUC	79.26	2.3	0.0474
Cervix-MobileODT	**88.38**	1.9	0.0199

Another experiment consisted of introducing already refined outputs as extra training examples, effectively training the network to refine its own output, as an extra data augmentation transformation. This was done up to 2 times, for both training a network from scratch and starting from the existing weights of one trained normally. The results are shown in Table 4.11. Since it is now possible to refine a segmentation more than twice without the quality starting to decrease (up to 12 times on some datasets), the number of iterations was determined on the validation set, and then using that as stopping criterion for the refinement of the test set, as it was done in Section 4.2.2. The best possible refinement is also shown, determined by refining the segmentation until the quality declines when compared with the ground-truth (although unrealistic, since when refining new examples the ground-truth is not known to be used as a stopping criterion, but it serves as a good indicator of the theoretical maximum performance).

The results show that the network that used the pre-trained weights outperformed most of the previous results, without overfitting. However, it is evident that using the quality output as a stopping criterion still falls short, stopping the refinement process too early in most cases.

Table 4.11: Refinement U-Net Performance, trained with its own output. **Scratch** corresponds to the network trained from scratch, and **Warm** to the network trained from the pre-trained weights of another network. In parenthesis we show the number of iterations.

Dataset	**Scratch**			**Warm**		
	Auto	**Validation**	**Best**	**Auto**	**Validation**	**Best**
PH2	90.71 (1.9)	90.84 (3)	91.23 (5)	90.61 (1.9)	90.84 (9)	90.88 (11)
ISBI 2017	82.56 (3.1)	82.46 (3)	82.68 (1)	80.85 (2.7)	80.59 (3)	81.23 (1)
Teeth-UCV	83.33 (1.6)	83.66 (3)	85.30 (7)	84.25 (1.5)	84.87 (3)	86.61 (7)
Breast-Aesthetics	93.26 (1.4)	93.18 (2)	93.35 (2)	94.30 (1.7)	94.17 (3)	94.24 (1)
Cervix-HUC	74.32 (7.2)	75.28 (1)	75.29 (1)	79.05 (2.3)	78.89 (2)	79.17 (1)
Cervix-MobileODT	86.79 (2.0)	87.08 (2)	87.08 (2)	88.34 (2.0)	88.22 (2)	88.23 (2)

4.7 Summary

Data augmentation plays a crucial role in the training of deep neural networks, and it should be tuned specifically for each situation. However, interpolation between masks did not improve the segmentation results.

The lack of a good stopping criterion for the segmentation refinement process is a clear problem, even more evident when the available ones either stop too soon and fall short of a much improved result or later than they should and start actually deteriorating the segmentation. No correlation was found between the number of iterations and the predicted quality metric or foreground/background ratio that would allow for a working stopping criterion.

Siamese networks can be used to better learn the quality concept by introducing the concept of a better segmentation directly in the training process, improving the actual quality prediction despite not improving the segmentation process.

Multiple quality metrics can be used concurrently, and they may complement each other by involving different quality concepts. However, the quality metric to be learned must be carefully chosen, with some like the Hausdorff Distance not being learnable by the network due to the highly irregular behavior they present.

The concept of segmentation quality can be directly learned by a network, not only to refine a segmentation but also as a regularizer during the training process of a network for segmentation refinement, improving the performance results even without the quality being directly used in the segmentation process.

Chapter 5

Conclusion

Contents

This chapter presents the conclusions of this research by giving an overview of the work covered by this dissertation, a list of its contributions and some yet to be investigated ideas and problems for future work.

5.1 Overview

In this dissertation we reviewed some of the traditional approaches to image segmentation and the current state of the art. We then described the reference work by Fernandes et al. [11] and approached the goals initially defined in Chapter 1.3:

1. **Research into an alternative data augmentation technique, by introducing segmentations that better prepare the network for the segmentation refinement process.**

   We implemented shape interpolation between base masks to be refined and ground-truth segmentations and introduced that into the training process, but we saw no clear segmentation refinement improvement, despite the quality prediction being enhanced when interpolation is used during training.

2. **Research into possible stopping criteria for the segmentation refinement process.**

   We tried to find a stopping criterion for the refinement process based on the variation in predicted quality and relative foreground/background size, but the irregular behavior they present did not allow for the determination of a clear correlation to be used as a reliable stopping criterion.

3. **Research into the application of triplet loss and siamese networks to the quality inference, comparison and segmentation refinement.**

We implemented an architecture based on triplet loss and siamese networks that uses the existing architecture to compare the difference in quality between segmentations, while at the same time training each sub-network to predict a more accurate quality. While the segmentation refinement process saw no performance improvements, the segmentation quality prediction has seen an enhancement in performance, through a lower MSE in the quality prediction, which indicates a better learning of the quality concept.

4. **Research into the usage of different and/or multiple quality metrics for quality inference and segmentation refinement.**

   We tested two alternative metrics to the Dice Coefficient: Cohen's Kappa and the Hausdorff Distance. Cohen's Kappa has shown some residual improvements in segmentation performance when combined with the dice coefficient. The Hausdorff Distance was not learnable by the network, due to its very irregular behavior even when the changes betwen segmentations are very small.

5. **Research into the direct refinement of a segmentation using an encoder-decoder architecture, possibly extended with a quality output.**

   By extending a U-Net with an extra channel for an existing segmentation, we showed improved segmentation results when compared to a normal U-Net. Adding the quality output revealed an additional improvement in segmentation refinement performance, even though the quality output itself was not being directly used for the segmentation process, indicating that the quality concept can be used as a regularizer.

5.2 Contributions

The main contributions of this dissertation are the following:

- Implementation of mask shape interpolation as a data augmentation technique and its evaluation.
- Implementation of segmentation quality comparison using siamese networks and its evaluation on quality prediction.
- Evaluation of alternative metrics and their combination for segmentation quality inference and segmentation refinement.
- Application of the quality concept as a regularization method for image segmentation.

5.3 Future Work

5.3.1 Direct gradient optimization for backpropagation refinement

Currently, the network described in Section 3.1 refines a segmentation using backpropagation, but the gradients are not directly trained for that purpose. It should be possible to include the

application of the gradient to the mask (*i.e.* one refinement iteration) in the loss as another output, allowing for the optimization of not only the quality prediction but also the quality increase of one refinement step for the current image/mask pair.

5.3.2 Fine-tuning for transfer learning

In the original work the segmentation refinement was tested across datasets, showing that the segmentation quality concept can be applied to other datasets. This was done with no further fine-tuning specific to the target dataset, which might improve the results even further.

5.3.3 Quality-based ensembles

The quality predicted by the oracle could be used with an ensemble of other segmentation models allowing, for example, for custom weighing of the predictions from each model according to the predicted quality.

5.3.4 Multi-Class Segmentation

All of the work was done in a binary segmentation setting. The same principle could be applied to multi-class segmentation using either one network for all of the classes in question or one network per class, both for the quality prediction and segmentation refinement.

5.3.5 Weakly supervised learning in sequences of similar images

When segmenting a sequence of similar images (*e.g.* from a video) the changes between frames are usually quite small, being mostly confined to translations, so the segmentation from the current frame could be reused as a base for the next one, speeding up the segmentation process.

5.3.6 Weakly annotated data

Instead of starting from a coarse segmentation, a simple bounding box containing the region of interest to be segmented could be used as an input for the network, focusing the quality prediction, segmentation and subsequent refinement process to the given region and thus reducing the possible solution space.

5.3.7 Alternative refinement architectures

In the architecture presented in Section 4.6, only the U-Net was extended with the quality output and extra segmentation channel for refinement. The resulting architecture should be further analyzed, tested and possibly modified to better suit the desired goal. Other deep segmentation architectures should also be considered (*e.g.* DilatedNet).

5.3.8 Oracle as stopping criterion for other refinement processes

The predicted quality can not be reliably used as a stopping criterion for the own refinement process of a network when using backpropagation, but it may be useful as such for the refinement process of a different network, such as one for direct refinement.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, dec 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2644615. URL http://ieeexplore.ieee.org/document/7803544/.

[2] David Barber. Bayesian Reasoning and Machine Learning. *Machine Learning*, page 646, 2011. ISSN 9780521518147. doi: 10.1017/CBO9780511804779.

[3] K Bhargavi. A Survey on Threshold Based Segmentation Technique in Image Processing. *International Journal of Innovative Research and Development*, 3(12):234–239, 2014. ISSN 2278-0211.

[4] Jaime S. Cardoso and Maria J. Cardoso. Towards an intelligent medical system for the aesthetic evaluation of breast cancer conservative treatment. *Artificial Intelligence in Medicine*, 40(2):115 – 126, 2007. ISSN 0933-3657. doi: 10.1016/j.artmed.2007.02.007. URL http://www.sciencedirect.com/science/article/pii/S0933365707000206.

[5] Chang Wen Chen, Jiebo Luo, and Kevin J. Parker. Image segmentation via adaptive K-mean clustering and knowledge-based morphological operations with biomedical applications. *IEEE Transactions on Image Processing*, 7(12):1673–1683, 1998. ISSN 10577149. doi: 10.1109/83.730379.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. pages 1–14, 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2699184. URL http://arxiv.org/abs/1412.7062.

[7] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and Allan Halpern. Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). pages 1–5, 2017. ISSN 19458452. doi: 10.1109/ISBI.2018.8363547. URL http://arxiv.org/abs/1710.05006.

[8] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. ISSN 00220280. doi: 10.1080/01969727308546046.

[9] A. Fenster and B. Chiu. Evaluation of Segmentation algorithms for Medical Imaging. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 7186–7189. IEEE, 2005. ISBN 0-7803-8741-4. doi: 10.1109/IEMBS.2005.1616166. URL http://ieeexplore.ieee.org/document/1616166/.

[10] Kelwin Fernandes, Jaime S Cardoso, and Jessica Fernandes. Pattern Recognition and Image Analysis. 10255:243–250, 2017. doi: 10.1007/978-3-319-58838-4. URL http://link.springer.com/10.1007/978-3-319-58838-4.

[11] Kelwin Fernandes, Ricardo Cruz, and Jaime S Cardoso. Deep Image Segmentation by Quality Inference. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2018.

[12] Kelwin Fernandez and Carolina Chang. Teeth/Palate and Interdental Segmentation Using Artificial Neural Networks. pages 175–185. 2012. doi: 10.1007/978-3-642-33212-8_16. URL http://link.springer.com/10.1007/978-3-642-33212-8_16.

[13] Zhaoxia Fu and Liming Wang. Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm. pages 61–66. 2012. doi: 10.1007/978-3-642-35286-7_9. URL http://link.springer.com/10.1007/978-3-642-35286-7_9.

[14] Hongyang Gao, Hao Yuan, Zhengyang Wang, and Shuiwang Ji. Pixel Deconvolutional Networks. 2017. URL http://arxiv.org/abs/1705.06820.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

[16] Amit Gupta and Monica Lam. The weight decay backpropagation for generalizations with missing values. *Annals of Operations Research*, 78:165–187, 1998. ISSN 02545330. doi: 10.1023/A:1018945915940. URL http://link.springer.com/article/10.1023/A:1018945915940.

[17] Seyed Raein Hashemi, Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, Sanjay P. Prabhu, Simon K. Warfield, and Ali Gholipour. Asymmetric Similarity Loss Function to Balance Precision and Recall in Highly Unbalanced Deep Medical Image Segmentation. pages 1–10, 2018. URL http://arxiv.org/abs/1803.11078.

[18] Alexander Hernández-García and Peter König. Data Augmentation Instead of Explicit Regularization. *ICLR 2018 Conference*, pages 1–12, 2018.

[19] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. pages 1–18, 2012. ISSN 9781467394673. doi: arXiv:1207.0580. URL http://arxiv.org/abs/1207.0580.

[20] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

[21] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep Networks with Stochastic Depth. 2016. ISSN 0302-9743. doi: 10.1007/978-3-319-46493-0_39. URL http://arxiv.org/abs/1603.09382.

[22] Vladimir Iglovikov and Alexey Shvets. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. *ArXiv e-prints*, 2018. URL http://arxiv.org/abs/1801.05746.

[23] Kaggle Inc. Intel & MobileODT Cervical Cancer Screening, 2017. URL https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening.

[24] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. ISSN 0717-6163. doi: 10.1007/s13398-014-0173-7.2. URL http://arxiv.org/abs/1502.03167.

[25] Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:1175–1183, 2017. ISSN 21607516. doi: 10.1109/CVPRW.2017.156.

[26] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. ISSN 1063-6919. doi: 10.1109/CVPRW.2009.5206848. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206848.

[27] M. Kass, a. Witkin, D Tetzopoulos, and D. Terzopoulos. Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. ISSN 09205691. doi: 10.1007/BF00133570.

[28] Udayan Khurana, Horst Samulowitz, and Deepak Turaga. Feature Engineering for Predictive Modeling using Reinforcement Learning. 2017. URL http://arxiv.org/abs/1709.07150.

[29] Edward Kim and Xiaolei Huang. A Data Driven Approach to Cervigram Image Analysis and Classification. In *Macroscopic Pigmented Skin Lesion Segmentation and Its Influence on the Lesion Classication and Diagnosis*, pages 1–13. 2013. doi: 10.1007/978-94-007-5389-1_1. URL http://www.springerlink.com/index/10.1007/978-94-007-5389-1_1.

[30] Jung Uk Kim, Hak Gu Kim, and Yong Man Ro. Iterative deep convolutional encoder-decoder network for medical image segmentation. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 685–688, 2017. ISSN 1557170X. doi: 10.1109/EMBC.2017.8036917.

[31] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. pages 1–15, dec 2014. URL http://arxiv.org/abs/1412.6980.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 15:1097–1105, 2012. URL http://linkinghub.elsevier.com/retrieve/pii/S2212017314001224.

[33] Nikolas Lessmann, Bram van Ginneken, Pim A. de Jong, and Ivana Išgum. Iterative fully convolutional neural networks for automatic vertebra segmentation. (Midl):1–10, 2018. ISSN 16057422. doi: 10.1117/12.2292731. URL http://arxiv.org/abs/1804.04383.

[34] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5168–5177. IEEE, jul 2017. doi: 10.1109/CVPR.2017.549. URL http://ieeexplore.ieee.org/document/8100032/.

[35] Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8725 LNAI(PART 2):225–239, 2014. ISSN 16113349. doi: 10.1007/978-3-662-44851-9_15.

[36] Ilya Loshchilov and Frank Hutter. Fixing Weight Decay Regularization in Adam. 100, 2017. URL http://arxiv.org/abs/1711.05101.

[37] M. Mary Synthuja Jain Preetha, L. Padma Suresh, and M. John Bosco. Image segmentation using seeded region growing. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, volume 3034, pages 576–583. IEEE, mar 2012. ISBN 978-1-4673-0212-8. doi: 10.1109/ICCEET.2012.6203897. URL http://ieeexplore.ieee.org/document/6203897/.

[38] Teresa Mendonca, Pedro M. Ferreira, Jorge S. Marques, Andre R.S. Marcal, and Jorge Rozeira. PH2- A dermoscopic image database for research and benchmarking. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 5437–5440, 2013. ISSN 1557170X. doi: 10.1109/EMBC.2013.6610779.

[39] Senthilkumaran N and Vaithegi S. Image Segmentation By Using Thresholding Techniques For Medical Images. *Computer Science & Engineering: An International Journal*, 6(1):1–13, 2016. ISSN 2231329X. doi: 10.5121/cseij.2016.6101. URL http://www.aircconline.com/cseij/V6N1/6116cseij01.pdf.

[40] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. ISSN 0018-9472. doi: 10.1109/TSMC.1979.4310076. URL http://ieeexplore.ieee.org/document/4310076/.

[41] Dinesh D Patil, Sonal G Deore, and Ssgbcoet Bhusawal. Medical Image Segmentation: A Review. *Ijcsmc*, 2(1):22–27, 2013.

[42] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017. URL http://arxiv.org/abs/1712.04621.

[43] Sai Prasad Raya and Jayaram K. Udupa. Shape-Based Interpolation of Multidimensional Objects. *IEEE Transactions on Medical Imaging*, 9(1):32–42, 1990. ISSN 1558254X. doi: 10.1109/42.52980. URL https://ieeexplore.ieee.org/document/52980/.

[44] Mayra Z. Rodriguez, Cesar H. Comin, Dalcimar Casanova, Odemir M. Bruno, Diego R. Amancio, Francisco A. Rodrigues, and Luciano da F. Costa. Clustering Algorithms: A Comparative Approach. pages 1–31, 2016. URL http://arxiv.org/abs/1612.08388.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. ISBN 9783319245737. doi:

10.1007/978-3-319-24574-4_28. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a.

[46] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:815–823, 2015. ISSN 10636919. doi: 10.1109/CVPR.2015.7298682. URL https://arxiv.org/abs/1503.03832.

[47] Dinggang Shen, Guorong Wu, and Heung-il Suk. Deep Learning in Medical Image Analysis. *the Annual Review of Biomedical Engeneerring*, (March):221–248, 2017. ISSN 1523-9829. doi: 10.1146/annurev-bioeng-071516-044442. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5479722/.

[48] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. ISSN 01628828. doi: 10.1109/34.868688. URL https://ieeexplore.ieee.org/document/868688/.

[49] Pierre Soille. *Morphological Image Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-662-03941-0. doi: 10.1007/978-3-662-03939-7. URL http://link.springer.com/10.1007/978-3-662-03939-7.

[50] Abdel Aziz Taha, Allan Hanbury, and Oscar A. Jimenez del Toro. A formal method for selecting evaluation metrics for image segmentation. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 932–936. IEEE, oct 2014. ISBN 978-1-4799-5751-4. doi: 10.1109/ICIP.2014.7025187. URL http://ieeexplore.ieee.org/document/7025187/.

[51] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation. 2015. ISSN 21607516. doi: 10.1109/CVPRW.2016.60. URL http://arxiv.org/abs/1511.07053.

[52] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1058–1066, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/wan13.html.

[53] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? 27, 2014. ISSN 10495258. URL http://arxiv.org/abs/1411.1792.

[54] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015. URL http://arxiv.org/abs/1511.07122.