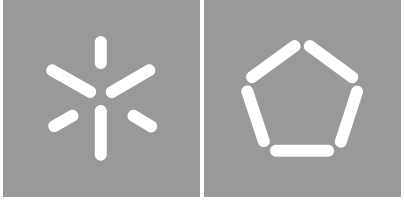
Universidade do Minho
Escola de Engenharia

Ana Patrícia Ribeiro Lopes

Study of Deep Neural Network architectures for medical image segmentation

Outubro de 2020

Study of Deep Neural Network architectures for medical image segmentation
Ana Patrícia Ribeiro Lopes
UMinho | 2020



Universidade do Minho
Escola de Engenharia

Ana Patrícia Ribeiro Lopes

Study of Deep Neural Network architectures for medical image segmentation

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica
Ramo de Eletrónica Médica

Trabalho efetuado sob a orientação do
Professor Doutor Carlos Alberto Batista Silva

Outubro de 2020

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Agradecimentos

Primeiramente, agradeço ao meu orientador, o Professor Carlos Silva, pelos conhecimentos transmitidos, paciência, disponibilidade e toda a dedicação.

Ao Adriano, estou grata por ter dispensado o seu tempo a ajudar-me e a aconselhar-me. Tenho a certeza que te espera um futuro brilhante.

Agradeço ao *lab of cool people* por todas as gargalhadas, jogos, bolos incríveis e caminhadas ao Bom Jesus. São, sem dúvida, pessoas espetaculares que nunca irei esquecer.

Aos meus amigos de Engenharia Biomédica, Ana, Cristiana, João, Luís e Margarida, agradeço os jantares e os bons momentos. Agradeço especialmente à minha compincha Joana por todas as horas de trabalho que passamos juntas e pelas conversas. Apesar de vos ter conhecido quase no final desta etapa, espero que a nossa amizade perdure.

Agradeço também às waffles, Ana, Leonor, Patrícia e Virgínia, por todo o apoio, companheirismo e amizade desde o início desta etapa. Sem vocês os maus momentos não seriam suportáveis e os bons teriam sido apenas medianos. A distância não nos separa!

Aos meus amigos de longa data, principalmente, Daniela, Gabriela, Pedro e Sara, estou grata pelas conversas, passeios e memórias felizes. É um prazer fazer parte das vossas vidas e espero que isso nunca mude!

Ao Tiago, obrigada por estares sempre disponível para ouvir as minhas lamentações e problemas e por conseguires melhorar todas as situações. Sem ti teria ficado completamente perdida. Tudo o que passamos juntos é inesquecível.

Por fim, agradeço aos meus pais pelo amor incondicional, apoio e sacrifícios. Vocês é que tornaram tudo possível, dando-me a oportunidade de frequentar o ensino superior e de terminar esta dissertação. Ao resto da minha família, sou grata por todo o carinho e pelas nossas conversas. Tambem agradeço aos meus animais de estimação, Vadia, Dama e Justa, por toda a fofura e companhia.

Statement of Integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Resumo

Estudo de arquiteturas de Deep Neural Networks para segmentação de imagem médica

A segmentação de imagens médicas desempenha um papel fundamental na área médica, pois permite realizar análises quantitativas usadas no rastreio, monitorização e planeamento do tratamento de inúmeras patologias. A segmentação manual é demorada e varia consoante o técnico. Assim, diversas abordagens automáticas têm sido propostas para a segmentação de imagens médicas e a maioria é baseada em *Deep Learning*. Estas abordagens tornaram-se especialmente relevantes após o desenvolvimento da *Fully Convolutional Network*. Neste método, as camadas totalmente ligadas foram eliminadas e foram incorporadas camadas de *upsampling*, permitindo que uma imagem seja segmentada de uma só vez. Atualmente, as arquiteturas desenvolvidas baseiam-se na FCN, sendo a U-Net uma das mais populares.

O objetivo desta dissertação é estudar arquiteturas de *Deep Learning* para a segmentação de imagens médicas. Foram selecionadas duas tarefas desafiantes e muito distintas, a segmentação de vasos retinianos a partir de imagens do fundo da retina e a segmentação de tumores cerebrais a partir de imagens de MRI. As arquiteturas estudadas neste trabalho são baseadas na U-Net, devido às elevadas performances que esta obteve em diversas tarefas de segmentação médica.

Os modelos desenvolvidos para segmentação de vasos retinianos e de tumores cerebrais foram testados em bases de dados públicas, DRIVE and BRATS 2017, respetivamente. Vários estudos foram realizados para a primeira tarefa, nomeadamente, comparação de operações de downsampling, substituição de uma camada de *downsampling* por convoluções dilatadas, incorporação de uma camada composta por RNNs e aplicação de técnicas de aumento de dados na fase de teste. Na segunda tarefa, três modificações foram avaliadas, a incorporação de *long skip connections*, a substituição de convoluções *standard* por convoluções dilatadas e a substituição de uma camada de *downsampling* por convoluções dilatadas.

Quanto à segmentação de vasos retinianos, a abordagem final obteve *accuracy*, sensibilidade e AUC de 0.9575, 0.7938 e 0.9804, respetivamente. Esta abordagem consiste numa U-Net, que contém uma convolução *strided* como operação de *downsampling* e convoluções dilatadas com *dilation rate* de 3, seguida de uma técnica de aumento de dados em fase de teste, executada por uma ConvLSTM. Em relação à segmentação de tumores cerebrais, a bordagem proposta obteve *Dice* de 0.8944, 0.8051 e 0.7353 e HD_{95} de 6.79, 8.34 e 4.76 para o tumor completo, região central e região contrastante, respetivamente. O método final consiste numa arquitetura DLA com uma *long skip connection* e convoluções dilatadas com *dilation rate* de 2. As duas abordagens são competitivas com os métodos do estado da arte.

Palavras chave: *Fully Convolutional Network*, *Recurrent Neural Network*, Segmentação de Imagem, Tumor Cerebral, Vasos Retinianos

Abstract

Study of Deep Neural Network architectures for medical image segmentation

Medical image segmentation plays a crucial role in the medical field, since it allows performing quantitative analyses used for screening, monitoring and planning the treatment of numerous pathologies. Manual segmentation is time-consuming and prone to inter-rater variability. Thus, several automatic approaches have been proposed for medical image segmentation and most are based on Deep Learning. These approaches became specially relevant after the development of the Fully Convolutional Network. In this method, the fully-connected layers were eliminated and upsampling layers were incorporated, allowing one image to be segmented at once. Nowadays, the developed architectures are based on the FCN, being U-Net one of the most popular.

The aim of this dissertation is to study Deep Learning architectures for medical image segmentation. Two challenging and very distinct tasks were selected, namely, retinal vessel segmentation from retinal fundus images and brain tumor segmentation from MRI images. The architectures studied in this work are based on the U-Net, due to high performances obtained in multiple medical segmentation tasks.

The models developed for retinal vessel and brain tumor segmentation were tested in publicly available databases, DRIVE and BRATS 2017, respectively. Several studies were performed for the first segmentation task, namely, comparison of downsampling operations, replacement of a downsampling step with dilated convolutions, incorporation of a RNN-based layer and application of test time data augmentation techniques. In the second segmentation task, three modifications were evaluated, specifically, the incorporation of long skip connections, the substitution of standard convolutions with dilated convolutions and the replacement of a downsampling step with dilated convolutions.

Regarding retinal vessel segmentation, the final approach achieved accuracy, sensitivity and AUC of 0.9575, 0.7938 and 0.9804, respectively. This approach consists on a U-Net, containing one strided convolution as downsampling step and dilated convolutions with dilation rate of 3, followed by a test time data augmentation technique, performed by a ConvLSTM. Regarding brain tumor segmentation, the proposed approach achieved Dice of 0.8944, 0.8051 and 0.7353 and HD_{95} of 6.79, 8.34 and 4.76 for complete, core and enhanced regions, respectively. The final method consists on a DLA architecture with a long skip connection and dilated convolutions with dilation rate of 2. For both tasks, the proposed approach is competitive with state-of-the-art methods.

Keywords: Fully Convolutional Network, Recurrent Neural Network, Image Segmentation, Brain Tumor, Retinal Vessels

Contents

Acronyms

1D One-dimensional. 35

2D Two-dimensional. 10, 27, 35, 37, 56

3D Three-dimensional. 36, 37, 56, 73

Acc Accuracy. 42, 44–47, 50, 54, 55, 58–61

Adam Adaptive Moment Estimation. 9, 42, 65

AMD Aged-related Macular Degeneration. 28, 38

ANN Artificial Neural Network. 3, 4, 19

AUC Area Under the ROC Curve. 43, 44, 46, 47, 50, 54, 55, 58–61

BN Batch Normalization. viii, 15, 16, 19, 41, 64, 65

BRATS Brain Tumor Segmentation Challenge. xii, 63, 73

CNN Convolutional Neural Network. 10, 12, 14, 16–19, 21, 24, 25, 31, 32, 35–37, 39, 47, 48

ConvLSTM Convolutional Long Short-Term Memory. ix, 21, 25, 57, 58, 60, 61

DCB Dilated Convolutional Block. ix, 40–43, 45, 47, 49, 65, 74

DLA Deep Layer Aggregation. viii, 18, 19, 25, 62, 63, 68, 69, 74

DR Diabetic Retinopathy. 28, 29

DRIVE Digital Retinal Images for Vessel Extraction. xi, 40, 44, 46, 49, 50, 54, 57, 59, 61, 63

DTPA Diethylenetriamine Penta-acetic Acid. 33

FCN Fully Convolutional Network. 17, 18, 23, 31, 32, 35, 36, 38, 48, 62, 73

FLAIR Fluid-Attenuation Inversion Recovery. 33, 34, 38, 63

FN False Negative. ix, 42, 45, 46, 50, 51, 59, 60

FOV Field of View. 40, 42, 50, 58

FP False Positive. ix, 42, 44, 45, 50, 51, 54, 59, 60, 70–72

GRU Gated Recurrent Unit. 22, 25, 49, 50

HD Hausdorff Distance. 68

HD$_{95}$ 95% quantile of Hausdorff Distance. 65, 68–74

HDA Hierarchical Deep Aggregation. viii–x, 18, 19, 62, 64, 66–68, 74

HGG High-Grade Glioma. 33, 34, 38, 63

IDA Iterative Deep Aggregation. ix, x, 19, 62, 64, 66–68, 74

IRMA Intraretinal Microvascular Abnormality. 28

LGG Low-Grade Glioma. 33, 34, 38, 63

LSTM Long Short-Term Memory. 21, 22, 25

MCC Mathews Correlation Coefficient. 43, 44, 46, 47, 50, 54, 55, 58–61

MRI Magnetic Resonance Imaging. 10, 11, 26, 33–37, 63

NP Number of Parameters. xi, 44

RANO Response Assessment in Neuro-Oncology. 34

RCB Regular Convolutional Block. ix, 41, 42, 49

ReLU Rectified Linear Unit. viii, 15, 19, 24, 41, 64, 65

RNN Recurrent Neural Network. v, vi, 20–22, 48–55

ROC Receiver Operating Characteristic. 43, 50, 58

RPE Retinal Pigment Epithelium. 26, 28

SD Standard Deviation. ix, 52, 53

Sens Sensitivity. xi, 42–47, 50, 54, 55, 58–61, 65, 66, 69–74

SGD Stochastic Gradient Descent. 8, 9

Spec Specificity. 42–44, 46, 47, 50, 54, 55, 58–61, 70

T1 T1-weighted. 33, 34, 38, 63

T1c T1-weighted with contrast enhancement. 33, 34, 38, 63

T2 T2-weighted. 33, 34, 38, 63

TN True Negative. 42

TP True Positive. 42, 46, 59, 69, 71, 72

VEGF Vascular Endothelial Growth Factor. 28, 29

WHO World Health Organization. 32, 33

List of Figures

List of Tables

Chapter 1

Introduction

This chapter starts by presenting the motivation behind the work developed throughout this dissertation. Next, the main objectives and contributions of this work are described. Ultimately, the structure of the remaining dissertation is presented.

1.1 Motivation

Medical image segmentation is indispensable to perform quantitative analyses that are extremely relevant for screening, monitoring and planning the treatment of numerous pathologies. This task, when done manually, requires a long amount of time, it's tedious and the final result may considerably vary between raters, even when they are experienced. Taking this into account, several automatic approaches have been proposed over the last few years for medical image segmentation.

In Deep Learning, the development of Convolutional Neural Networks [1] was crucial for, medical and non-medical, image segmentation tasks. This algorithm was specifically designed to process structured data, such as images, using a mathematical operation designated as convolution. This kind of network contains alternating convolutional and downsampling layers that are followed by fully-connected layers. The stacking of convolutional layers allows the network to extract increasingly complex features from the data. Then, the last layers of the network utilize these features to identify the structures of interest. This method showed to be efficient, obtaining superior performances when compared with its predecessor. Sometime ago, the development of Fully Convolutional Networks [2] further facilitated the automatic segmentation of images. The replacement of fully-connected layers by convolutional layers and the inclusion of upsampling layers allowed the network to operate on images with different sizes and classify all pixels of an image at once. Furthermore, in this method, features with different levels of complexity were combined though the usage of skip connections, improving the segmentation detail and, consequently, the model's performance. Since then, novel architectures based on the work of Long et al. [2] are constantly proposed for medical image segmentation, being U-Net [3] one of the most popular.

1.2 Objective

Medical image segmentation comprises a vast range of tasks with very distinct characteristics. In this work, two challenging medical segmentation tasks were selected, namely, retinal vessel and brain tumor segmentation from retinal fundus images and MRI images, respectively. In the first task, the structures of interest significantly vary in terms of shape, length and width, thin vessels being the most difficult to detect. The retinal images present complex structures, caused by the branching and crossing of vessels. Retinal vessel segmentation consists in a binary task with imbalanced classes. Regarding the second task, brain tumors can highly vary in size and appearance. They also have an irregular shape. Additionally, a tumor can be located in any part of the brain. Lastly, brain tumor segmentation consists in a multiclass task in which the classes have different distributions.

The original U-Net, proposed by Ronneberger et al. [3], have been applied to numerous medical image segmentation tasks, including retinal vessel and brain tumor segmentation, and it achieved high performances. The modification of this architecture according to the characteristics of each problem may facilitate the segmentation task and lead to superior performances.

Thus, the main goal of this dissertation is to study Deep Learning architectures, based on the U-Net [3], for the tasks of retinal vessel and brain tumor segmentation. The development of this work requires knowledge in multiple areas, namely, Deep Learning, programming techniques and health sciences.

1.3 Contributions

To the best of the author's knowledge, some methods developed throughout this dissertation can be considered original contributions, namely:

- Study of the U-Net architecture for retinal vessel segmentation, comparing different downsampling operations and analyzing the effects of replacing a downsampling step by convolutions with enlarged receptive field. The last topic is especially important since the reduction of resolution may lead to the loss of small details, in other words, thin vessels.
  - Initial phase of the study can be found in the article of the IEEE sixth Portuguese Meeting on Bioengineering [4].
- Application of recurrent neural networks to features extracted by a pre-trained U-Net. This allows retrieving long term dependencies between pixels within a patch and, consequently, utilize both local and global information for retinal vessel segmentation.
- A test time data augmentation technique, in which a neural network learns how to merge the multiple probability maps of the same example into the final segmentation, taking into account the context of each pixel.

- Study of the Deep Layer Aggregation architecture for brain tumor segmentation, evaluating the importance of low level information and the effects of increasing the network's receptive field, through the incorporation of long skip connections and dilated convolutions, respectively.

1.4 Structure of the Dissertation

The remaining dissertation is organized in seven chapters. Chapter 2 contains the theoretical fundamentals of deep learning, focusing on convolutional and recurrent neural networks, the algorithms employed for medical image segmentation in this work. Chapter 3 presents the clinical context of retinal vessels and brain tumors that includes the description of the structures, the importance of segmenting and the respective difficulties. The state-of-art methods for retinal vessel and brain tumor segmentation are also presented in this chapter. In chapter 4, the U-Net architecture is studied for retinal vessel segmentation. Specifically, two types of downsampling operation are compared and the replacement of a downsampling step by dilated convolutions is analyzed. In chapter 5, a layer composed by recurrent neural networks is incorporated into a pre-trained model, proposed in the previous chapter, for retinal vessel segmentation. The effect of this layer directly on the output probability maps is also examined. In chapter 6, test time data augmentation techniques are applied to the best preforming model in chapter 4 for the task of retinal vessel segmentation. Chapter 7 contains the study of Deep Layer Aggregation architecture for brain tumor segmentation. This study incorporated several modifications, namely, insertion of long skip connections, substitution of standard convolutions by dilated convolutions and replacement of a downsampling step by dilated convolutions. Finally, chapter 8 presents the main conclusions and future lines of research related with the developed work.

Chapter 2

Deep Learning

In this chapter, the theoretical foundations of deep learning are introduced. As deep learning is a machine learning technique, this chapter starts by describing the basic principles of machine learning. Then, feedforward neural networks and their learning process are presented. Finally, Convolutional Neural Networks, Fully Convolutional Networks and Recurrent Neural Networks, which were used for image segmentation in the scope of this work, are described. This chapter also discusses techniques for regularization and parameter initialization.

2.1 Machine Learning Basic Notions

A machine learning algorithm is able to learn from data how to perform a given task [5].

There are many machine learning tasks. The one of interest in the scope of this work is classification, which is one of the most common machine learning tasks. A classification task can be mathematicaly defined as a function $f : \mathbb{R}^n \rightarrow \{1, ..., k\}$. The input x, a sample of the data, is a vector composed by a group of n features, which are quantitative measures that represent the data in question. The unknown function f assigns each vector x to a category y, one of the k possible classes or labels. A classification task can be designated as binary and multiclass when $k = 2$ and $k > 2$, respectively [5, 6].

The goal of the machine learning algorithm or model is to learn a function $\hat{f}$, an estimation of f, using the data. Instead of directly predicting a class, the output of the model is usually the probability distribution over classes, $\hat{y}$. The final label of a given sample x is obtained with the maximum a posteriori estimate and corresponds to the most probable label [5, 6].

In addition to data, the learning algorithm needs a performance measure, usually an error measure, to evaluate the model. Throughout the learning process, training, the parameters θ of the function $\hat{f}$ are updated in order to improve that performance. The data observed by the model during training is denominated as training set. The performance of the algorithm is then measured on new data, test set. This way, it is possible to know how well the algorithm will behave in a real life application [5].

Most machine learning algorithms have settings, denominated hyperparameters. During training, the hyperparameters control the algorithm's behavior and their value is not optimized. A subset of the training set, validation set, is used to choose the hyperparameters of the model. The performance of

the model is measured on the validation set, during training, which allows guidding the selection of the hyperparameters. Basically, the selection process can be made by trail and error. The validation set isn't used to train the machine learning algorithm, so it's a set of unobserved data, just like the test set [5].

2.1.1 Types of Learning

The machine learning methods can be divided into two groups, unsupervised and supervised [5, 6].

In unsupervised learning, several samples of a random vector x are used to learn the probability distribution $p(x)$ or interesting properties of that distribution. Thus, unsupervised methods use image's features to divide its pixels into different groups, for example, vessel and non-vessel. This class of methods don't make use of the label associated to each pixel [5, 6].

In contrast, supervised methods require several samples of a vector x and its correspondent label y. The algorithm learns how to predict y from x, by estimating the probability distribution $p(y|x)$. The supervised methods require pre-labeled data. In other words, a classifier needs both data's features and label to learn a function capable of distinguishing the various groups. The deep learning models, also known as artificial neural networks (ANNs), used throughout this work are included in supervised methods [5, 6].

2.1.2 Generalization and Capacity

A machine learning problem is more than an optimization problem. The aim of machine learning is to create a model with a good performance not only on the training set, but especially on the test set. In other words, the machine learning algorithm must be able to generalize from the training set to samples of data not used for training. The error measure on unobserved inputs is denominated as test error or generalization error. During training, the training error is minimized, which indirectly reduces the test error. In practice, the test error of a model is equal to or greater than the training error [5].

Hereupon, two factors indicate how well a machine learning algorithm will perform, its ability to minimize the training error and its ability to make the difference between training and test error small [5].

These factors are related with two problems in machine learning, underfitting and overfitting, which are the causes of poor performance. Underfitting happens when the training error obtained by the model is not low enough. Overfitting occurs when the gap between the training and test error of the model is too large, i. e., the model isn't able to generalize [5].

In Figure 1, the relation between these two problems and model's capacity is shown. The capacity of a model can be informally defined as the variety of functions that a model can fit, being strongly correlated with the number of model parameters. On one hand, a model with low capacity is likely to underfit. On the other hand, a model with high capacity might memorize exclusive properties of the training set and, consequently, overfit. Thus, the capacity of the model should be chosen according to the complexity of the task. In other words, a model with greater capacity should be used to learn a more complex task [5].

The duration of the learning process can also cause underfitting or overffiting. If the training duration is short, the model won't be able to learn enough information from the training set, resulting in underfitting.

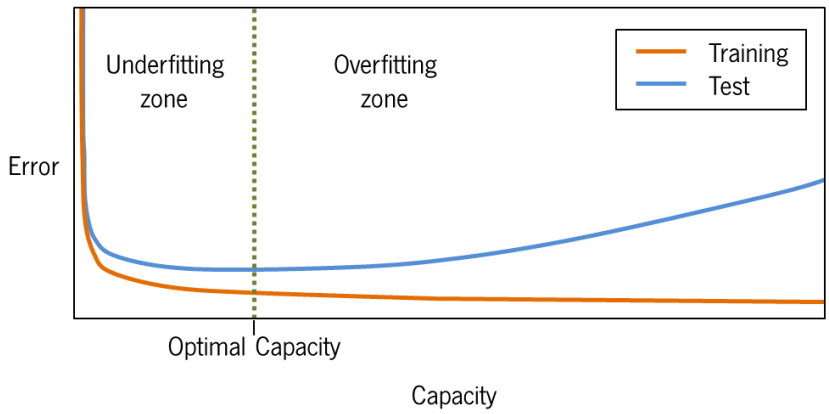


Figure 1: Variation of training and test errors as a function of model's capacity. Based on Goodfellow et al. [5].

Overfitting can be observed when the training is too long. The duration of trainning corresponds to the number of iterations over te training set, also denominated as epochs [5].

Lastly, overfitting can also be related with the number of training samples. The model's ability to generalize will increase together with the amount of samples used for training [5].

2.2 Feedforward Neural Networks

An ANN is categorized into a feedforward neural network when the information flows only in one direction, from the input to the output of the model. A deep feedforward network can be seen as a function, $f(x)$, composed by many different and simpler functions, connected in a chain. Each of this functions correspond to a layer of the network. Then, if the network has three layers (Figure 2), $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$ and $f^{(l)}$ represent the transformation executed in each layer l. The input layer, $l = 0$, doesn't perform any computation over the data. The last layer of the network, in this concrete case $f^{(3)}$, is called the output layer. The remaining layers correspond to hidden layers, because their desired output is not specified by the data. The number of layers determines the depth of the network [5].

The most common example of feedforward neural networks are the multilayer perceptrons, also denominated as fully-connected neural networks. The base element of this type of network is denominated as perceptron or neuron. The operations performed in each neuron are the following,

$$z = \sum_{i=1}^{n_i} w_i x_i + b \tag{1}$$

$$h = g(z) \tag{2}$$

First, the n_i input units, contained in a vector $x \in \mathbb{R}^{n_i}$, are linearly combined using the weights $w \in \mathbb{R}^{n_i}$ and bias $b \in \mathbb{R}$. Then, a non-linear activation function $g(z)$ is applied to obtain the output unit $h \in \mathbb{R}$. A layer is usually composed by multiple neurons and, consequently, multiple output units. In each layer, all output units are connected with all input units trough a weight matrix. Thereby, each layer is designated

as fully-connected and is defined as,

$$z = wx + b \tag{3}$$

$$h = g(z) \tag{4}$$

where $x \in \mathbb{R}^{n_i}$, $w \in \mathbb{R}^{n_i \times n_o}$, $b, z, h \in \mathbb{R}^{n_o}$ and n_o is the number of output units. The parameters of the equation 3, weight matrix w and biases b, are the ones learned during training. The non-linear activation function don't usually contain learnable parameters and it's individually applied to each element of z [5, 7].

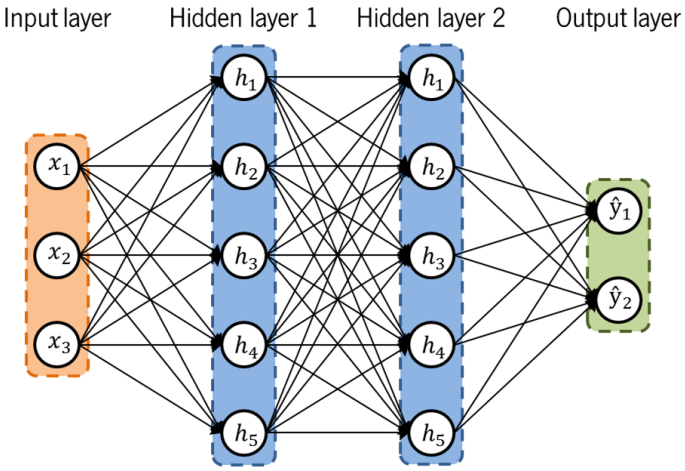


Figure 2: Example of a fully-connected network with two hidden layers. In this network, each hidden layer contains five neurons and three features are utilized to perform binary classification.

Another important concept to retain is architecture. The architecture of a network is its overall structure. This includes the number and type of layers, their settings and how they are connected to each other. The fundamental architecture of a network is highly correlated with the task in hand [5].

2.2.1 Training

After choosing the model, the learning process requires an error measure and an algorithm capable of optimizing the learnable parameters, just like any machine learning algorithm [5].

The error measure is usually denominated as loss function or cost function and it's applied to the output of the network. Then, an optimizer uses the gradient of the cost function to update the parameters and, consequently, increase the performance [5].

2.2.1.1 Gradient-based optimization

In general terms, an optimization problem consists of minimizing or maximizing some function $f(x)$ by altering the input x. The derivative or gradient of this function is denoted as $\frac{df(x)}{dx}$. The gradient at a point x corresponds to the slope of the function at that point. This way, the gradient specifies what change should be made to x in order to improve the output [5].

In the scope of this work, the minimization of a function is the relevant operation. In Figure 3, it's shown that, to reduce a given function, x must be iteratively moved in small steps according to the opposite

sign of the gradient. This technique is known as gradient descent. In this method, the size of the step is determined not only by the absolute value of the gradient, but also by a preset scalar. It should also be noted that the function presented in Figure 3 is convex, so the optimal solution, which corresponds to the global minimum, is easily found [5].

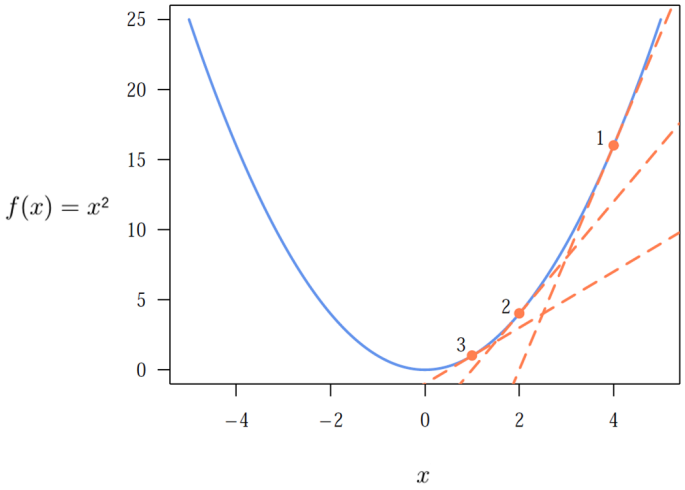


Figure 3: Gradient descent algorithm iteratively applied to a quadratic function. The dashed orange lines indicate the slope of the function at each orange point. The start point is marked as 1.

In deep learning, the functions are much more complicated than a quadratic function and have multiple inputs. Additionally, the nonlinearity of the neural networks makes the loss function non-convex, i.e., the loss function contains many local minimums and saddle points[1]. In these conditions, the use of a gradient-based optimizer doesn't ensure global convergence. The presence of local minimums was, at first, seen as a problem, but, as the depth of the network increases, the majority of local minimums correspond to a low cost function value. Therefore, during the learning of the algorithm, the cost function is guided to a very low value. In practice, the value of the loss function at the end of training is not necessarily a global or a local minimum. Furthermore, the initial values of the parameters affect the process and deeper models tend to be harder to optimize [5].

2.2.1.2 Loss function

In this work, the model defines a probability distribution $p(y|x;\theta)$. This is ensured by the output layer of the network, softmax (section 2.3.6). Thus, the most common used loss function in a classification task is the cross-entropy loss, which is defined as

$$J(\theta) = L(\hat{y}, y) = -\sum_{i=1}^{k} y_i \log(\hat{y}_i) \tag{5}$$

where k refers to the number of classes, $\hat{y}$ is the prediction of the network and y is the desired label represented as an one-hot vector. The cross-entropy loss is also denominated as the negative log likelihood

[1] Points with zero derivative that have lower and higher neighboring points.

and its minimization causes maximum likelihood estimation [5, 6].

Other well-known loss function is the mean squared error loss. Models using this loss function usually suffer from slow learning and saturation, because it gives rise to low gradients. When compared with the mean squared error loss, the cross-entropy loss achieves better performances, principally when applied to a softmax output [5].

The cost function may include additional terms, not necessarily related with the prediction error [5]. In section 2.6.1, there's an example of a term usually added to the cost function, which is a parameter norm penalty.

2.2.1.3 Backpropagation

During training, the information propagates forward, through the network, from the input x to the output $\hat{y}$, until producing a scalar cost $J(\theta)$. This flow of information is called forward propagation. The back-propagation algorithm allows the information to propagate backwards, from the cost and through the hidden layers, in order to compute the gradient [5].

The back-propagation algorithm only computes the gradient of the cost function with respect to the parameters, $\nabla_\theta J(\theta)$. Then, other algorithm is responsible for the learning itself, i.e., for updating the parameters of the network [5].

Basically, the back-propagation algorithm computes the chain rule of calculus, in a way that is highly efficient. The chain rule of calculus is used to calculate the gradients of composite functions, in other words, complex functions composed by simpler functions with known derivative. If $y = g(x)$, and $z = f(y) = f(g(x))$, the chain rule states that

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} \tag{6}$$

This can be extended to vectors. In this case, $x \in R^n$, $y \in R^m$ and z remains a scalar. Thus, the chain rule can be computed as [5]

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j}\frac{\partial y_j}{\partial x_i} \tag{7}$$

In the context of deep learning, the chain rule of calculus is applied recursively, to each layer of the network, from the last to the first layer. This allows computing the gradient of the cost function with respect to each layer's parameters. The first operation in back-propagation is defined as

$$g \leftarrow \nabla_{\hat{y}} J = \nabla_{\hat{y}} L(\hat{y}, y) \tag{8}$$

and obtains the gradient of the cost function in relation to its input, the probability distribution $\hat{y}$, which corresponds to the network's output [5]. In practice, this operation is usually made for more than one training sample at once, what will be discussed in the next section.

As already mentioned, the last layer of the network is an activation layer, which do not contain learnable parameters. In the presence of an activation layer h_l, the gradient is calculated in relation to its input,

which in this case corresponds to the output of a learnable layer [5],

$$g \leftarrow \nabla_{z_{l-1}} J = g \odot \nabla_{z_{l-1}} h_l \tag{9}$$

The operator $\odot$ represents the element-wise multiplication of two matrices.

In the presence of a layer with learnable parameters, two computations must be made. The gradient is calculated with respect to the parameters, which is the main objective, as

$$\nabla_{\theta_{l-1}} J = g \odot \nabla_{\theta_{l-1}} z_{l-1} \tag{10}$$

The gradient is also computed with respect to layer's input, commonly the output of an activation layer,

$$g \leftarrow \nabla_{h_{l-2}} J = g \odot \nabla_{h_{l-2}} z_{l-1} \tag{11}$$

allowing the propagation of the gradient to previous layers [5].

The equations 9, 10 and 11 are applied recursively until the gradient of all layer parameters is calculated. It should be noted that the output of a layer may be used as input in multiple other layers of the network. In this case, the gradients arriving from different paths are summed [5].

2.2.1.4 Optimizer

As already mentioned, a training set with large size is required to obtain a model with good generalization. Nevertheless, the use of a large training set has a downside, it increases the computational cost of the optimization process [5].

The cost function of the training set can be decomposed into a sum of loss functions, applied to the M training samples. This way, in order to update the parameters of the network, the gradient descent algorithm demands computing

$$\nabla_\theta J(\theta) = \frac{1}{M} \sum_{i=1}^{M} \nabla_\theta L(\hat{y}^{(i)}, y^{(i)}) \tag{12}$$

The equation shows that the gradient must be calculated individually for each training sample. The equation also shows that the parameters of the model are updated only once per iteration over the entire training set. This way, for an extremely large training set, the time needed to do a single update, with the gradient descent algorithm, becomes prohibitively long [5].

The Stochastic Gradient Descent (SGD), an extension of the gradient descent algorithm, overcomes the issue expressed previously. The gradient of the training set can be approximately estimated by using a small set of training samples. In this algorithm, a minibatch containing m samples is used on each step to update the parameters. The training examples are continuously sampled from the training set, until all examples are used. Thereby, during an epoch, the learnable parameters are updated multiple times. In

practice, the backpropagation algorithm computes the gradient as

$$\nabla_\theta J(\theta) = \frac{1}{m}\nabla_\theta \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) \tag{13}$$

and, then, the SGD algorithm updates the parameters according to the following equation

$$\theta = \theta - \epsilon g \tag{14}$$

where ϵ is the learning rate. The minibatch size and the learning rate are two hyperparameters of a network training [5].

The training is dependent of ϵ, so its value should be set carefully. When ϵ is too hight, learning becomes severely unstable. Through training, the cost function contains violent oscillations, and its value often increases significantly. When ϵ is set too low, the learning process is slow and may stabilize. In this case, the cost function decreases slowly and may get stuck with a high value. In practice, the learning rate is usually not fixed, it's reduced throught training, according to a pre-defined schedule [5].

Regardless of learning rate, SGD can retard the learning of the neural network. When this delay is particularlly related with noisy gradients, consistently small gradients or high curvature of the cost function's surface, the momentum [8] algorithm can be used to speed up learning [5].

A new variable, velocity v, is introduced in the basic SGD and gives information about the direction and speed at which the parameters are updated. At each update, the algorithm accumulates the current gradient in v, which ensures that the parameters continue to move in the direction of the former gradients. The velocity is updated as follows,

$$v = \alpha v - \epsilon\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) \tag{15}$$

where $\alpha \in [0, 1)$ is a hyperparameter denominated as momentum. As indicated in equation 15, α quantifies the contribution of previous gradients to the current direction. This contribution is enlarged when α is greater. This hyperparameter can also be altered during tranning, usually from a low value to a high value. In the momentum learning algorithm, the update rule is given by [5]

$$\theta = \theta + v \tag{16}$$

As already mentioned, the training of the network, and, consequently, its performance, is extremely conditioned by the learning rate. To ease this issue, some algorithms automatically adapt the learning rate through training. These methods are denominated as algorithms with adaptive learning rates. In these methods, the learning rate is individually adapted for each model parameter. One of these algorithms is the adaptive moment estimation (Adam) [9], a stochastic optimizer based on adaptive lower-order moments. In Adam, two estimates are computed, the firs-order moment and the second-order moment of the gradients. Each of this moments accumulates respectively the gradient and the squared gradient. Additionally, the first moment can be seen as the, previously described, momentum. The estimates of the

first and second moments of the gradients are respectivly updated as follows

$$s \leftarrow \rho_1 s + (1 - \rho_1) g \tag{17}$$

$$r \leftarrow \rho_2 r + (1 - \rho_2) g \odot g \tag{18}$$

where $\rho_1, \rho_2 \in [0, 1)$ are decay rates. The algorithm also applies an initialization bias correction technique to both moment estimates, since they are initialized as zero,

$$\hat{s} \leftarrow \frac{s}{1 - \rho_1} \tag{19}$$

$$\hat{r} \leftarrow \frac{r}{1 - \rho_1} \tag{20}$$

Then, the update rule is is given by [5]

$$\theta = \theta - \epsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta} \tag{21}$$

where δ is a constant utilized for numeric stabilization.

2.3 Convolutional Neural Networks

Convolutional neural networks [1] are a subgroup of feedforward neural networks that contain one or more convolutional layers (Figure 4). While a fully-connected layer performs a general matrix multiplication, a convolutional layer, as indicated by its name, performs a mathematical operation denominated as convolution. This operation is ideal to process data with a grid-like topology, regardless of the number of dimensions [5]. Electroencephalogram signals, retinal fundus images and MRI images are examples of data organized in a grid with 1, 2 and 3 dimensions, respectively. Thus, this type of network is commonly used for tasks of image classification and segmentation. In the scope of this work, the task of interest is image segmentation, or semantic segmentation, and it refers to the classification of every pixel in an image.

In addition to convolutional layers, CNNs contain other layers that are described below. The characterization of the layers will be done taking into account a 2D input.

2.3.1 Convolutional layer

Each convolutional layer is composed by a set of kernels, also denominated as filters, much smaller than the input. The kernels are convolved over the input and originate feature maps, layer's output. The input of this layer may be the input image, when referring to the first layer, or feature maps, when referring to other convolutional layers. The input image may have more than one channel, i.e, each element of the grid may be characterized by a vector of features, per example, multiple sequences of MRI are used for tumor segmentation. Similarly, a convolutional layer contains multiple kernels, applied in parallel and

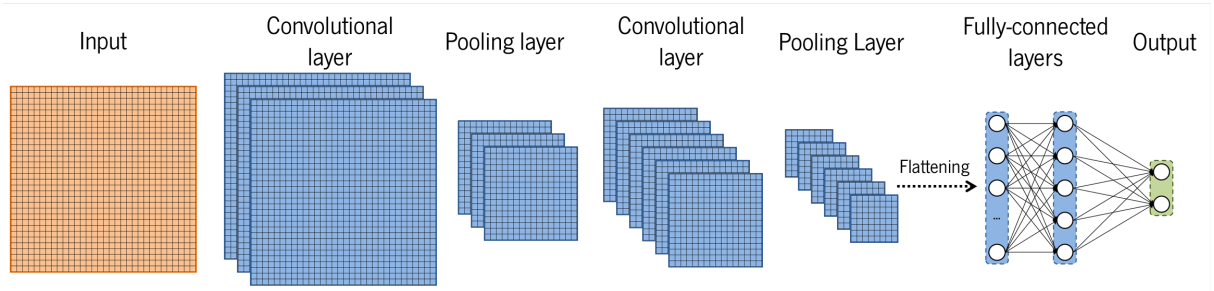


Figure 4: Example of a convolutional neural network composed by two convolutional layers, each one followed by pooling, and two fully-connected layers. Each convolutional layer is usally combined with batch normalization and an activation function.

responsible for extracting multiple features. This way, each kernel originates only one channel, or feature map, of the output and, to that, all input channels have a role. Associated with each kernel, there's a bias that is added after computing each convolution. Considering an output with only one channel, a convolutional layer can be defined as,

$$Z = b + \sum_{i=1}^{n_i} I_i * K_i \tag{22}$$

where I correponds to the input with n_i channels, K is the kernel, b represents the bias and $*$ denotes the convolutional operation. The result of this operation at a given location, row j and column k, is computed as,

$$Z_{j,k} = \sum_{i=1}^{n_i} \sum_{m=-\frac{k_1-1}{2}}^{\frac{k_1-1}{2}} \sum_{n=-\frac{k_2-1}{2}}^{\frac{k_2-1}{2}} I_{i,j-m,k-n} \times K_{i,m,n} \tag{23}$$

where k_1 and k_2 are the spatial dimensions of the kernel and $K_{i,0,0}$ corresponds to the center of the kernel at channel i. The convolutional operation is often implemented as cross-correlation as shown in Figure 5, i.e., without flipping the kernel. Considering the number of output feature maps as n_o, the number of learnable parameters associated to a convolutional layer results from the sum of the number of kernels' weights, given by $k_1 \times k_2 \times n_i \times n_o$, with the number of biases, whose value is equal to n_o [5].

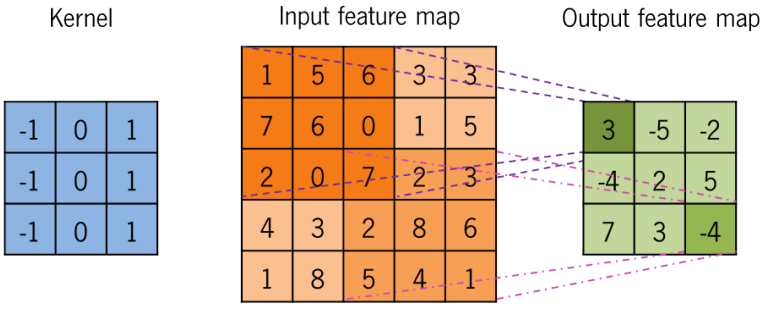


Figure 5: Convolving a 3×3 kernel over a single channel input.

Regarding the treatment of border units, convolution can be denominated in two ways. In *valid* con-

volution, the operation is only computed if the kernel is totally contained inside the input, as observed in Figure 6a. In this case, every time a convolution is applied, the size of the feature map reduces by $k - 1$ units. This can limit the number of layers in the network, principally if large kernels are employed. In *same* convolution, zero-padding is added to the input in such a way that the size of the output is equal to the size of the input. Thus, the network is not limited in terms of number of layers that can use. However, the units near the border of the output feature map can be compromised by the introduction of 'false' information [5]. The *same* convolution is illustrated in Figure 6b.

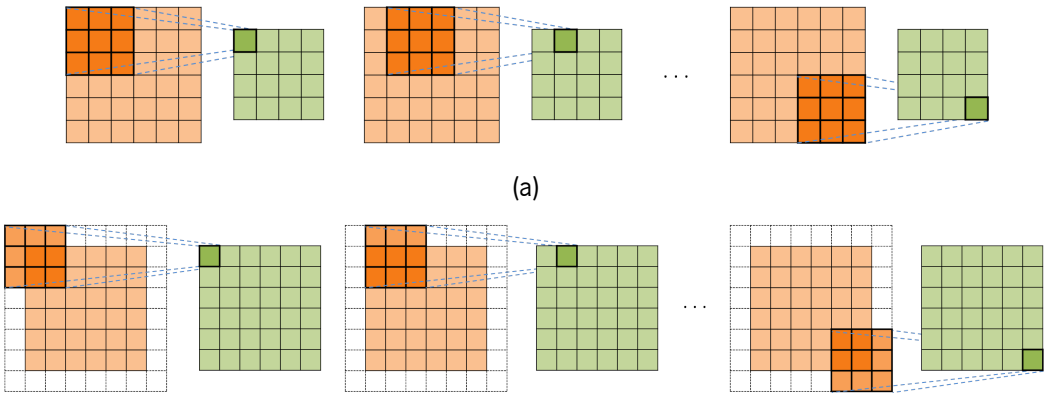


Figure 6: Regarding the treatment of the input borders, the convolutional operation can be denominated as (a) *valid* convolution and (b) *same* convolution. Based on Dumoulin and Visin [10].

Receptive Field A convolutional layer has a local receptive field that corresponds to the region of the input that affects a unit of the output and, consequently, to the kernel size. When multiple layers are sucessively stacked, the receptive field of the network is progressively enlarged since an output unit indirectly depends on a larger portion of the input image [5].

Feature extraction The convolutional layers exploit the correlation between nearby pixels, being responsible for the extraction of local features. The depth of the CNN is related to the complexity of the obtained features. The first convolutional layers emphasize simpler features and the deep layers recognize more complex or high-order features [5–7].

Advantadges Convolutional layers, when compared with fully-connected layers, present some advantages, namely sparse connectivity, parameter sharing and translation invariance. In fully-connected layers, an output unit depends on all input units, since all are linked. In convolutional layers, as the kernel is much smaller than the input, an output unit is only connected with a small portion of input units, hence the sparse connectivity. Additionally, in fully-connected layers, an input unit has an associated weight for each output unit. In contrast, an entire output feature map is computed by the same kernel, so the weights are shared by all the input units. These two properties greatly reduce the number of parameters and operations, making CNNs more efficient and with less memory requirements. Additionally, as the

filter is applied on every part of the input, the pattern or object detected by it will be recognized, regardless of its location. For example, if a filter is capable of detecting a vessel, with a defined range and angle, a similar vessel, shifted horizontally, vertically or a combination of both, will also be detected. Convolutional layers are invariant to translation but not to other types of transformations, per example, rotation and scale [5–7].

2.3.1.1 Variants

Strided Convolution A convolutional layer can also perform downsampling, i.e, decrease the resolution of feature maps. This happens when stride s_t is superior to 1. In this case, the convolution operation is only applied to a portion of the units in the input feature map. The value of $s_t - 1$ indicates the number of locations on the feature map that will be skipped between each operation. It should be noted that for each spacial dimension a different value of stride can be defined, but usually it's set as equal [5, 10]. Figure 7 contains two examples of this type of convolution.

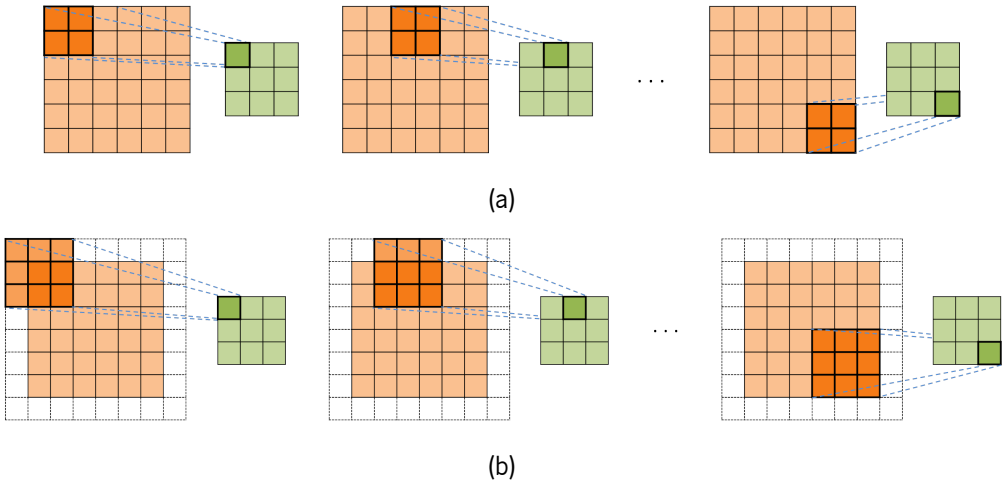


Figure 7: Strided convolution ($s_t = 2$) using a (a) 2×2 and a (b) 3×3 kernel. Based on Dumoulin and Visin [10].

Dilated Convolution The standard convolution can be modified to have an increased receptive field, without increasing the number of performed operations (Figure 8). In dilated convolutions [11], the original kernel size is effectively enlarged by placing zeros between the kernel elements, according to a parameter designated as dilation rate (d). Accordingly, only $k_1 \times k_2$ elements of the kernel, and the feature map, contribute for the computation of one output unit. Considering one dimension, the effective kernel size (k_s) is defined as,

$$k_s = k + (k - 1)(d - 1) \tag{24}$$

Transposed Convolution This kind of convolution is also denominated as fractionally strided convolution and it's usually adopted to perform upsampling. The spatial dimensions of the feature maps increase proportionally with the value of stride. Transposed convolution can be mimicked by a common

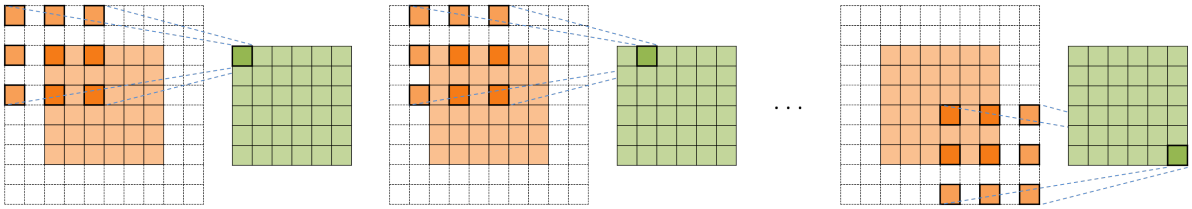


Figure 8: Dilated convolution ($d = 2$) using a 3×3 kernel. Based on Dumoulin and Visin [10].

convolution, in which the value of stride indicates the number of zeros inserted between the input units, as shown in Figure 9. The kernel is convolved through the extended feature maps in order to obtain the output [10, 12].

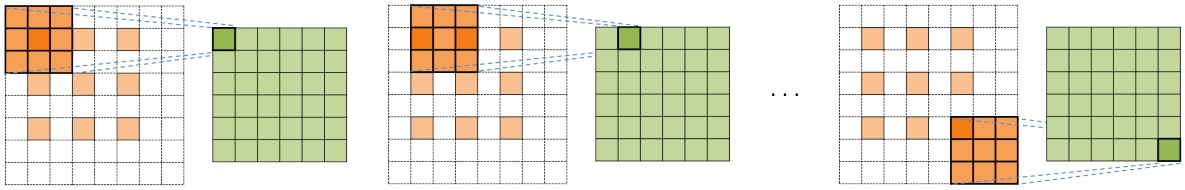


Figure 9: Transposed convolution ($s_t = 2$) using a 3×3 kernel. Based on Dumoulin and Visin [10].

2.3.2 Pooling Layer

Pooling layers are commonly used in CNNs as downsampling operations. Each portion of the feature map is replaced by a statistic of that portion, and the spacial dimensions of the feature map are reduced. The pooling operations include maximum, average and L^2 norm [13]. These operations are computed within a rectangular neighborhood defined by the pooling region. The operation is designated as max pooling [14], when the function corresponds to the maximum. An example of this operation is shown in Figure 10.

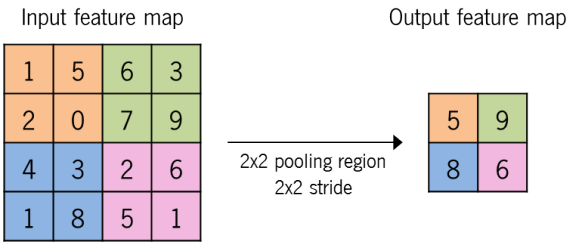


Figure 10: Max pooling operation.

The second parameter of this layer is stride, which indicates the dimension of the feature map that will be reduced to one output unit. Although pooling region and stride can be set with different values, their value is often equal. Note that the pooling operation is applied to each feature map individually [5].

The use of pooling layers make CNNs invariant to small translations. This property is particularly relevant when the task in hand is only the identification of an object, and not its location. Its usage is also relevant to increase the receptive field of the following layers of the network. The pooling layer also decreases the computational cost as the number of input units of the next layer is reduced. It should be noted that it doesn't increase the number of parameters of the network [5].

2.3.3 Activation Function

An activation function is a non-linear function applied individually to each unit, in all feature maps. The integration of this type of operation in a network allows the computation of a non-linear function capable of representing, in theory, any set of data. The activation function is usually fixed, i.e., it's not learned during training [5].

At first, the most used activation functions were sigmoid and hyperbolic tangent, but its use in hidden units is now discouraged. The mentioned functions saturate for high and low input values, which corresponds to the majority of their domain. The flatness of this function can complicate gradient-based learning, since gradient values become very small. When comparing sigmoid and hyperbolic tangent, the second function typically performs better than the first. The hyperbolic tangent has a behavior similar to the identity function near zero and, so, it facilitates training. Anyway, these activations are more commonly used in Recurrent Neural Networks [5].

The recommended activation function is the rectified linear unit [15], defined as

$$g(z) = \ max\{0, z\} \tag{25}$$

When $z > 0$, the function has a linear behavior and the gradient is equal to one, so it can't saturate. In the remaining domain, the output is zero. The process of leaning was demonstrated to be easier with ReLU, when compared with sigmoid and hyperbolic tangent [5].

2.3.4 Batch Normalization

The input distribution of each layer is affected by the parameters of former layers. During training, altering the parameters results in a change of this distribution and, consequently, the layers' parameters need to be continuously adapted to the new distribution. The normalization of the batch, between learnable layers, eliminates this issue by setting the input of all layers to a fixed distribution. More specifically, each feature is normalized independently to have zero mean and unit variance, taking into account all examples of the batch [16].

The simple normalization of features can constrain the network in terms of representation power. This issue is overcome by including parameters,γ and β, capable of scaling and shifting the normalized values, respectively. Thus, during training, the following operations are executed in a batch normalization (BN) layer,

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} a_i \tag{26}$$

$$\sigma_B = \frac{1}{m}\sum_{i=1}^{m}(a_i - \mu_B)^2 \tag{27}$$

$$\hat{a}_i = \frac{a_i - \mu_B}{\sqrt{\sigma_B^2 + \delta}} \tag{28}$$

$$b_i = \gamma\hat{a}_i + \beta \tag{29}$$

where δ is a constant added for numerical stability. By calculating equations 26 and 27, the mean and variance of the minibatch B are obtained to normalize the input (equation 28). Then, in equation 29, the distribution is altered according to the learnable parameters. As the distribution can be modified, the above mentioned problem persists but with less severity, specially for deeper layers, since the distribution only depends on the parameters of BN layer. It should be noted that the normalization can be undone by setting the parameters equal to the mean and standard deviation of the layer input, if it's favorable for optimization [16].

During inference, the BN layer input is normalized according to the population statistics of all training data, so that the normalization of one example of the batch does not depend on the remaining. In practice, a BN layer corresponds to a linear transformation [16].

This layer was initially placed before the nonlinear activation function and after the convolutional layer. The latter mentioned layer likely produces a symmetric and non-sparse distribution, so, its normalization probably originates a stable distribution. Placing BN after a convolutional layer has a consequence, it cancels the bias [16].

The use of batch normalization has many positive consequences, although its effect is not completely understood. It accelerates training and makes the learning process less sensitive to parameter initialization. This layer allows the use of higher learning rates, without the risk of divergence. Batch normalization also has a regularizing effect, reducing overfiting. Lastly, it prevents the network from saturating, so, activations that may lead to saturation can be used [16].

The performance of models containing batch normalization is compromised when utilizing batchs with few or dependent samples. In this cases the distribution of the training set can't be correctly approximated by a minibatch and, thereupon, different representations are used for training and inference [17].

2.3.5 Fully-connected Layer

The fully-connected layers (section 2.2) are placed after the convolutional and downsampling layers, being the last layers of the network. While the first layers are responsible for feature extraction, these layers are responsible for classification. The input of a fully-connected layer has a fixed-size, hence the spatial dimensions of the CNN input are constrained. Furthermore, when a CNN is used for image segmentation, the spatial information is lost and, consequently, only one pixel is classified at a time [5].

2.3.6 Softmax

Softmax function is the last operation applied in a network. The function is used to obtain a probability distribution and it's defined as

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)} \tag{30}$$

where z is a vector composed by k input units and y_i is the probability of a pixel belonging to a given class i. This normalization ensures that the value of each output unit will be between 0 and 1 and the sum of all the outputs units will be equal to 1 [5].

2.4 Fully Convolutional Networks

A Fully Convolutional Network (FCN) [2] doesn't contain an unique fully-connected layer. In this type of network, the fully-connected layers, normally present in a CNN, are replaced by convolutional layers containing 1×1 kernels. This way, the spatial dimensions aren't lost and an output unit is still computed in relation to all input features. The number of parameters of the network are also reduced. Furthermore, FCN can operate on an input of any size and generate an output, probability map, of correspondent size. For the task of semantic segmentation, this is a huge advance, since the prediction of all pixels of a patch, or even the whole image, can be obtained at once. Consequently, during trainning, the loss of all pixels is calculated and utilized for optimization [2].

The first part of a FCN is similar to a CNN, a combination of convolutional and downsampling layers yield a feature hierarchy. Then, upsampling layers retrieve the spatial dimensions and enable pixelwise prediction [2].

Long et al. [2] also noticed that high level information from deep layers gives rise to a coarse output. The addition of skip connections to combine the coarse features with the low level information from shallow layers improved segmentation detail. This way, the semantic information, "what", is provided by deep layers and the location, "where", is provided by shallow layers. A FCN, similar to the one proposed by Long et al. [2], is illustrated in Figure 11.

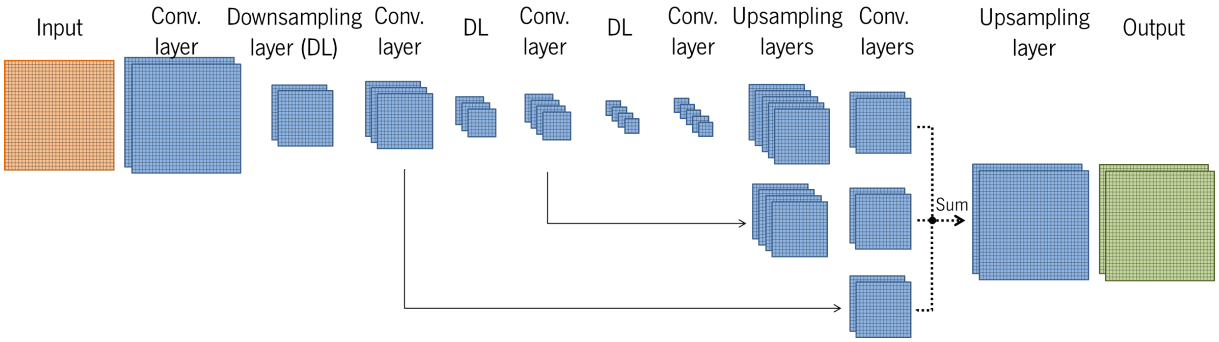


Figure 11: Example of a fully convolutional network. Each convolutional layer is usally combined with batch normalization and an activation function.

Nowadays, FCNs are widely employed for tasks of semantic segmentation, including medical image

processing. The winning approaches of well-known contests are based on this architecture, which allowed to achieve better performances than its predecessor, CNN.

U-Net The architecture proposed by Ronneberger et al. [3] is based on FCN [2] and it's shown in Figure 12. The U-Net is composed by a contracting path and an expanding path, that can be seen as symmetric. The contracting path consists of a set of convolutions and downsampling operations that captures the context information, just like a CNN. The novelty lies on the expanding path. In this path the downsampling layers are replaced by upsampling layers. The upsampled features are progressively combined with the high resolution features from the contracting path, by means of convolutions. Basically, the network learns how to fuse context and localization. It should also be noted that the number of feature channels is equal in both expanding and contracting paths, allowing the propagation of context information to high resolution features.

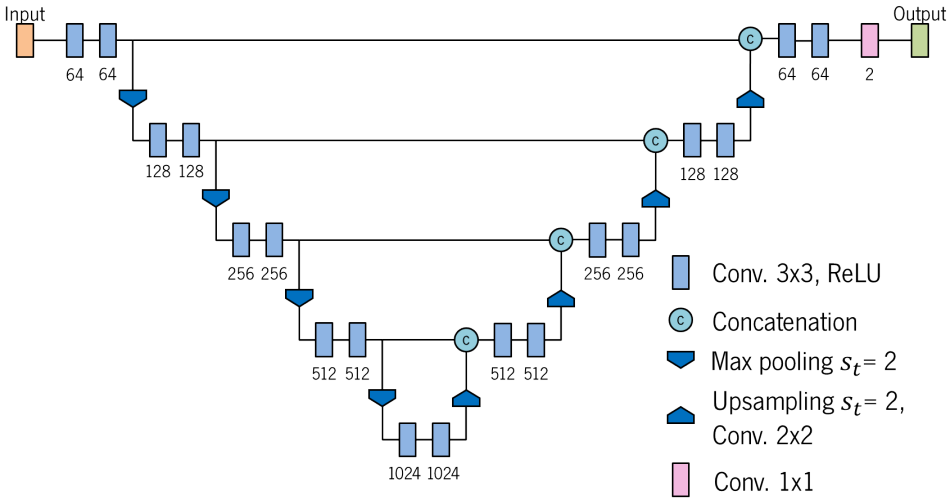


Figure 12: U-Net proposed by Ronneberger et al. [3].

Deep Layer Aggregation Yu et al. [19] proposed Deep Layer Aggregation (DLA) structures to improve combination of features with different complexity. The structures are denominated as hierarchical deep aggregation and iterative deep aggregation and were based on densely connected networks [20] and feature pyramid networks [21], respectively. HDA is responsible for merging semantic information in order to improve recognition. It has tree-structured connections that ensemble features with different levels of representation and equal resolution. This way, HDA allows a deeper fusion when compared with concatenation. IDA merges features with different resolutions, state by stage, to better infer the location. Unlike U-Net [3], where shallow features are aggregated in a single step, the fusion is progressive. In IDA, upsamling is also applied to features from intermediate blocks, which allows the gradual refinement of the shallow features [19]. An example of a DLA architecture containing two downsampling steps is shown in Figure 13.

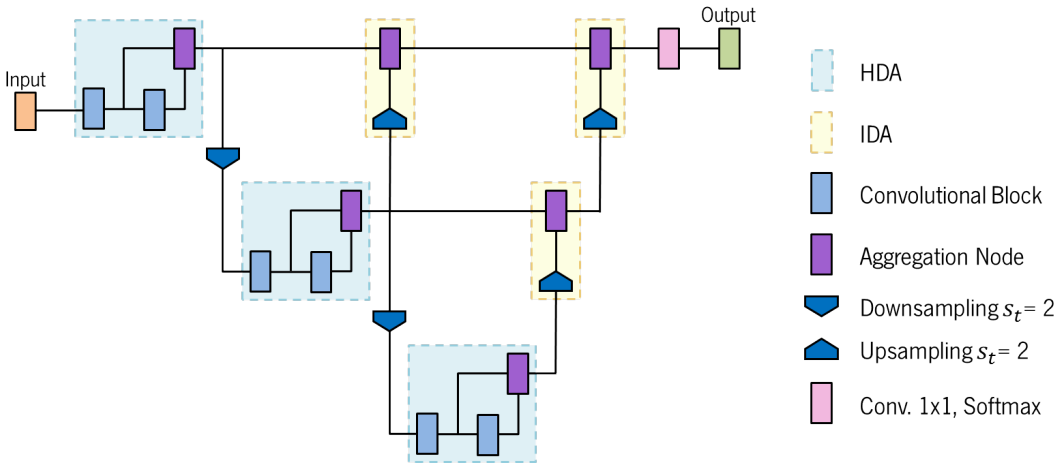


Figure 13: DLA architecture, containing one level HDAs. Acording to Yu et al. [19], each convolutional block contains at least two convolutional layers and each one is followed by ReLU and BN.

2.4.1 Upsampling layer

As previously stated, an upsampling layer is responsible for increasing the resolution of a feature map. The simplest way of upsampling a feature map is to use an interpolation function, such as Nearest Neighbor or bilinear [18]. Using these methods, no trainable parameters are added to the network. In Figure 14, the same feature map is upsampled with both interpolation functions. The transpose convolution, described in section 2.3.1.1, can also be used as upsampling but it has learnable parameters. Regardless of the upsampling method, the output dimensions of this layer are proportional to the value of the stride.

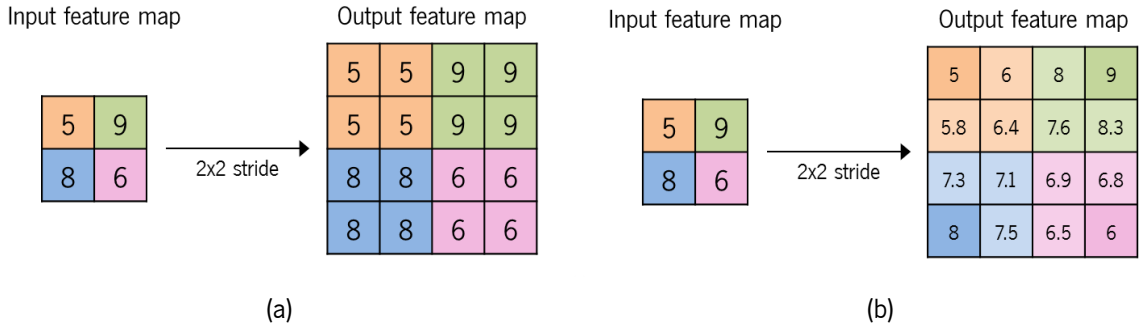


Figure 14: Upsampling operation utilizing (a) Nearest Neighbor and (b) bilinear interpolation.

2.5 Recurrent Neural Networks

Recurrent Neural Networks are a subgroup of ANNs that contain feedback connections, in which the output is fed back into the model. This kind of network was specifically created to process sequential data. Some of the most common examples are temporal sequences and natural language [5].

Similarly to the CNN, this kind of network presents an advantage when compared with the traditional fully-connected neural networks, namely parameter sharing. In practice, at each element of the sequential data, usually denominated as time step, are applied the same weights. This allows the model to generalize

to sequences of variable size. Parameter sharing is also important when a specific information can occur at any position of the given sequence. Unlike CNNs, each member of the output depends on the previous members of the input. Thus, the output of the final time step encodes information regarding the entire sequence [5].

Being the network called recurrent, it contains a function involving recurrence. A function of this kind is responsible for computing the values of the hidden units, also called hidden states. This function adopts the following form,

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}) \tag{31}$$

The hidden state $h^{(t)}$ holds information on the previous hidden states of the recurrent network and on $x^{(t)}$, the current input. In fact, as $h^{(t-1)}$ depends on $h^{(t-2)}$ and $x^{(t-1)}$ and so on, $h^{(t)}$ also holds information on the previous inputs of the network [5].

2.5.1 RNN Layer

The input of a RNN is a sequence containing multiple vectors $x^{(t)}$, where t is the index of the time step which can vary from 1 to the sequence's length τ. During forward propagation, the following update equations are applied for each time step

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)} \tag{32}$$

$$h^{(t)} = \tanh(a^{(t)}) \tag{33}$$

$$o^{(t)} = c + Vh^{(t)} \tag{34}$$

where the bias vectors b and c and the weight matrices U, W and V must be learned throughout training. The weight matrices are responsible for input-to-hidden, hidden-to-hidden and hidden-to-output transformations, which can be represented by fully-connected layers. It should be noted that for $t = 1$, the initial state $h^{(0)}$ needs to be initialized. The activation function was defined as a hyperbolic tangent but can be defined in a different way. If the output of the RNN is the output of the network, softmax can be directly applied to obtain the normalized probabilities $\hat{y}$. Furthermore, the input sequence may be mapped to an output sequence, having both the same length, or to a single output $o^{(\tau)}$, summarizing the entire sequence [5].

A recurrent network can be represented in two ways, both are shown in Figure 15. Unfolding the RNN for a finite number of time steps yields a chain structure and, consequently, it's easier to understand the flow of information [5].

The back-propagation algorithm (section 2.2.1.3) can be directly applied to a recurrent neural network to calculate gradients. However, in this context it's called back-propagation through time [5].

Long Term Dependencies As previously mentioned, the same operation, i.e., the same parameters are applied to each time step of a given sequence. During learning, it's required to compute the gradient of the equation 32 with respect to the hidden state $h^{(t-1)}$, which results in W. For simplicity let's assume

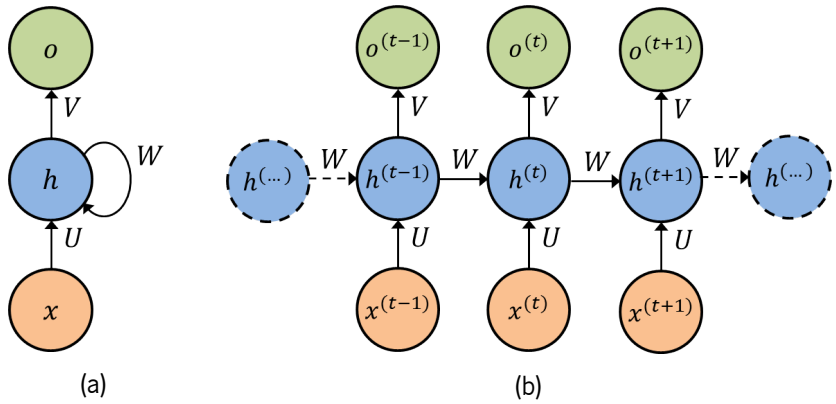


Figure 15: Representation of a RNN (a) with recurrent connections and (b) unfolded.

that all members of this equation are scalars. According to the back-propagation algorithm, W needs to be multiplied by itself several times, to allow the information to flow backwards from the last output to the first input. The number of times that this multiplication happens depends on the length of the sequence. This observation can be easily perceived by analyzing Figure 15b. Considering a long sequence, the gradient will vanish, when W has a low value, less than 1, or explode, when W is greater than one. These are the inherent problems of a RNN, exploding gradients and vanishing gradients [5].

The vanishing gradient problem is the most common and can't be avoided. This problem mostly affects the first stages of the network, where the earlier time steps should be encoded. Therefore, RNNs have difficulty learning long-term dependencies, compared to short-term dependencies. In other words, the information from the first time steps is hardly retained by this kind of network, when compared with the latest time steps [5].

2.5.2 Gated Recurrent Neural Networks

Gated recurrent neural networks were proposed to avoid the problems of exploding and vanishing gradients. Therefore, they also facilitate the task of learning long-term dependencies. In this category of RNNs are included two effective networks, the long short-term memory and the gated recurrent unit [5].

This kind of network uses mechanisms denominated as gates. Gates decide if a given information should be kept or forgotten. In other words, the gates decide which information, time step, is relevant for the task in hand. Thus, these systems basically control the flow of information. The functions applied by these mechanisms are learned throughout training. This involves more parameters and, consequently, more computational cost, when compared with the simple RNN (section 2.5.1). Gates can also be represented as fully-connected layers [5].

2.5.2.1 Long short-term memory

In addition to the hidden state, LSTMs contain another state, denominated as cell state, which is also recursively computed. The cell state is also used to store information, functioning like the memory of the network. Depending on the gates behavior, it's possible for the cell state to take relevant information from initial time steps to final time steps [5, 22].

A LSTM contains three gates, namely forget gate, input gate and output gate. All gates use sigmoid activations, so its output will vary from 0 to 1. The first two gates affect the cell state in different ways. First, the forget gate controls the cell state by removing, or not removing, information from previous time steps. Then, the input gate updates the cell state by deciding what information, from the current time step, should be added to it. Finally, the output gate is responsible for selecting the information that should pass from the cell state to the hidden state and, consequently, to the output of the network [5, 22].

Convolutional LSTM ConvLSTM, an extension of LSTM, was proposed to process spatiotemporal sequences. This kind of layer takes advantage of RNN and CNN capacity to process sequential and spatial data, respectively. The matrix multiplication operators, presented in LSTM, are replaced by convolutional operators, capable of dealing with spatially correlated information, to form ConvLSTM [23].

2.5.2.2 Gated Recurrent Units

Although GRU was created for the same purpose as LSTM, they are quite different. GRU doesn't contain a cell state, so, just like a simple RNN, only the hidden state is used to carry information [5].

Regarding gates, GRU uses only one gate, update gate, to modify the hidden state, as opposed to LSTM, which uses both forget and input gates. Therefore, this gate decides what information should be forgotten, from the previous time steps, and what new information should be added. The second, and last, gate is denominated as reset gate and selects which part of the previous state is used to compute the current one. Both gates can individually reject information from the previous state [5].

2.6 Regularization

As previously stated, one of the main difficulties of machine learning is to create a model that can perform well, not only on training data but also on new inputs (test data). Specifically in deep learning, this issue is exacerbated due to the high number of models parameters. Regularization includes several strategies to decrease the test error, with a possible drawback, the increase of training error. Basically, these strategies help to avoid overfitting and, consequently, promote generalization [5, 6].

2.6.1 Parameter Norm Penalty

The capacity of the model can be limited by adding to the loss function a parameter norm penalty $\Omega(\theta)$, where θ corresponds to the parameters of the model, as previously mentioned. The loss function is now denoted as,

$$J(\theta) = L(\hat{y}, y) + \alpha\Omega(\theta) \tag{35}$$

where $\alpha \in [0, \infty)$ is a hyperparameter that quantifies the weight of the norm penalty on the loss function. When α is set to zero it results in a network without regularization. On the contrary, a large value of α stands for high regularization and possible underfiting. This way, α should be chosen with caution [5].

The use of this penalty allows the value of the parameters to decrease together with the original cost function. Additionally, each layer can have a different penalty and value of α, but in most cases, they are set as equal to all layers of a netwok [5].

The parameter norm penalty usually isn't responsible for regularizing the biases, it regularizes the weights that participate in the layer transformation (w). For example, in a convolutional layer the weights of the kernel will be penalized. Biases are more easily learned and their regularization can lead to severe underfiting [5].

The most common penalty is L^2 regularization, which minimizes the squared L^2 norm of the weights. In this case the regularization term and the loss function are, respectively defined as [5],

$$\Omega(\theta) = \frac{1}{2}\|w\|_2^2 \tag{36}$$

$$J(\theta) = L(\hat{y}, y) + \frac{\alpha}{2} w^\top w \tag{37}$$

L^1 regularization can also be used and it reduces the sum of absolute values of parameters [5].

2.6.2 Data Augmentation

The generalization of a network can be significantly improved by augmenting the amount of training data. On the one hand, the available data for medical image processing is reduced, principally for supervised methods, where data needs to be manually segmented. On the other hand, new data can be obtained from original data, by applying a transformation. This process is called data augmentation [5].

Different operations can be used to increase the amount of data. In the particular case of images, rotating, flipping and scaling are the most used transformations. The manual annotation, for these cases, must be changed according to the data. Another common operation is adding noise to data [5].

It should also be noted that some transformation may be inappropriate for some tasks. This way, the new data should match a possible situation [5].

2.6.3 Dropout

The dropout algorithm consists of removing a unit or a set of units of the network, during training. According to a probability p assigned to each layer, a set of units, from that layer, is randomly set to zero and, consequently, not used to compute the output. The units to drop are changed for each batch of inputs. During prediction, no units are removed, but its value is multiplied by $1-p$, the probability retaining the unit. The application of dropout demands each unit to have good performance, i.e., to encode relevant information, regardless of which other units are in the network. Basically, it prevents the co-adaptation between units [5, 24].

In context of FCNs, dropout proved to be inefficient for overfiting prevention, since the units of the same feature map are strongly correlated. So, a new kind of dropout, spatial dropout, was proposed. Instead of dropping out a unit, a complete feature map is removed according to p [25].

2.7 Parameter Initialization

The algorithms involved in the training of a deep learning model are iterative, so, they demand the specification of the initial points. In fact, initialization can severely affect the learning process. The initial point, when unstable, may forbid the algorithm convergence. The choice of initialization may also affect the time of convergence and the cost function value, at the point of convergence. Furthermore, a given initialization may benefit optimization and impair generalization [5].

Within a layer, weights or kernels with the same initial points, connected to the same input units, will compute the same output and will be updated in the same way, throughout training. So, one property of interest is to initialize weights, specially within the same layer, with different values. In other words, the initial parameters need to break symmetry between different units. Some training algorithms are capable of updating parameters in different ways, but, even in this cases, it's generally beneficial to initialize them with different values. How the initial parameters affect training is poorly understood, therefore avoiding redundant units is the only property required to the initial parameters.

This property motivates random initialization, a simple and heuristic procedure. Often, the biases are initialized as constants and the weights are the ones randomly initialized according to a distribution. The scale of the selected distribution greatly influences the optimization procedure. On one hand, the presence of larger initial weights helps to prevent redundant units and to propagate information through the network. On the other hand, it may lead to exploding or vanishing gradients, the latter situation happens when the activation function is able to saturate. Thus, the selected scale must take into account the described factors [5].

Glorot and Bengio [26] suggested the following normalized initialization strategy

$$W_{i,j} \sim U\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right) \tag{38}$$

where m and n correspond, respectively, to the number of inputs and outputs of a layer l, and U refers to the uniform distribution. The layer weights are sampled from the uniform distribution in an interval restricted, indirectly, by n and m. This strategy was designed to maintain approximately the same activation or response variance and gradient variance in all layers of the network. The formulation of this procedure was based on models with linear activations, but it also performs reasonably well on nonlinear models.This strategy is also called Xavier initialization [5, 26].

He et al. [27] suggested a new initialization, where the weights are sampled from a Gaussian distribution with a standard deviation given by,

$$\sqrt{\frac{2}{n^l}} \tag{39}$$

Similarly to Glorot and Bengio [26], this approach was based on the responses and the gradients variance in each layer. In contrast with Xavier initialization, this strategy was formulated in the context of CNNs and considering models with nonlinear activation functions, more specifically, ReLU activations. He et al. [27] showed that this initialization method helped extremely deep networks to converge.

2.8 Summary

Machine learning methods are able to learn directly from data how to perform a task. These algorithms can be used for image segmentation. In other words, the models are able to learn which category should be assign to each pixel of an image. Deep learning refers to the use of artificial neural networks. A network is a non-linear function whose parameters are learned during training. In this process, the gradients of the loss function are utilized by an optimizer to update the model parameters. The training of the model is extremely conditioned by the learning rate value. The use of Adam, an optimizer with adaptive learning rate, mitigates this dependency.

Convolutional neural networks contain layers that instead of performing matrix multiplication, perform convolution. This networks were specially conceived to process data with a grid structure, such as images. The convolutional layers combine features of neighbor pixels in order to successively extract more complex features. The standard convolution can also be modified to perform downsampling and upsampling and to have an enlarged receptive field. In these cases, the operation is respectively denominated as strided convolution, transposed convolution and dilated convolution. In CNNs, pooling operations are usually employed to increase the receptive field by performing downsampling. When all fully-connected layers are replaced by convolutional layers, the network is designated as fully convolutional. The elimination of matrix multiplications allows the network to operate on images of any size and classify multiple pixels at once. As the first layers encode location and the deeper layers encode semantic information, the combination of features from these layers improves segmentation detail. The U-Net and DLA architectures use different strategies to fuse these features. The latter also merges features with equal resolution.

Recurrent neural networks, LSTM and GRU, are utilized to process sequential data. These networks utilize the information of the current input and the previous inputs to obtain its output. The ConvLSTM is a recurrent network capable of processing data that varies in space and time.

The objective of machine learning is to create a model that presents good performance on training and test data. Overfitting corresponds to the model inability to generalize. This problem is usually related with excessive representation capacity, excessive training duration or lack of training data. The model generalization ability can be improved by using regularization techniques during training, namely, L^2 regularization, data augmentation and dropout.

Chapter 3

Clinical Context

In the scope of this work, deep learning algorithms are employed for image segmentation of retinal vessels and brain tumors. The first part of this chapter presents an overview regarding the retina and its vessels. The standard retinal imaging method is described along with the vessels appearance. Some diseases that manifest in retina are also characterized. Then, the clinical relevance and the difficulties of retinal vessel segmentation are presented. This part of the chapter is finished with the description of state-of-the-art methods for retinal vessel segmentation. The second part of this chapter contains information regarding brain tumors, specially diffuse gliomas. The tumor substructures are identified and their appearance in MRI images is characterized. This chapter also presents the importance of brain tumor segmentation and the main challenges of this task. Lastly, methods for automatic brain tumor segmentation are described.

3.1 Retinal Vessels

The retina is a thin neural tissue located at the back of the eye, between the vitreous and the choroid, which is placed before the sclera. This tissue is responsible for the sense of vision, it converts light into an electrical signal, through specialized structures denominated as photoreceptors [28].

Retinal function is metabolically demanding, therefore the retina is supplied by two different vasculatures. Retinal blood vessels supply the inner half of the retina. The retinal vasculature is a bilayer system composed by two networks of vessels, one is deep and the other is superficial. The large retinal vessels and precapillary arterioles are located in the nerve fiber layer. The small vessels, capillaries and postcapillary venules, are located in the inner nuclear layer. The outer avascular retina is solely supplied by the choriocapillaris. The retinal pigment epithelium (RPE) and the Bruch's membrane, located between the choroid and the photoreceptors, are responsible for regulating the transport of metabolic wast, nutrients and water [28]. The retinal layers can be observed in Figure 16.

The fovea, located in the center of the macula, is an avascular zone with high density of photoreceptors. Besides, in this region, there's a depression in the retina duo to lack of some inner retinal layers. These characteristics contribute to fovea being the region with the highest visual acuity [28].

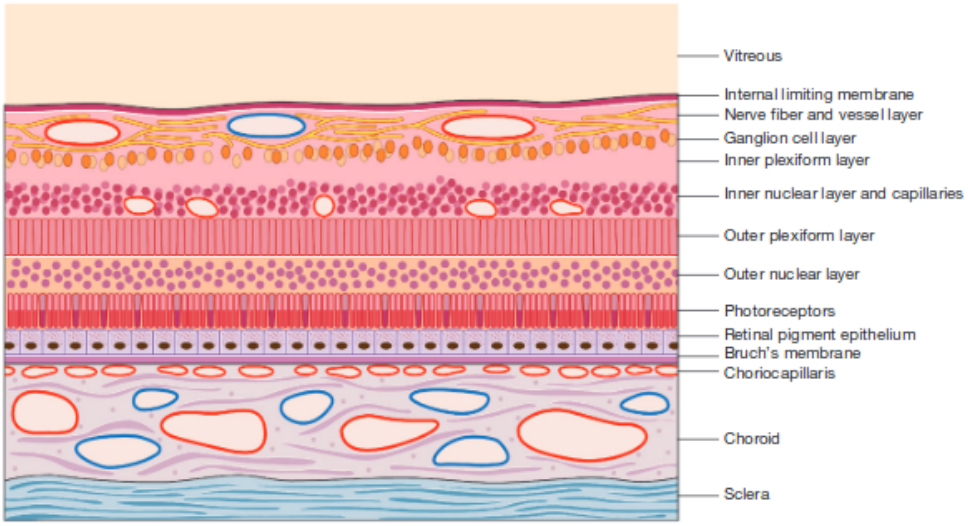


Figure 16: Layers of the retina. Reproduced from Ryan et al. [28].

3.1.1 Retinal Imaging

Retinal fundus photography is the standard method for imaging the fundus of the retina and provides structural information about the eye [29, 30].

This imaging technique consists on using a camera to obtain a photograph, 2D image, of the retina. The utilization of white or green light allows the acquisition of a color or red-free image. The images are reproducible and have high resolution. Retinal fundus photographies are also rapidly acquired, immediately available and amenable for image enhancement [29, 30].

There are two types of fundus photography that vary in terms of field of view. Standard photography provides a 30° to 50° image, where the posterior pole, back of the eye, is optimally observed. Other noticeable structures include retinal vasculature, macula and optic disc. Only 5% to 15% of retinal surface area is observed, so, a large portion of the peripheral retina is not captured. The field of view may be restricted by media opacities, such as cataract and vitreous hemorrhage. This type of photography is widely available. Ultrawide field imaging produces a retinal image with a view up to 200°, which allows visualization of the peripheral retina. With this technique, over 80% of retinal surface is imaged. A larger field of view allows detection of pathology in peripheral retina and more thorough documentation. Ultrawide field photography has many disadvantages, namely, image distortion, eyelash artifacts, false color representation of fundus findings and higher equipment cost. Thus, standard photography is more used than ultrawide field photography [29, 30].

Retinal fundus photography is used to analyse the retinal vasculature. In these images, the vessels have a particular form and appearance. The retinal vessels are arranged in a tree-like structure. The vessel intensity and orientation don't change abruptly along the length. Regarding their cross-sectional profile, vessels have a Gaussian or a mixture of Gaussians shape. A light reflex may appear in the center of the vessel, specially in arteries [29, 31, 32].

Color fundus photography is an essential component of ophthalmic examination, being utilized for population screening, natural history studies and assessment of response to treatment [29, 30, 32, 33].

3.1.2 Retinal Diseases

The retina and its vasculature are affected by a great diversity of diseases, namely, conditions with ophthalmic and cardiovascular origin [32–35]. Some of these diseases and their respective clinical signs are described bellow.

3.1.2.1 Aged-related macular degeneration

Aged-related macular degeneration (AMD) causes retinal degeneration and visual loss. AMD is characterized by the presence of drusen within the macula. Drusen corresponds to a combination of proteins and lipids that is deposited somewhere between Bruch's membrane and RPE. The presence of drusen is not necessarily related with AMD. Small amounts of hard drusen usually appears due to aging. Only when these accumulations are large, confluent or soft, the patient suffers from AMD [31, 36].

There are two types of AMD, dry or non-neovascular AMD and wet or neovascular AMD. In adition to drusen, dry AMD causes changes in the RPE, usually visible as pigment clumps. Throughout disease progression, the amount and size of drusen increases and RPE degenerates, causing the loss of photoreceptors. Advanced dry AMD is associated with geographic atrophy. Wet AMD, also an advanced state of AMD, is characterized by proliferation of choriocappillaris into subretinal space, whithin the macula. The new vessels may bleed, causing subretinal hemorrhage that may lead to RPE detachment and vitreous hemorhage. The neovascular membrane may undergo fibrosis, forming a disciform scar. The advanced forms of the disease cause vision loss [31, 36].

The treatment of non-neovascular AMD slows down the progression of the disease and consists of nutritional supplementation, antioxidants and minerals. Neovascular AMD is treated with anti-VEGF, laser photocoagulation or a combination of both. The laser theraphy prevents further deterioration while the administration of anti-VEGF may improve the patient vision [31, 36].

3.1.2.2 Diabetes mellitus

Diabetes mellitus is characterized by an elevated level of glucose in the blood. There are two types of diabetes. Type I is caused by the destruction of pancreatic cells, which leads to low levels of insulin and consequently high levels of blood glucose. In this case, exogenous supplementation of insulin is needed. In Type II, the body is capable of producing insulin but not in sufficient quantity or the cells don't respond to it. This type of diabetes is strongly correlated with obesity and sedentary lifestyle. Hyperglycemia damages all organ systems, specially, the vasculature. Diabetic retinopathy (DR), a condition of diabetes, is the most common cause of blindness [30, 31].

The earliest form of DR is denominated as nonproliferative. The first sign of this condition is the presence of microaneurysms. Throughout the disease progression, other signs appear, namely, intraretinal hemorrhage, cotton wool spots and hard exudates. Nonproliferative DR also causes intraretinal microvascular abnormalities (IRMAs), which are usually near areas of capillary nonperfusion. IRMAs include vessels with anomalous curvature, contortion, dilation and branching. These atypical vessel structures may result from remodeling of preexisting vessels or growth of new vessels. Other symptom is venous beading,

consisting in a vein that is alternately contracted and dilated [30, 31].

The previous described form may progress to proliferative diabetic retinopathy. This condition is characterized by neovascularization of the optic disk, retina and iris. The new vessels may bleed, causing preretinal or vitreous hemorrhage. The growth of retinal new vessels is accompanied by fibroglial proliferation, viewed as a white tissue in the neovascular frond. This process is denominated as fibrovascular proliferation. The contraction of the fibrous component may lead to tractional retinal detachment. Both vitreous hemorrhage and tractional retinal detachment induce vision loss [30, 31].

Diabetic macular edema or retinal thickening, the main cause of visual loss, may appear in both forms of diabetic retinopathy and is caused by vascular leakage [30, 31].

The treatment of DR prevents or retards the visual loss. It consists of intravitreal injection of anti-VEGF, panretinal laser photocoagulation, surgery or a combination of them. Macular edema may require macular laser photocoagolation and injection of intravitreal corticosteroids [30, 31].

3.1.2.3 Hypertension

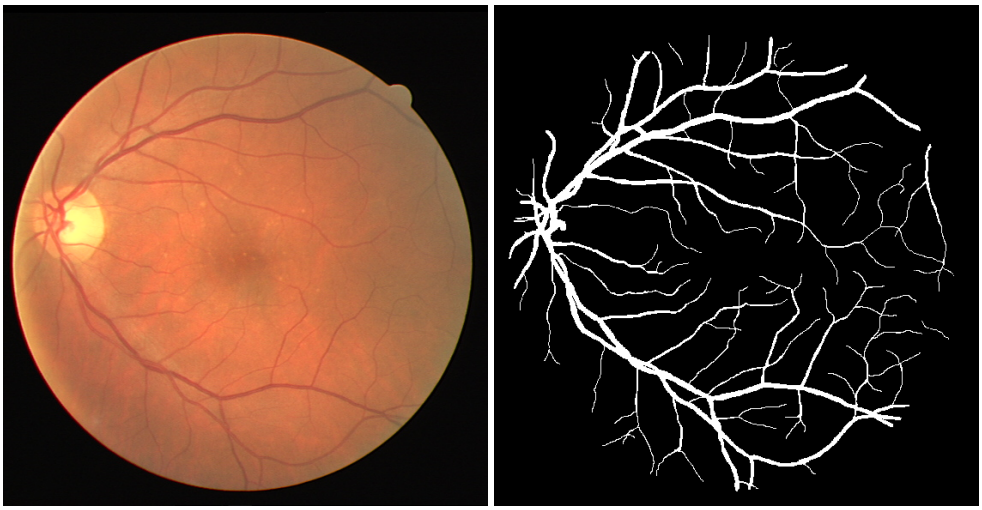
Hypertension is one of the major causes of morbidity and mortality. This systematic disease affects the whole body, but some organs are specially affected by it, namely, heart, kidneys, brain and eye. Regarding the eyes, disease duration and severity determines the structural and functional changes suffered by the vasculature, which includes vessels from optic disc, choroid and retina. Elevated blood pressure is associated with many clinical conditions, being hypertensive retinopathy the most common manifestation [31, 36].

Initially, hypertensive retinopathy causes retinal arteriolar narrowing. Then, the wall of arterioles thickens and becomes opaque, a condition named arteriosclerosis that changes the appearance of vessels. The light reflex along the vessel progressively widens until the vessel becomes similar to either copper or silver wire. Other observable sign is arteriovenous nicking, in which an arteriole compresses a vein, causing its tapering, in each side of the crossing, and deflection. The affected vein also dilates and has increased tortuosity. Other symptom of this condition is venular widening. The prolongated exposure to elevated blood pressure leads to blood-brain barrier disruption. The consequences of the disruption are microaneurysms, hemorrhages, hard exudates, in the form of macular star, and cotton wool spots. In severe cases, hypertension may also cause optic disc swelling [31, 36].

3.1.3 Retinal vessel segmentation

The quantitative analysis of vessels morphological characteristics, such as length, width, tortuosity, branching patterns and angle, are utilized for diagnosis, screening, treatment and evaluation of diseases affecting the retina. This analysis requires the segmentation of retinal vasculature [31–33]. Figure 17 presents a retinal fundus image and the respective segmentation of blood vessels.

Manual segmentation of retinal vessels is time-consuming and a tedious task for ophthalmologists. It also requires training and skill. The obtained segmentations are prone to inter and intra-observer variability [31–33]. Therefore, an automatic vessel segmentation method is needed.



(a) (b)

Figure 17: (a) Retinal fundus image and (b) the respective blood vessel segmentation, from DRIVE database [77].

Although some appearance aspects of vessels are well known, retinal vasculature segmentation is a complicated task. The vessels characteristics, including shape, length and width, vary greatly. The crossing and branching of vessels also hinders the task. The vessels intensity differ in a wide range across the image. The contrast between vessels and background may be low, specially for narrow vessels. Lastly,the presence of pathological signs, such as microaneurysms, cotton wool spots, bright and dark lesions and exudates, further difficults the segmentation, since they may have similar attributes to vessels [31, 32, 35].

The low quality of the acquired images provides more challenges for retinal vessel segmentation. The images may present noise, intensity drift and lack of contrast. The image usually contains specular reflections from the cornea and lens. Consequently, the image contains very bright and dark regions, with widely different contrast between vessels and background. [31, 32].

3.1.3.1 Automatic retinal vessel segmentation approaches

The automated retinal vessel segmentation algorithms proposed in literature can be divided into two main groups, supervised and unsupervised. Regardless of the type of algorithm, pre-processing and post-processing techniques are also employed to improve the performance of the approach.

Unsupervised methods attempt to find intrinsic patterns of blood vessels in retinal images that can be used to decide whether a pixel belongs to a vessel or the background. Thus, these approaches don't require labeled data for the design of the algorithm [32, 37]. Badawi and Fraz [38] employed a Bar-Combination of Shifted Filter Responses capable of detecting symmetric and asymmetric bar-shaped patterns. This method utilizes difference of Gaussian filters. Subsequently, artifacts were removed using three different mechanisms. The background artifacts filtering algorithm removes the small objects that are disconnected from the vascular tree. The K-median clustering algorithm is used to remove background artifacts. Lastly, the black and white artifact clearance eliminates small objects and fills small holes. Only the green channel

of the retinal fundus images is used in the experiments. Aguirre-Ramos et al. [39] started by reducing the noise of the retinal image's green channel with a Low Pass Radius Filter. Then, the usage of a Gabor filter and a Gaussian fractional derivative allowed the enhancement of the vessels and their borders. After thresholding the enhanced image, segmentations were improved by post-processing techniques. The optic disc is detected on the green channel of the image using a double threshold process in order to be removed from the segmentation results. An iterative length filter is employed to remove the border of the field of view. Then, the segmentation artifacts are reduced with a series of filters and morphologic operations.

On the contrary, supervised methods require pre-labeled data during the training phase for algorithm's design. Manual segmentation and the respective image, or its features, are used by a classifier to learn the better suited function for vessel delineation [32, 37]. In the first proposed methods, features were manually extracted according to the vessels characteristics and used as input to a classifier. Zhang et al. [40] formed a feature vector for each pixel that is composed by orientation scores based features, Gaussian differential features and the intensity of the green channel. The pixels were categorized as vessel and non-vessel by a Random Forest classifier. Before applying a threshold, the maximum orientation score of the exudates were subtracted from the probability map. The proposed approach by Wang et al. [41] used 100 features extracted with different techniques, such as, matched filter, 2-D Gabor wavelet transform, Frangi filter, difference of Gaussian and gray-level-based features. The features were extracted from the three color channels of the retinal image. Retinal vascular tree was obtained with Mahalanobis distance classifier. A set of morphological operations are employed to obtain the number of branch points of each connected component and, subsequently, remove pathological regions. Regarding pre-processing, they applied background normalization and a truncation filter with the aim of minimizing image intensity fluctuation and noise, respectively. Recently, methods that can automatically extract a set of complex features have been widely adopted. This way, the extraction of manual features, which is problem dependent and requires expert knowledge, isn't needed. Liskowski and Krawiec [42] proposed the use of CNNs and studied the effects of employing pre-processing techniques and data augmentation. Two pre-processing techniques were evaluated, namely, normalization of each individual patch to have zero mean and unit variance and zero-phase component analysis whitening. The training data was augmented using scaling, rotation, flipping and gamma correction. As they utilized CNNs, each pixel is individually classified. Dasgupta and Singh [43] employed a FCN [2] for retinal vessels segmentation, in which an entire patch is segmented at once. The input of the network consists of patchs extracted from the green channel of the retinal images. Four pre-processing techniques were applied before the experiments, namely, normalization, contrast limited adaptive histogram equalization, gamma adjustment and intensity scaling. Mo and Zhang [44] investigated the use of auxiliary losses in a FCN for vessel segmentation. The auxiliary losses were applied to each branch output. The authors argues that different regions of the retinal image require different sizes of receptive field for a correct segmentation. The training data was augmented using multiple transformations, namely, rotation, scaling, flipping, and mirroring. Yan et al. [45] trained a FCN with two loss functions, pixel-wise loss and segment-level loss. The last one deals with the imbalance between thick and thin vessels. Only the green channel of the retinal image was utilized to segment the vessels. During training, flipping, rotation, resizing and adding random noise were the strategies utilized for data

augmentation. In Hu et al. [46], a FCN with a class-balanced cross-entropy loss function generates a probability map for each retinal fundus image. Afterwards, a fully-connected conditional random field is employed to refine the segmentation result, using the information of the complete image. Regarding data augmentation, the data was scaled, rotated and flipped. Oliveira et al. [47] used Stationary Wavelet Transform to create new input channels to a U-Net based network. The green channel was also used as input channel, after being normalized. Rotation operations were employed to augment the data used during training. They also proposed data augmentation for prediction, i. e. the probability maps of the original and rotated test images were averaged to obtain the final segmentation. Guo et al. [48] employs a FCN with short connections, which pass low level information to high level layers and high level information to low level layers. It's intended to exploit the semantic and structural information provided by low and high level layers, respectively. The FCN takes as input the green channel of the retinal fundus images, which was rotated, flipped and scaled to increase the training data. Shin et al. [49] proposed a novel architecture that results from the combination of a CNN with a graph neural network. The authors intended to take advantage of local features and the graphical structure of the vessels. Regarding pre-processing, the mean value of the pixels is subtracted from each image channel. Some data augmentation strategies were employed, namely, horizontal flipping, random brightness and contrast adjustment.

3.2 Brain Tumors

A tumor, which can also be denominated as neoplasm, is the result of abnormal cell proliferation [50, 51]. Tumors located in the brain can be classified into two categories according to their origin. A primary tumor derives from brain cells while a metastatic tumor derives from other body cells that have spread or metastasized to the brain [51, 52].

Tumors are also categorized into benign and malignant. Benign tumor cells grow slowly and are similar to their tissue of origin. Generally, benign neoplasms don't metastasize or invade adjacent tissues and are associated with good prognosis. On the contrary, malignant cells are capable of unlimited multiplication at a rapid growth rate. These cells present irregular shape and size and enlarged nuclei, a condition denominated as anaplasia, duo to lack of cellular differentiation. Malignant tumors usually contain necrotic areas and induce neovascularization. Moreover, these neoplasms frequently originate metastases and have poor diagnosis. A malignant tumor is also designated as cancer [50].

The World Health Organization (WHO) proposed a grading scheme that indicates the degree of tumor malignancy. Brain tumors can be classified into four grades, from I to IV. Grade I is assigned to tumors with low cell proliferation. Grade II neoplasms still have slow cell growth, but are infiltrative and may progress to higher grade tumors. Neoplasms with grade III and IV, or high-grade tumors, are anaplastic and present high cell proliferation. Grade IV neoplasms often exhibit necrosis and are the most aggressive tumors [53, 54].

The patient symptoms depend on the tumor's location and growth rate. The symptoms can be included in three main categories. Elevated intracranial pressure can be felt as headache, vomiting, diplopia and altered level of consciousness. These signs may be related with tumor mass or surrounding edema. The

signs of focal neurologic deficit are extremely correlated with the tumor location. Some examples are aphasia, ataxia, sensory loss and cognitive impairment. The last major symptom is seizures [52].

The treatment of brain tumors include surgery, chemotherapy and radiotherapy, or a combination of all them. The chosen therapy is influenced by the tumor grade and type and the patient medical history [52].

Patient prognosis worsens with the increase of the tumor grade. Prognosis also depends on other factors such as patient age, tumor location, genetic features and proliferation indices [53, 54].

The incidence of brain tumors is low, around 2%, but these tumors are associated with higher mortality rate when compared with other types of tumor [52].

Within brain tumors, the most frequent primary neoplasms in adults are denominated as gliomas, which arise from glial cells. Glial cells include astrocytic, oligodendroglial and ependymal cells. The tumors derived from these cells are astrocytoma, oligodendroglioma, oligoastrocytoma and ependymoma, respectively [51, 52].

WHO classification groups all diffuse gliomas together, which include neoplasms originated by astrocytic, oligodendroglial and mixed cells. Within this kind of glioma, neoplasm grade ranges from II to IV. Low-grade gliomas (LGGs) resemble normal glial tissue, but contain an excessive number of cells. These gliomas often contain cysts and may present calcification. The low aggressiveness of LGGs allows a long median survival, between 8 and 12 years. Grade III neoplasms are denominated as anaplastic gliomas. These neoplasms may contain hemorrhage. Calcification is frequently present in anaplastic oligodendrogliomas and oligoastrocytomas. Anaplastic neoplasms usually disturb the surrounding tissues by causing significant vasogenic edema and mass effect. The latter results in sulcal effacement, ventricular compression and corpus callosum thickning. The median survival rate may vary between 2 and 5 years, depending on the histological features of the tumor. In addition to anaplastic glioma characteristics, glioblastomas, WHO grade IV, present necrosis and endothelial proliferation. Glioblastomas can be primary, when there's no precursor lesion, or secondary, when they arise from lower level astrocytomas. In this case, the median survival can be extended to 14 months, when treatment is applied. In adults, the incidence of high grade gliomas (HGGs) is higher than the occurrence of LGGs and glioblastoma is the most common glioma [52–54].

3.2.1 Glioma Imaging

Magnetic Resonance Imaging provides good contrast for soft tissues, which include the brain tissues. Additionally, two parameters of MRI, excitation and repetition times, can be varied to acquire different MRI sequences and, consequently, observe different structures. In other words, images with different tissue contrast are acquired with MRI. This characteristics make MRI suitable for the visualization of the brain and its sub-structures [51].

In fact, MRI is the standard technique for imaging brain tumors and, specifically, gliomas [51, 55, 56]. To correctly examine the glioma and its substructures, independently of its grade, four MRI sequences are needed, namely, T1-weighted (T1), T1-weighted with contrast enhancement (T1c), T2-weighted (T2) and fluid-attenuation inversion recovery (FLAIR). The healthy tissues are easily observed in T1. The use of a

contrast agent, gadolinium-DTPA, allows the detection of the proliferative tumor region in T1c. This is only possible due to accumulation of contrast agent in the extracellular space, which is caused by disruption of the blood-brain barrier and formation of abnormal new vessels, characteristics of malignant tumors. This substructure, also called enhancing or active region of the tumor, is hyperintense in the T1c sequence and its predominantly present in HGG. The subtraction of T1 from T1c can be used to ease the delineation of this tumor region. Also in T1c, the necrotic structures are noticeable inside the enhancing region, because of their low intensity. The non-enhancing core of the tumor is visualized in T2 and FLAIR, where this region is hypointense when compared with edema. To properly delineate this substructure of the tumor, the T1 and T1c sequences may be needed. The monitoring of this region is principally relevant for LGGs, since they don't usually contain enhancing region. Moreover, the growth of the non-enhancing substructure may also indicate posterior growth or emergence of enhancing tumor. The remaining part of tumor is outlined using two sequences. In T2, edema appears as a hyperintense region, surrounding the tumor's core. FLAIR is used to verify the extension of the tumor's edema near the cortex and ventricles. In this sequence, the signal from bulk fluid is suppressed, so, the structures filled with cerebrospinal fluid can be distinguished from edema. It should be noted that, in T2 and FLAIR, all tumor substructures are shown as hyperintense regions. The negative side is that the imageable part of the tumor may not match the complete extent of the tumor [51, 56–58].

3.2.2 Analysis of Gliomas

The MRI-based analysis is used in diagnosis, patient monitoring, treatment planning and clinical trials [51, 55, 56, 59].

The acquired MRI images are accessed based on qualitative criteria and quantitative measures [57]. The tumor size is traditionally estimated using diameter-based approaches. The Response Evaluation Criteria in Solid Tumors introduced the use of unidimensional measurements to estimate the tumor burden. In this criteria, the largest diameter of the lesion in the axial plane is measured, using only one MRI slice. The Macdonald criteria was created to assess HGG response, using bidimentional measurements. According to this criteria, the enhancing tumor size corresponds to the product of the two maximal and orthogonal diameters in the axial plane. The non-enhancing region of the tumor isn't evaluated in any way, what is detrimental for HGG since the enhancing region alone is not sufficient to characterize tumor growth. It also means that this criteria can't be extended to LGG [51, 59, 60]. To overcame this issues, the Response Assessment in Neuro-Oncology (RANO) Working Group proposed a new criteria, denominated as RANO criteria, that takes into account both enhancing and non-enhancing regions of the tumor. In the presence of a HGG, the enhancing tumor size is quantified using the maximal cross-sectional diameters, just like in Macdonald criteria, and the non-enhancing part is only qualitatively evaluated [60]. In the presence of a LGG, the non-enhancing region of the tumor is quantified, using bidimensional measurements, while the enhancing region, if present, is qualitatively evaluated [58].

Although unidimensional and bidimensional measurements are simple and fast to perform, they present some limitations. First, the irregular shape and the eccentric growth of the tumor aren't taken into account. Second, the presence of cystic or necrotic regions in the lesion poses a challenge for tu-

mor measurement. Third, the obtained value of tumor size is very sensitive to head tilt during acquisition. Lastly, the inter-observer variability is high due to the identification of the longest diameter(s) by the experts [56, 58–61].

The described limitations can be surpassed by using robust, reproducible and accurate measures of the tumor size, namely volumetric measures. These measures are sensitive to small volume changes, as opposed to 1D and 2D measures. Additionally, the tendency is to use volumetric measures to quantify the tumor and its substructures, providing a more complete information about the tumor. To obtain the volume of the tumor and its subregions, they must be outlined in all MRI slices, in other words, the tumor regions need to be segmented [51, 56–60].

3.2.3 Brain tumor segmentation

Segmentation of brain tumors, exemplified in Figure 18, is done manually, in current clinical practice. Manual segmentation is time-consuming and a tedious task for radiologists. Moreover, the segmentations performed by different experienced raters may contain considerably different delineations of the tumor and its substructures, issue denominated as inter-rater variability [51, 57, 59]. This way, an automatic segmentation approach is needed.

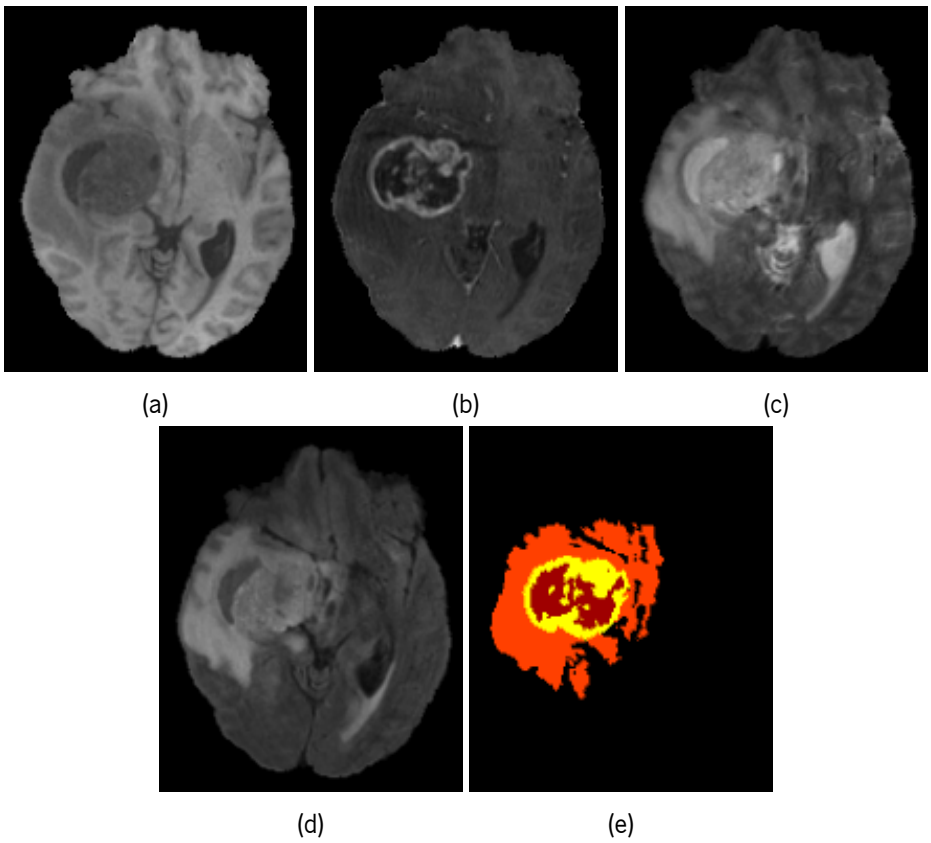


Figure 18: MRI sequences of a patient with an HGG: (a) T1, (b) T1c, (c) T2 and (d) FLAIR. In (e) the brain tumor segmentation, orange corresponds to edema, yellow identifies the enhancing region and red represents the necrotic region. In this case, the non-enhancing core region isn't present. These images are incorporated in the BRATS 2017 database [57, 90].

The characteristics of the lesion difficult the development of an automatic method for brain tumor

segmentation. The structures of the tumor can highly vary in size, shape and appearance, across patients. Additionally, the tumor location, in the brain, isn't always the same and the tumor's growth causes a mass effect that may displace normal brain tissues. Thus, no prior knowledge can be used to facilitate the task [51, 57].

The brain tumor segmentation is further hindered by the imaging technique. The MRI images may contain partial voluming or bias field artifacts, what smooths the image gradient between adjacent structures and, consequently, complicates edge delineation. Furthermore, the image contrast may be significantly different when using different scanners or different acquisition parameters. Per example, a higher magnetic field strength results in an image with higher lesion contrast [56, 57]

3.2.3.1 Automatic brain tumor segmentation approaches

Similarly to the approaches used for retinal vessel segmentation, the brain tumor segmentation algorithms can be divided in two main groups, unsupervised and supervised. The unsupervised methods utilize domain-specific prior knowledge, namely, the appearance and spatial distribution of the different tissues to obtain the segmentation. The supervised methods require the data and the respective manual annotations to learn how to distinguish the different tissues. Initially, local image features were manually extracted from the MRI images and used as input to a classifier. Recently, methods capable of extracting features directly from the data have been widely adopted [57].

The current methods employed for brain tumor segmentation are mainly supervised and based on CNNs. Some of these state-of-the-art methods are described bellow.

Soltaninejad et al. [62] used texton based histograms and the probability map originated by a FCN as input features to a Random Forest classifier. In this work, various pre-processing methods were applied, namely, elimination of the highest and lowest values of intensity, normalization of the sequences to have zero mean and unit variance, histogram normalization and linear normalization.

Jesson and Arbel [63] employed a 3D FCN for brain tumor segmentation. During training, a multi-scale loss, based on the cross-entropy loss, was applied on predictions given at each resolution of the FCN. This allowed the combination of higher resolution features and lower resolution predictions. The aim is to enable the network to learn context in image and label domains. In this work, the problem of class imbalance was tackled by using a curriculum on sample weights. Regarding pre-processing, each image was standardized to have zero mean and unit variance. The training data was augmented with rotation, shearing and flipping.

In Isensee et al. [64], a 3D U-Net based network was utilized to perform the task. The final output resulted from the sum of segmentations obtained at different network levels. A multiclass dice loss function was used to deal with class imbalance. In order to prevent overfitting, various data augmentation techniques were applied during training, namely, rotations, scaling, elastic deformations, gamma correction operations and mirroring. Test time data augmentation was also employed, using the mirroring transformation. They utilized 5 fold cross-validation, originating multiple models. These models were used to segment the test data and the results were averaged, similarly to an ensemble. The data was pre-processed, each sequence of each patient was normalized to have zero mean and unit variance, clipped

to remove outliers and, then, rescaled.

Kamnitsas et al. [65] employed an ensemble of multiple 3D architectures, namely, two multi-scale CNNs, three FCNs and two U-Nets. In addition to architectural changes, the networks differ in terms of training configurations, such as, optimizer, loss function, regularization and data augmentation techniques. Each network was trained on data pre-processed in three different ways, further increasing the number of models. The data were normalized using one or more of the following techniques, Z-score normalization, bias-field correction and piece-wise-normalization. The final segmentation resulted from the average of each model prediction.

Wang et al. [66] proposed a cascade of FCNs to sequentially segment the brain tumor regions, decomposing a multiclass problem into three binary segmentation tasks. Within each FCN, the authors utilize multi-scale prediction to combine features of different levels and dilated convolutions to enlarge the receptive field. Anisotropic convolutions are also used in order to take advantage of 3D information without excessively increasing the memory consumption. The cascade was separately trained with slices from the three orthogonal views. The final segmentation resulted from averaging the outputs of each cascade. Regarding pre-processing, the mean and variance of training images were used to normalize the entire data.

Islam and Ren [67] utilized a CNN based network, PixelNet [68], in which each pixel is classified according to its multiscale hypercolumn features. Basically, the features obtained with the multiple convolutional layers are concatenated and used as input to the fully-connected layers. In order to tackle the problem of class imbalance, the network was trained with equal number of samples from each class. The MRI sequences were previously pre-processed, namely, the N4 algorithm was utilized to remove the bias field distortion and each axial slice was normalized to have zero mean and unit variance.

In Zhou et al. [69], the brain tumor segmentation task is decomposed in simpler tasks. As opposed to cascaded methods, where each task is individually learned, they proposed the use of a single model to learn the three tasks. In practice, the majority of the model parameters is shared by all tasks and each task has two independent convolutional layers and a loss function. The authors argue that the correlation between tasks should be taken into account. Furthermore, the complexity of the model decreases. The training of the network, with U-Net based architecture, was based on curriculum learning. In other words, the model started by learning one simpler task and the remaining were gradually added, aiming to improve the convergence quality of the model. These authors also applied post-processing methods, consisting of removing small and isolated clusters and using K-means clustering algorithms to better distinguish edema from non-enhancing tumor. Before the experiments, each MRI sequence was normalized to have zero mean and unit variance.

Pereira et al. [70] employed two U-Net based networks for the task. The first is a 3D model and defines a region of interest, containing the whole tumor. The second is responsible for segmenting the tumor regions. The latter 2D network contains segmentation Squeeze-and-Excitation blocks, which recombine the feature maps through linear expansion and contraction. The goal is to create more complex features by emphasizing or suppressing certain regions of the feature maps. The pre-processing methods included bias field correction, standardization of the intensity histogram of each MRI sequence and normalization according to the training set. Regarding data augmentation, images suffered rotation and flipping.

Chen et al. [71] proposed a dual force training scheme on both U-Net, CNN and cascade based networks. According to the authors, an auxiliary loss function should be applied to high-level features in order to obtain high quality hierarchical features. The ground truth used in the added loss, Kullback-Leibler divergence loss, corresponds to a label distribution map. A post-processing method based on a multilayer perceptron was also employed to refine the segmentations. Three features were used as input, namely, the output probabilities of the CNN, voxel intensity and volume of the predicted tumors. For pre-processing, each MRI sequence was normalized to have zero mean and unit variance.

In Mlynarski et al. [72], three 2D models were used to capture large context information on axial, sagittal and coronal planes, respectively. In addition to the original MRI sequences, the features extracted from these networks were set as input to a 3D U-Net. The authors state that this approach allows capturing a large 3D context without computational constraints associated to 3D networks. Each 2D model contained multiple subnetworks, one processed all sequences and the remaining processed each MRI sequence individually, followed by a main network, which used the features extracted from the subnetworks to obtain the segmentation in a view. The mentioned architectures were changed to create other 5 models for brain tumor segmentation. They also proposed a voting strategy to fuse the multi-class segmentations originated by the multiple approaches. During training, a weighted cross-entropy loss was employed to deal with class imbalance. Before the experiments, each individual sequence was normalized, i.e., divided by its median value and multiplied by a fixed constant.

3.3 Summary

Located in the back of the eye, retina is responsible for the sense of vision. This tissue and its vasculature can be imaged using retinal fundus photography. Several opthalmic and cardiovascular diseases, such as AMD, diabetes mellitus and hypertension, manifest themselves in the retina and can lead to visual loss. These diseases cause microaneurysms, cotton wool spots, hard exudates, arteriovenous nicking and the growth of new vessels. The retinal vessels are also affected, in terms of width, length, tortuosity, appearance and branching angle. The changes in the vessels can only be quantified after segmenting the retinal vasculature. The proposed methods for automatic retinal vessel segmentation can be divided into supervised and unsupervised. Regarding recent approaches, the majority of the methods consist of FCNs.

Brain tumors are associated with high mortality rates, being diffuse gliomas the most common ones. These tumors can be divided into LGG and HGG, according to its malignancy. The standard technique to image tumors is MRI. The examination of the tumor substructures requires the acquisition of four sequences, namely, T1, T1c, T2 and FLAIR. A reliable quantitative analysis of the tumors can be obtained using volumetric measures. The tumor substructures need to be segmented in order to obtain their volume. Supervised and unsupervised methods are used for brain tumor segmentation. The majority of the recently proposed approaches utilize FCNs. Ensemble methods are also extensively employed for this task.

Chapter 4

Study of the U-Net architecture for retinal vessel segmentation

In this chapter, the structure of the U-Net is studied for retinal vessel segmentation. More specifically, this study focuses in the incorporation of dilated convolutions and in the downsampling operation. First, this chapter presents the motivation for the modifications implemented on the U-Net architecture. The next section describes the experimental setup, which includes the database, data pre-processing, architectures, training settings and evaluation metrics. Lastly, the results and discussion are presented.

4.1 Motivation

Downsampling operations have been widely and successfully employed in CNNs to increase the receptive field of subsequent layers. Nevertheless, the successive augment of context information provided by these operations is accompanied by the decrease of spatial acuity [5]. This reduction of resolution may result in the loss of details that may hamper the task of semantic segmentation [73].

The alternative is to use dilated convolutions (section 2.3.1.1), which have an enlarged receptive field and don't affect the spatial resolution of their input. Thus, the receptive field of a network can be maintained by removing the downsampling steps and replacing standard convolutions with dilated convolutions. However, the removal of downsampling steps leads to a high increase of computational cost. Given the trade-off between spatial resolution and computational cost, a reasonable solution is to preserve part of the downsampling steps and use dilated convolutions to replace the rest.

Nevertheless, the spacing between kernel elements in dilated convolution can cause gridding artifacts. This kind of artifacts were discussed in some works and handled in different ways [73, 74]. The strategy of Wang et al. [74] consists in choosing the dilation rates, of a series of convolutional layers, in a way that the receptive field covers a square region without missing edges or holes. Therefore, within a sequence of convolutional operations, dilation rates with equal value or common factor relationship shouldn't be used to avoid gridding artifacts. Alternatively, Yu et al. [73], after the needed layers for receptive field increase, stacked convolutional layers with progressively lower dilation rate. The network used by Yu et al. [73] contained residual connections in all its blocks. These connections had to be removed from the layers

with progressively lower dilation rate, because they propagate the gridding artifacts.

Regarding the downsampling operation, max pooling is usually employed. Yu et al. [73] showed that this pooling operation exacerbated the gridding artifacts, when used in conjunction with dilated convolutions. In section 2.3.1.1, an alternative downsampling operation was presented, strided convolution.

In this work, a new block, containing dilated convolutions, was develop based on the works of Wang et al. [74] and Yu et al. [73]. The dilated convolutional block (DCB) was incorporated into a U-Net architecture to study the affect of replacing a downsampling step with dilated convolutions. Additionally, two types of downsampling are compared, namely, max pooling and convolution with 2×2 kernel, both with stride 2. These studies were made for the task of retinal vessel segmentation using the DRIVE database.

4.2 Experimental Setup

This section presents the implementation details. The deep learning approaches were implemented using Keras [75] with TensorFlow [76] backend, libraries in Python programming language.

4.2.1 Database

The models were evaluated in a publicly available database, DRIVE [77]. This database contains 40 color images, in which 7 contain pathology. The images are divided into two sets, training and test, each containing 20 images. The resolution of the image is 565×584 and each channel has 8 bits. The field of view (FOV) of the images is 45° and a mask that delineates it is provided for each image. This dataset provides two manual segmentations. The manual segmentations from the first human observer were used as ground truth. The validation set consists of two images, subjects 34 and 40, randomly chosen from the training set.

4.2.2 Pre-processing and patch extraction

The green channel of the retinal fundus images was used in the experiments, since it presents the better contrast between retinal vessels and background when compared with the blue and red channels [78, 79]. After extracting the green channel, only one pre-processing technique was applied. All images, including the test set, were normalized using the mean and standard deviation of the training set.

In the training phase, 4000 patches were randomly extracted from each training image. Regarding data augmentation, rotation operations with 0 (no rotation), 90, 180 and 270 degrees were executed on the original and annotated patch. During training, the original and rotated patches were randomly presented to the network, according to a multimodal distribution with $p = 0.25$.

The patch size was 80×80 and the correspondent labeled patch was 32×32. The difference in patch size is due to the use of valid convolutions in the neural network.

4.2.3 Network Architectures

The implemented networks are based on the U-Net proposed by Ronneberger et al. [3]. In the contracting path of this network, after a block, downsampling is applied to generate higher level features. After each downsampling, the number of feature maps, obtained by convolutional layers, is doubled. Next, in the expanding path, the feature maps are upsampled and their number is halved. The features from the contracting path are added to the output of the upsampling layers, through long skip connections. The added features have the same resolution, belonging to the same level. This encoder-decoder architecture enables the netwok to capture both context and precise location. At the final layer of each network, a 1×1 convolution and Softmax are used to map each feature vector to the number of classes and compute prediction.

Two types of blocks were used, regular convolutional blocks (RCBs) and DCBs, which are presented in Figure 19. Both blocks incorporate convolutional layers with a 3×3 kernel followed by Batch Normalization (BN), Rectified Linear Unit (ReLU) and Dropout. RCBs are composed by two convolutional layers, in which the dilation rate was set to 1, without any residual connection. DCBs contain a set of convolutional layers with different dilation rates, capable of covering an entire square region. In the initial layers, the value of dilation rate increases from 1 to 3 with residual connections. The final layers present a dilation rate progressively lower. The design of DCBs was based in the works of Wang et al. [74] and Yu et al. [73]. In Figure 20a, it's shown that the usage of the previously described sequence of dilation rates is not prone to gridding artifacts. Other example, a sequence that may cause gridding, is also provided (Figure 20b).

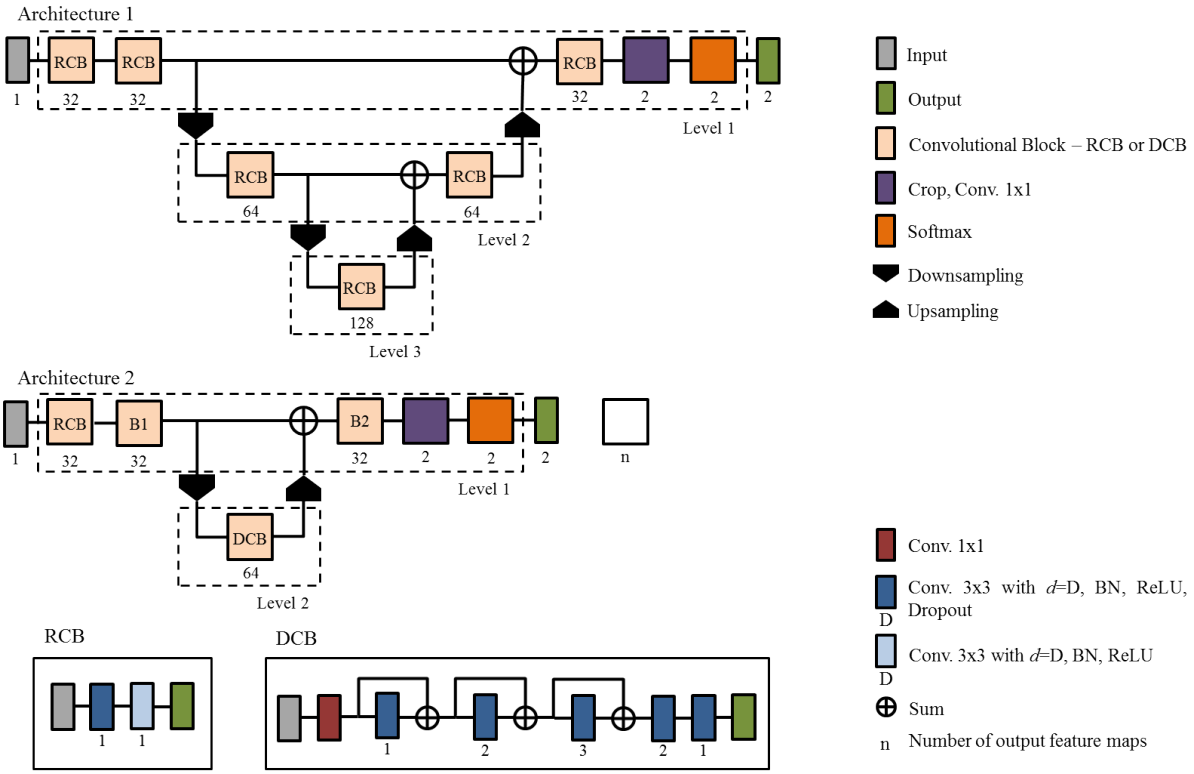
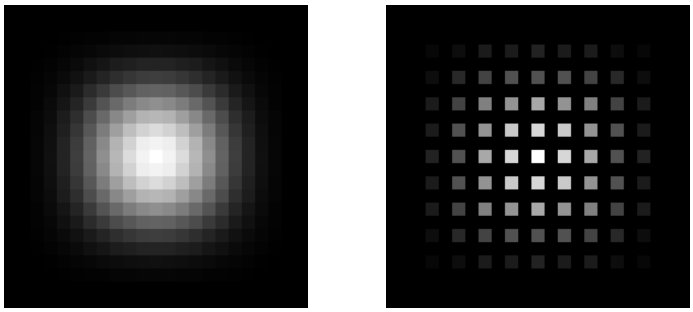


Figure 19: General architectures of the proposed approaches and structure of the two types of blocks used, DCB and RCB.



(a) (b)

Figure 20: Pixels' contribution to the calculation of the center pixel, when using convolutional layers with (a) $d = 1, 2, 3, 2, 1$ and (b) $d = 2, 4, 2$. The contribution of a pixel is proportional to its gray level.

Five different models were studied to segment retinal blood vessels. The Baseline-M and Baseline-C follow the first architecture (Figure 19), having all convolutional blocks defined as RCBs. Downsampling is performed with max pooling and convolution with 2×2 kernel with stride of 2, respectively. The remaining models follow the second architecture (Figure 19). In Dilate-C and Dilate-M, B1 and B2 blocks are set as RCB. The difference between these two networks is the downsampling operation, one uses convolution with 2×2 kernel and stride of 2 and the other uses max pooling. The last model, Full-Dilate, only has DCBs and a convolution with 2×2 kernel and stride of 2 is responsible for downsampling.

4.2.4 Training settings

The models were trained for 54 epochs with a batch size of 4. He normal [27] was used to initialize the weights of convolutional layers. Categorical cross-entropy and Adam [9] were chosen as loss function and optimizer, respectively. The learning rate and β_1, parameters of the optimizer, were changed throughout training by the cosine annealing scheduler [80]. In the first cycle, with a length of 4 epochs, the learning rate and β_1 started as 1×10^{-3} and 0.8 and ended as 1×10^{-5} and 0.95, respectively. During training, the maximum learning rate decreases by a factor of 0.9 and the cycle length increases by a factor 1.5. Dropout was the only regularization employed with probability $p = 0.2$ for all blocks, except the last one where p was set as 0.15.

4.2.5 Evaluation Metrics

The different methods were evaluated by comparing the output segmentation with the manual annotations. The computation of metrics was done regarding the pixels inside the FOV area.

Each individual pixel can be denominated as a true positive (TP) or false positive (FP), when correctly or incorrectly classified as vessel. Regarding background pixels, the designations are identical, namely, true negative (TN) and false negative (FN).

The most common performance measurements for retinal vessel segmentation are accuracy (Acc), sensitivity (Sens) and specificity (Spec). Acc corresponds to the proportion of correctly identified pixels, regarding both vessel and non-vessel pixels. Sens and Spec quantify the model's ability to detect vessels and background, respectively. These metrics are calculated as follows [81]:

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{40}$$

$$Sens = \frac{TP}{TP + FN} \tag{41}$$

$$Spec = \frac{TN}{TN + FP} \tag{42}$$

In this work, Mathews Correlation Coefficient (MCC) was also used due to its insensibility to class imbalance. MCC is computed as [81]:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{43}$$

Other reported metric is the area under the ROC curve (AUC). The computation of this metric requires the probability map outputted by the model. For each threshold, varied from 0 to 1, Sens and 1-Spec are calculated to create the ROC curve. As opposed to previous described metrics, the calculus of the AUC measures the performance of a model without depending on the chosen threshold [81].

4.3 Results and Discussion

In this section, the architectures that differ in terms of type and number of downsampling operations and usage of dilated convolutions (DCBs) are analyzed. Then, the obtained results are compared with state-of-the-art methods.

4.3.1 Architecture Study

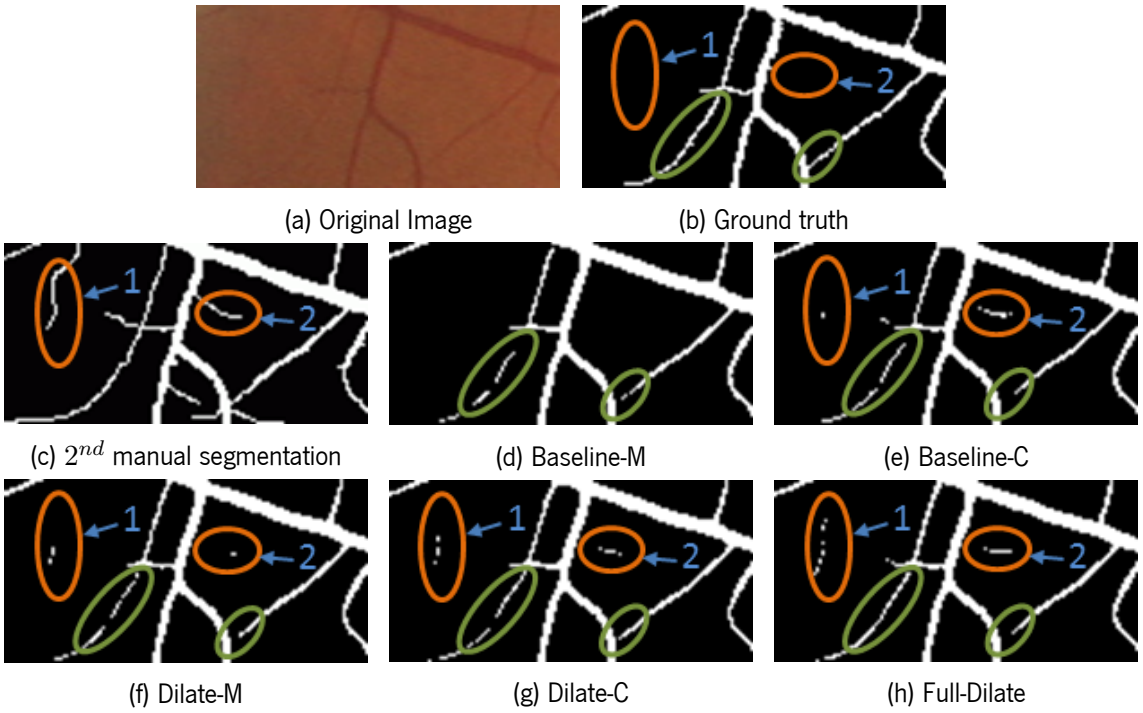
The obtained results are shown in Table 1. In Figures 21 and 22, the main differences between models' segmentation results are shown.

Fusion of levels Usually, the receptive field is increased by downsampling operations that have as side effect the reduction of spatial resolution. The Baseline-M and Baseline-C models contain two downsampling steps in order to increase the receptive field. A possible alternative to this kind of operation is the use of dilated convolutions, which augment the receptive field while maintaining the spatial resolution. In Dilate-M and Dilate-C models, the second downsampling operation was eliminated and a block, containing dilated convolutions, was employed to approximately maintain the receptive field of the models. Basically, in these models, one downsampling step was replaced by dilated convolutions. The similarity in terms of receptive field makes possible to evaluate the use of dilated convolutions, when comparing Baseline-M and Baseline-C with Dilate-M and Dilate-C models, respectively. In Table 1, it's shown that using dilated convolutions allow the achievement of a superior performance in terms of all metrics, except Spec. Also, in Figures 21 and 22, it can be noted that the Dilate-M and Dilate-C models, when compared with the non-dilated models, detect more thin vessels. The vessels annotated only by the 2^{nd} human observer (Figure

Table 1: Segmentation results of different architectures on the DRIVE test set. The metrics mean and standard deviation are presented, the later between parenthesis. Values in bold show the best score among all approaches. The number of parameters (NP) of each architecture is also presented.

Model	NP (10^3)	Acc	Sens	Spec	MCC	AUC
Baseline-M	410	0.9560 (0.0050)	0.7752 (0.0608)	0.9827 (0.0055)	0.7950 (0.0225)	0.9787 (0.0060)
Baseline-C	431	0.9564 (0.0050)	0.7767 (0.0573)	**0.9830** (0.0051)	0.7972 (0.0215)	0.9792 (0.0060)
Dilate-M	237	0.9561 (0.0045)	0.7845 (0.0582)	0.9815 (0.0055)	0.7968 (0.0200)	0.9799 (0.0054)
Dilate-C	241	**0.9569** (0.0043)	0.7931 (0.0553)	0.9811 (0.0055)	0.8010 (0.0192)	**0.9800** (0.0056)
Full-Dilate	300	0.9568 (0.0040)	**0.8072** (0.0528)	0.9789 (0.0058)	**0.8026** (0.0174)	0.9795 (0.0062)

21 - arrows 1 and 2) are barely noticeable due to their characteristics, narrow width and low contrast. Despite this and the fact that only the annotations from the 1^{st} observer were used to train the models, these vessels are detected, principally by the models containing dilated convolutions. This indicates that some of the FP, responsible for the decrease of Spec, are in fact real vessels. The hemorrhage, marked as 1 in Figure 22a, is partially segmented as vessel by some models, specially the Dilate-M and Dilate-C models. It should also be pointed out that the use of dilated convolutions replacing a downsampling step reduced the number of parameters in more than 40%, which can be critical when using small databases.



(a) Original Image (b) Ground truth

(c) 2^{nd} manual segmentation (d) Baseline-M (e) Baseline-C

(f) Dilate-M (g) Dilate-C (h) Full-Dilate

Figure 21: Pieces of segmentation results from image 19, regarding the study of the U-Net architecture. The increase of FP and FN, in relation to the ground truth, are marked in orange and green, respectively.

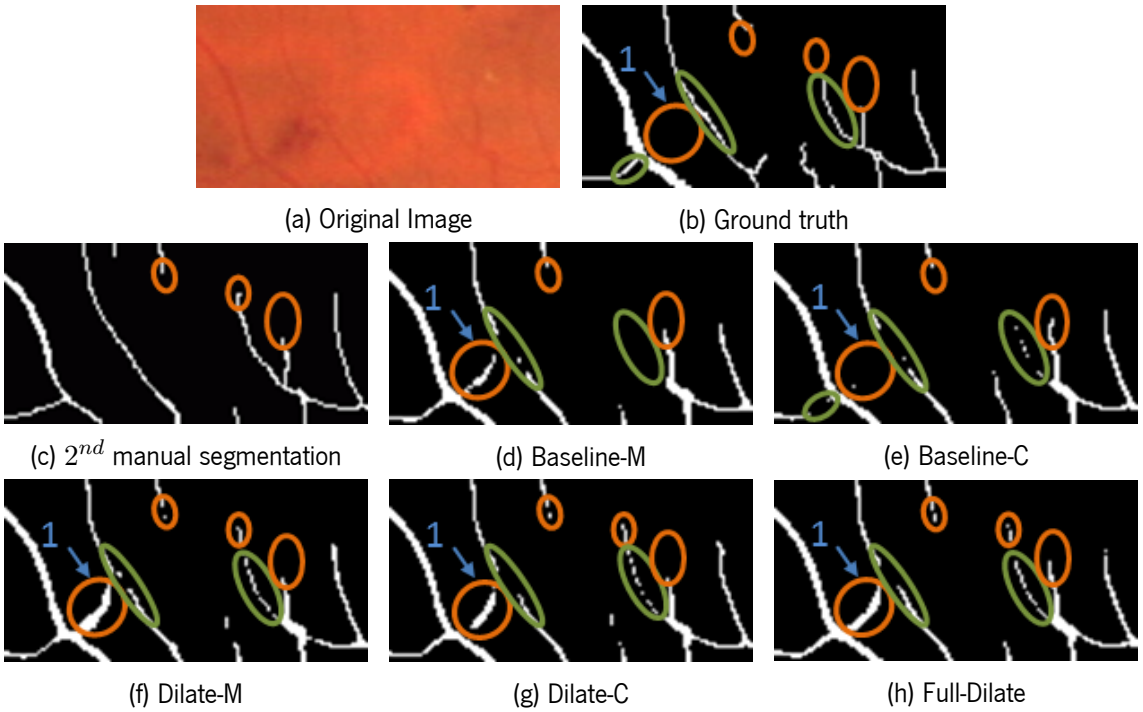


Figure 22: Pieces of segmentation results from image 14, regarding the study of the U-Net architecture. The increase of FP and FN, in relation to the ground truth, are marked in orange and green, respectively.

Downsampling Block Max pooling is the most common downsampling technique for image classification and segmentation. This operation is used in Baseline-M and Dilate-M models. In Baseline-C and Dilate-C models, downsampling is performed by convolution with 2×2 kernel and stride 2. Analysing Baseline-M, Baseline-C, Dilate-M and Dilate-C models, the effect of this substitution can be evaluated. In general, models using convolution as downsampling have substantially higher performance, when compared with models utilizing max pooling (Table 1). The tendency previously observed, regarding the detection of vessels annotated by the 2nd human observer but marked as background by the 1st one, was accentuated with the substitution of max pooling with convolution. In Figure 22 for the baseline models, we verified that using convolution as downsampling, the network was able to correctly discriminate between hemorrhage and vessel. Regarding dilated models, convolution was only able to partially attenuate the false detection of hemorrhage as vessel. Specially in the presence of dilated convolutions, the use of max pooling is detrimental for retinal vessel segmentation. This is according to the study by Yu et al. [73], that showed an exacerbation of gridding artifacts when combining max pooling with dilated convolutions. This can explain the higher performance decrease observed in Dilate-M model, comparing with Dilate-C model.

DCB Block As dilated convolutions were shown to be beneficial for retinal vessel segmentation, allowing the detection of more vessels, the use of DCBs (Figure 19) was extended, forming the Full-Dilate model. Analyzing the results in Table 1, the model indeed detected more vessels since Sens increased by 1.41%, but with a downside, the number of FP increased enough to negatively affect the value of Acc. Besides, AUC slightly decreases. Figure 21h shows the presence of less FN for thin vessels, comparing with Dilate-

C model. Also in this figure, it's possible to note a higher detection of vessels that were only annotated by the second human observer. Regarding Figure 22h, the area of hemorrhage misclassified as vessel increased.

As Dilate-C model presents a higher value of Acc, it was selected as best performing model. Nevertheless, it should be pointed out that Full-Dilate model presents a good performance, specially in terms of Sens and MCC.

4.3.2 Comparison with the state-of-the-art

The proposed approach for retinal vessel segmentation, Dilate-C model, and the baseline model, Baseline-M model, are compared with state-of-the-art approaches in Table 2. This table contain recently proposed methods, previously described in section 3.1.3.1.

Table 2: Segmentation results of different approaches, including the Dilate-C model, on the DRIVE test set. Values in bold show the best score among all methods.

Methods		Year	Acc	Sens	Spec	MCC	AUC
Unsupervised	Badawi and Fraz [38]	2018	0.9547	0.7898	0.9709	-	-
	Aguirre-Ramos et al. [39]	2018	0.9503	0.7854	0.9662	-	-
Supervised	Liskowski and Krawiec [42]	2016	0.9515	0.7520	0.9806	-	0.9710
	Dasgupta and Singh [43]	2017	0.9533	0.7691	0.9801	-	0.9744
	Zhang et al. [40]	2017	0.9466	0.7861	0.9712	0.7673	0.9703
	Mo and Zhang [44]	2017	0.9521	0.7779	0.9780	-	0.9782
	Yan et al. [45]	2018	0.9542	0.7653	0.9818	-	0.9752
	Hu et al. [46]	2018	0.9533	0.7772	0.9793	-	0.9759
	Oliveira et al. [47]	2018	**0.9576**	0.8039	0.9804	**0.8054**	**0.9821**
	Wang et al. [41]	2019	0.9541	0.7648	0.9817	0.7851	-
	Guo et al. [48]	2019	0.9561	0.7891	0.9804	0.7964	0.9806
	Shin et al. [49]	2019	0.9271	**0.9382**	0.9255	-	0.9802
	Baseline-M	2019	0.9560	0.7752	**0.9827**	0.7950	0.9787
	Dilate-C	2019	0.9569	0.7931	0.9811	0.8010	0.9800

The proposed method stood in second for Acc and MCC, being surpassed by Oliveira et al. [47]. As stated in section 3.1.3.1, Oliveira et al. [47] utilized test time data augmentation and extra input channels obtained with Stationary Wavelet Transform. Excluding the effects of this two procedures, Acc drops to 0.9567, which is lower than the value achieved by the Dilate-C model. In terms of Sens, the proposed method has the third highest value being outperformed by Shin et al. [49] and Oliveira et al. [47]. Shin et al. [49] detected around 94% of the vessels, but it has a much lower performance in terms of Acc and Spec. Thus, the approach proposed by Shin et al. [49] has both an elevated number of TP and false detections, compromising the detection of background. The Dilate-C model achieved the third best score in relation to Spec, when the baseline model isn't taken into account. Nevertheless, it's noted that the works with superior Spec, Yan et al. [45] and Wang et al. [41], have lower Acc and Sens, detecting less vessels. Regarding AUC, the proposed approach stood in fourth, following Oliveira et al. [47], Guo et al.

[48] and Shin et al. [49].

It should be pointed out that the Baseline-M model, a simple U-Net with three levels, obtained the highest value of Spec. Although this model has a low value of Sens, rank tenth, it stood in fourth regarding Acc. Furthermore, it has the fourth and fifth highest performance in terms of MCC and AUC, respectively. This shows that both models are competitive with state-of-the-art methods and suitable for retinal vessel segmentation.

4.4 Summary

In CNNs, downsampling operations are regularly employed to increase the receptive field of subsequent layers and decrease the computational cost. The use of this operation also reduces the resolution which may cause the loss of spatial details, necessary for the task of semantic segmentation. Alternatively, dilated convolutions, which allow the increase of receptive field without affecting the spatial acuity, can be used to replace part of the downsampling operations. Downsampling is usually employed with max pooling. The simultaneous use of max pooling with dilated convolutions was shown to be harmful for semantic segmentation. Convolution with stride is a possible alternative.

In this chapter, the deep learning methods are employed for retinal vessel segmentation. A U-Net based architecture was used as baseline to study the effect of replacing one downsampling step with dilated convolutions. As dilated convolutions are associated with gridding artifacts, a block containing dilated convolutions, DCB, was developed. The downsampling operations are also compared in two situations, when using standard and dilated convolutions. Lastly, the use of DCBs was extended, replacing the regular convolutional blocks.

Regardless of the downsampling operation, the use of a DCB replacing a downsampling step was beneficial for retinal vessel segmentation and allowed the detection of more thin vessels. Some of these vessels were only marked by the observer not used for training. Comparing the two downsampling operations, strided convolution obtained higher overall performance, specially in the presence of dilated convolutions. When strided convolution is used, less hemorrhage is erroneously detected as vessel and more thin vessels are annotated. Extending the use of DCBs decreased the performance of the model in terms of Acc. Thus, the architecture containing one DCB and strided convolution as downsampling operation, Dilate-C model, obtained the best performance. The proposed approach achieved Acc, Sens, Spec, MCC and AUC of 0.9569, 0.7931, 0.9811, 0.8010 and 0.9800, respectively. Dilate-C and baseline models are competitive with state-of-the-art methods.

Chapter 5

Increase of context information using RNNs on retinal vessel segmentation

In this chapter, the ReNet, a layer composed by recurrent neural networks, is employed for retinal vessel segmentation. This layer is applied to the features extracted by the model Dilate-C, the best performing model presented in the previous chapter. The motivation for using ReNet is initially presented. Then, the chapter describes the experimental setup, including the database, data pre-processing, architectures, training settings and evaluation metrics. The results are exhibited and discussed in the last section.

5.1 Motivation

A CNN only exploits local information, ignoring long distance dependencies between pixels [82].

Visin et al. [82] proposed a RNN-based layer, named as ReNet, to provide relevant global information. The ReNet is composed by four uni-dimensional recurrent neural networks. Two RNNs sweep across the patch horizontally in both directions. Then, the remaining two RNNs sweep across the patch vertically, also in both directions. The sequential use of these RNNs ensures that the output computed for each pixel depends on the entire patch and, consequently, the patch dimension becomes the receptive field of the model. This layer was originally designed to replace CNNs in image classification tasks [83]. In the context of semantic segmentation, ReNet was stacked on top of a FCN, so the model benefits from both local and global features [83]. This layer was also used in the area of medical image processing for stroke lesion segmentation, in combination with an U-Net [84, 85].

In this work, the ReNet layer was incorporated into the pre-trained Dilate-C model to try to improve its performance by retrieving long distance dependencies.

5.2 Experimental Setup

In this section, the implementation details are presented. The deep learning approaches were implemented in Python, using Keras [75] with TensorFlow [76] backend.

5.2.1 Database

The proposed approach was evaluated in the DRIVE database [77], already described in section 4.2.1. In summary, DRIVE is composed by two sets, training and test, containing 20 retinal fundus images. The ground truth corresponds to the manual segmentations from the first human observer provided by the database. Two images from the training set, subjects 34 and 40, compose the validation set.

5.2.2 Pre-processing and patch extraction

The images were pre-processed and the patches were extracted according to description given in section 4.2.2. In summary, the green channel of each image was normalized using the mean and standard deviation of the training set. Each training image provided 4000 patches for the model's training. During the learning process, rotations with 0 (no rotation), 90, 180 and 270 degrees were applied to the patches.

5.2.3 Network Architectures

The Dilate-C network is based on the U-Net [3] and it's presented in Figure 23. Basically, this model has two levels, the first contains blocks with standard convolutions (RCBs) and the second has a block including dilated convolutions (DCB). A detailed description of the network and convolutional blocks can be found in section 4.2.3.

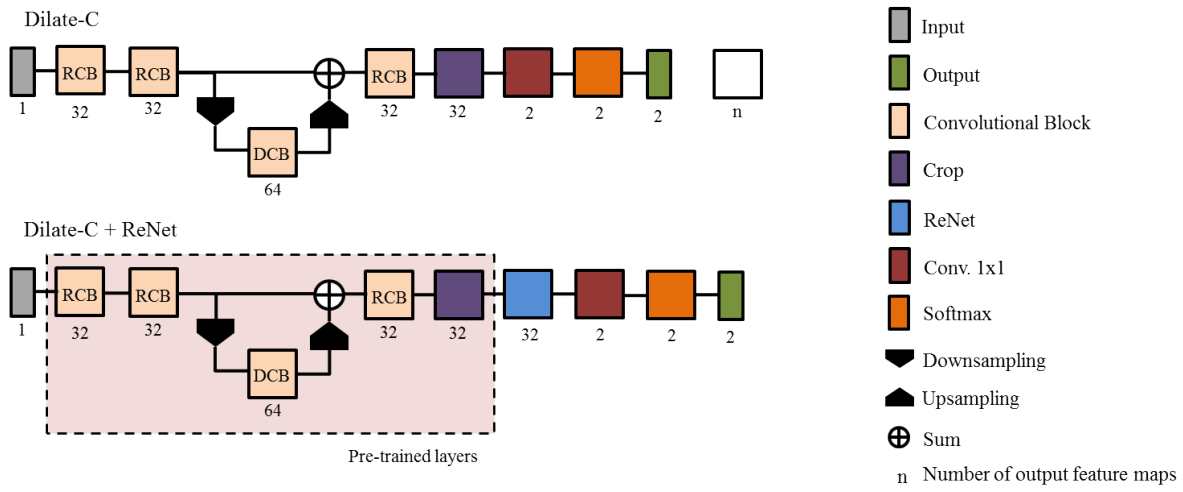


Figure 23: Architectures of the Dilate-C model and the proposed approach, Dilate-C + ReNet model.

The ReNet layer was implemented with four GRUs. First, the input of this layer is sweeped horizontally in both directions. Two GRUs, one for each direction, are simultaneously applied and their outputs concatenated. Then, the remaining GRUs sweep vertically across the concatenated features also in both directions. Lastly, the outputs of these RNNs are concatenated.

The proposed approach, Dilate-C + ReNet, consists on incorporating the ReNet layer in the Dilate-C network, according to Figure 23. After ReNet, a 1×1 convolution and Softmax are utilized to obtain the probability distribution over the classes.

5.2.4 Training settings

Glorot uniform [26] was used to initialize the weights of GRUs and He normal [27] was used to initialize the weights of the 1×1 convolutional layer. The remaining setup was equal to the setup described in section 4.2.4. The layers from the pre-trained model were set as non-trainable.

5.2.5 Evaluation Metrics

The performance of the approaches was evaluated using accuracy (Acc), sensitivity (Sens), specificity (Spec), Mathews Correlation Coefficient (MCC) and area under the ROC curve (AUC). These metrics are described in detail in section 4.2.5. Only the pixels inside the FOV were utilized to compute the metrics.

5.3 Results and Discussion

This section initially presents the comparison between the performance of the Dilate-C model and the proposed approach. Next, the output probability maps of both models are analysed. Lastly, the proposed approach is compared with state-of-the-art methods for retinal vessel segmentation.

5.3.1 Incorporation of ReNet

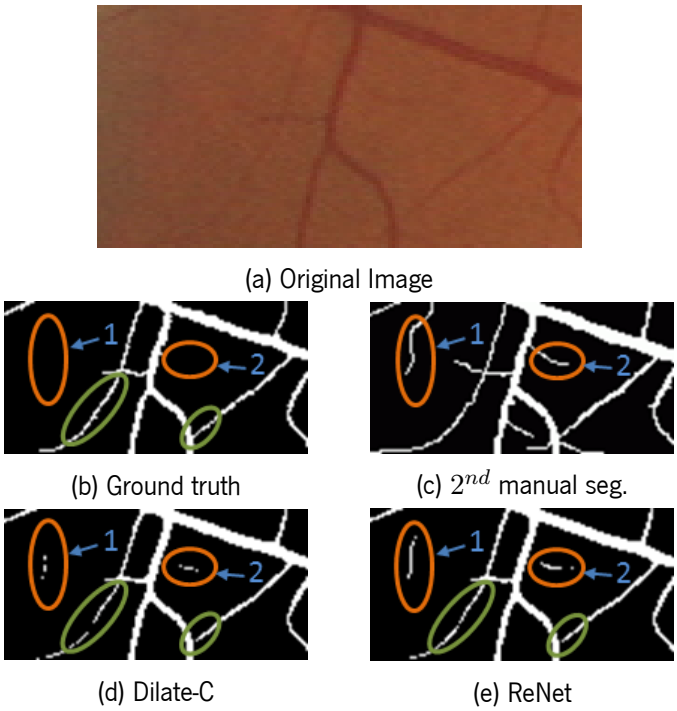
Table 3 contains the results regarding the usage of the ReNet. Some of the changes introduced in images' segmentation are shown in Figures 24 and 25.

Analyzing Table 3, this layer allowed the increase of performance in terms of all metrics, except Spec. Essentially, the global features extracted by ReNet were beneficial for the detection of vessels. The decrease of FN was accompanied by an increment of FP that wasn't prejudicial to the overall results. The standard deviation only increased in two cases and by a small amount, 0.0001. In general, this approach contributed to less variation between subjects results.

Table 3: Segmentation results before and after using ReNet layer on the DRIVE test set. The metrics mean and standard deviation are presented, the later between parenthesis. Values in bold show the best score among all approaches.

Model	Acc	Sens	Spec	MCC	AUC
Dilate-C	0.9569 (0.0043)	0.7931 (0.0553)	**0.9811** (0.0055)	0.8010 (0.0192)	0.9800 (0.0056)
Dilate-C + ReNet	**0.9571** (0.0044)	**0.8003** (0.0543)	0.9803 (0.0054)	**0.8028** (0.0187)	**0.9804** (0.0057)

The detection of more vessels, namely thin vessels, by the Dilate-C + ReNet model can be noted in Figures 24 and 25, when compared with the Dilate-C model. The FP presented in the Figure 24, arrows 1 and 2, were segmented as vessels by the second human observer. Besides, the hemorrhage (Figure 24 - arrow 1) was segmented as vessel in a larger area by the Dilate-C + ReNet model.

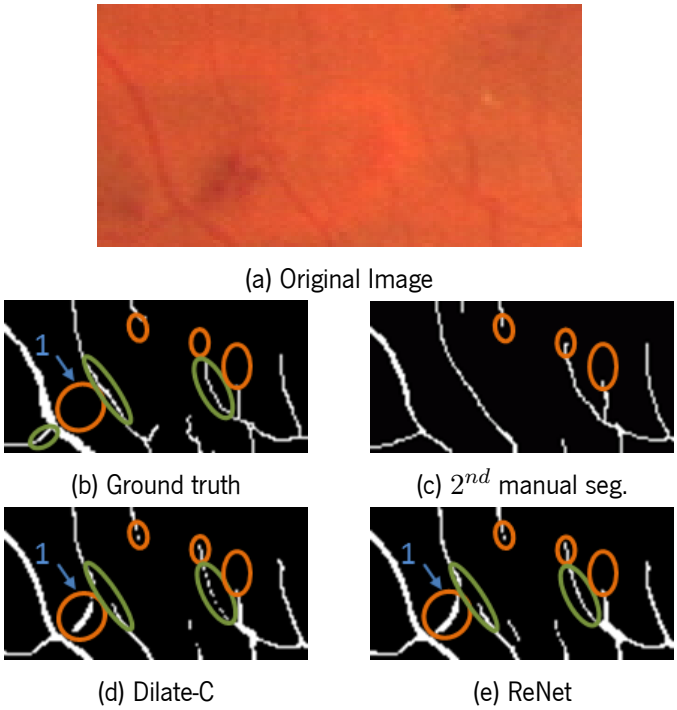


(a) Original Image

(b) Ground truth (c) 2^{nd} manual seg.

(d) Dilate-C (e) ReNet

Figure 24: Pieces of segmentation results from image 19, regarding the incorporation of the ReNet layer. The increase of FP and FN, in relation to the ground truth, are marked in orange and green, respectively.



(a) Original Image

(b) Ground truth (c) 2^{nd} manual seg.

(d) Dilate-C (e) ReNet

Figure 25: Pieces of segmentation results from image 14, regarding the incorporation of the ReNet layer. The increase of FP and FN, in relation to the ground truth, are marked in orange and green, respectively.

In Figures 26 and 27, the output probabilities of the models, with and without ReNet, were examined in order to better understand the effects of using the ReNet layer. The retinal vessel segmentation is a binary problem, so only one probability is needed, the one regarding the vessels class was used.

As previously stated, the ReNet input consists of features extracted from a pre-trained model, set as

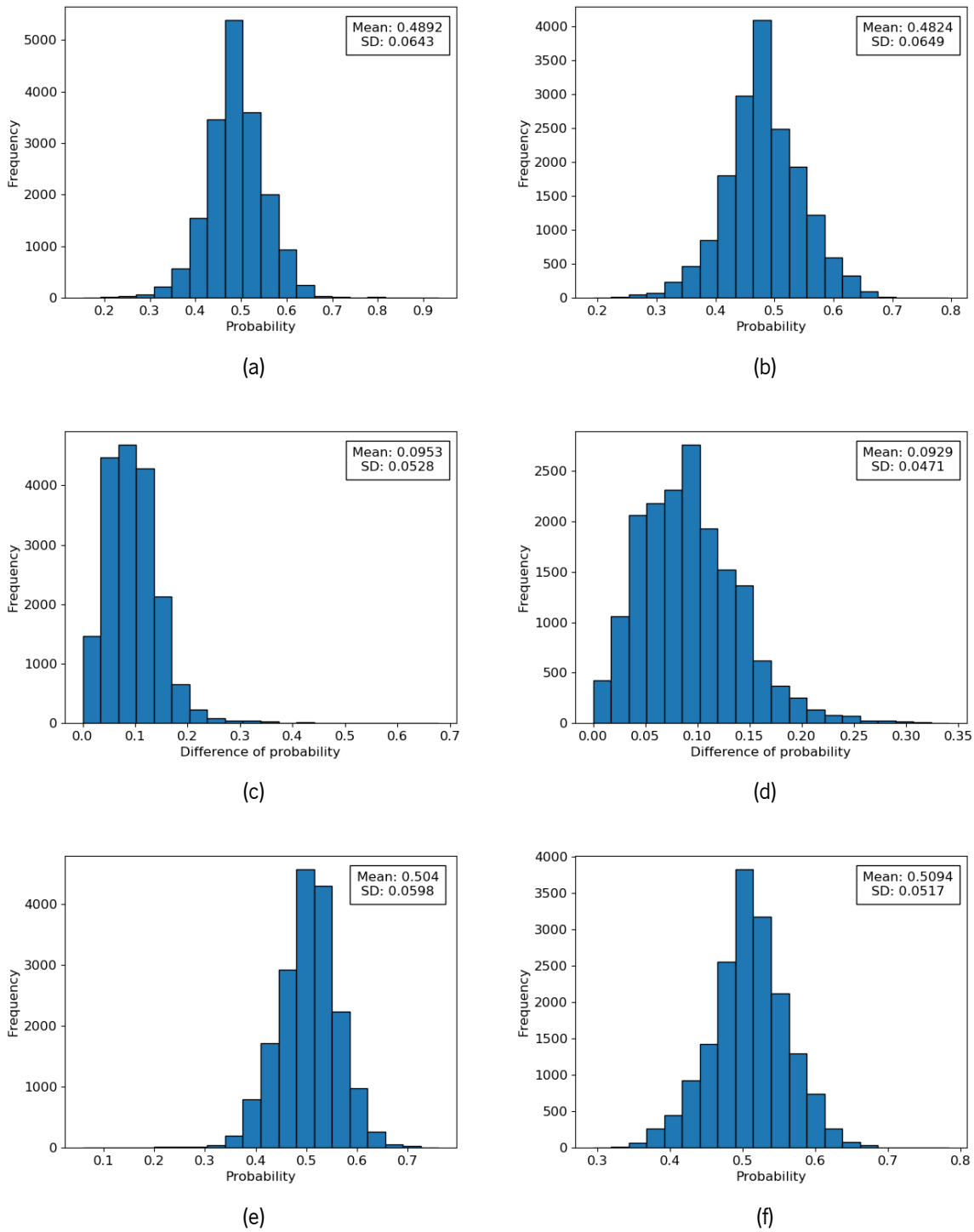


Figure 26: Analysis of the pixels whose classification was altered with the use of ReNet layer. The histograms contain the frequency of the following data: (a) original probability of the corrected pixels, (b) original probability of the wrongly modified pixels, (c) absolute value of the probability variation of the corrected pixels, (d) absolute value of probability variation of the wrongly modified pixels, (e) new probability of the corrected pixels, (f) new probability of the wrongly modified pixels. The mean and standard deviation (SD) of the data are also presented.

non-trainable. This way, ReNet was expected to make small changes in the segmentations. In other words, it should act on pixels with uncertain label, probability close to 0.5, in order to correctly alter their

classification.

Using the ReNet layer, the classification of some pixels was rectified while the classification of other pixels was wrongly modified. As indicated by the performance of the ReNet model, the number of corrected pixels was superior than the number of erroneously altered pixels. Considering all pixels whose classification was altered, the pixels' original probabilities (Figures 26a and 26b), output of Dilate-C model, were indeed concentrated near 0.5. It is noted that the new probabilities (Figures 26e and 26f), output of Dilate-C + ReNet model, are also near this value, but the correctly altered pixels present a greater dispersion. Analyzing the probabilities variation from Dilate-C to Dilate-C + ReNet model (Figures 26c and 26d), it is observed that the highest value of variation is obtained when the ReNet layer rectifies the pixel.

The negative consequences of using ReNet may be related with the size of the receptive field. Some regions of the image may benefit from large receptive field while others don't. Per example, a thin vessel may be neglected if it's totally surrounded by background or if it's located near a large and easily detectable vessel.

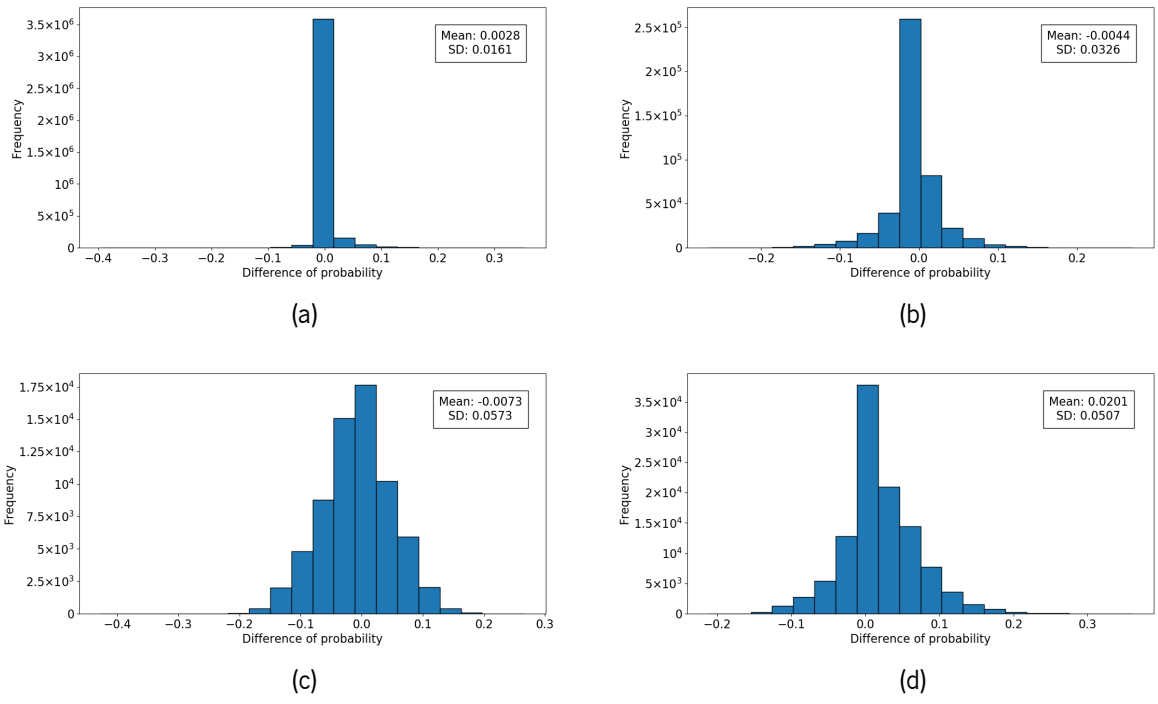


Figure 27: Analysis of the pixels whose classification remained equal with the use of ReNet layer. The histograms contain the frequency of following data: (a) probability variation of the pixels correctly classified as background, (b) probability variation of the pixels correctly classified as vessel, (c) probability variation of the pixels wrongly classified as vessel and (d) probability variation of the pixels wrongly classified as background. The mean and SD of the data are also presented.

The variations of probability were also determined for the pixels with unaltered classification (Figure 27). In this case, a pixel probability was expected to increase when it belongs to a vessel. Similarly, a non-vessel pixel should present a decrease of probability. Analyzing Figure 27, the variations of probability were small, being the mean close to 0 and the maximum standard deviation of 0.0573. Even in a small amount, the probability of a pixel both increased and decreased regardless of its assigned label and ground truth. These observations demonstrate that, for the pixels with unchanged classification, the ReNet had

an neutral effect on the majority of the pixels.

In general, the global information provided by this layer demonstrated to be valuable for the task of retinal vessel segmentation.

5.3.2 Comparison with the state-of-the-art

In Table 4, the Dilate-C + ReNet model is compared with other state-of-the-art methods.

Table 4: Segmentation results of different approaches, including the Dilate-C + ReNet model, on the DRIVE test set. Values in bold show the best score among all methods.

Methods		Year	Acc	Sens	Spec	MCC	AUC
Unsupervised	Badawi and Fraz [38]	2018	0.9547	0.7898	0.9709	-	-
	Aguirre-Ramos et al. [39]	2018	0.9503	0.7854	0.9662	-	-
Supervised	Liskowski and Krawiec [42]	2016	0.9515	0.7520	0.9806	-	0.9710
	Dasgupta and Singh [43]	2017	0.9533	0.7691	0.9801	-	0.9744
	Zhang et al. [40]	2017	0.9466	0.7861	0.9712	0.7673	0.9703
	Mo and Zhang [44]	2017	0.9521	0.7779	0.9780	-	0.9782
	Yan et al. [45]	2018	0.9542	0.7653	**0.9818**	-	0.9752
	Hu et al. [46]	2018	0.9533	0.7772	0.9793	-	0.9759
	Oliveira et al. [47]	2018	**0.9576**	0.8039	0.9804	**0.8054**	**0.9821**
	Wang et al. [41]	2019	0.9541	0.7648	0.9817	0.7851	-
	Guo et al. [48]	2019	0.9561	0.7891	0.9804	0.7964	0.9806
	Shin et al. [49]	2019	0.9271	**0.9382**	0.9255	-	0.9802
	ReNet	2019	0.9571	0.8003	0.9803	0.8028	0.9804

The proposed method obtained the second highest values of Acc and MCC. In relation to these metrics, the approach that stood in first was the one proposed by Oliveira et al. [47], which adds features obtained with Stationary Wavelet Transform to the input. Considering the method of Oliveira et al. [47] without using the extra inputs, the Dilate-C + ReNet model achieves greater Acc. The proposed approach has the third highest Sens, being surpassed by Shin et al. [49] and Oliveira et al. [47]. In Shin et al. [49], the large detection of vessels was accompanied with FPs, resulting in low Spec and Acc. In terms of Spec, it was obtained the sixth best result. Excluding Oliveira et al. [47], all works with superior Spec, had a much lower value of Sens, which indicates that the detection of more background compromised the detection of vessels. The Dilate-C + ReNet model has the third highest AUC, following the works of Oliveira et al. [47] and Guo et al. [48].

5.4 Summary

The use of convolutional layers allows the extraction of local features while ignoring long distance dependencies between pixels. This issue can be surpassed by using ReNet, a layer composed by four RNNs that sweep the image horizontally and vertically.

The proposed approach consisted on incorporating the ReNet layer into a pre-trained model, Dilate-C, in order to improve the performance of the retinal vessel segmentations. During training, the layers from these model was set as non-trainable.

The long term dependencies captured with the ReNet layer were beneficial for the task, allowing the increase of performance in all metrics, except Spec. The proposed approach detected more thin vessels and exacerbated the detection of a hemorrhagic lesion. The decrease of Spec was partially caused by the detection of thin vessels only marked in the annotations not used as ground truth. The effect of ReNet was further analyzed by examining the probabilities of the models with and without ReNet. The ReNet layer altered the classification of pixels with probability close to 0.5. This layer corrected the classification of some pixels but wrongly modified the classification of others. This indicates that the increase of receptive field isn't beneficial for all regions of an image. Regarding the unaltered pixels, the ReNet layer mostly had a neutral effect. The proposed approach achieved Acc, Sens, Spec, MCC and AUC of 0.9571, 0.8003, 0.9803, 0.8028 and 0.9804, respectively. This approach is competitive with state-of-the-art methods for retinal vessel segmentation.

Chapter 6

Test time data augmentation as a learnable technique applied to retinal vessel segmentation

In this chapter, test time data augmentation is studied for the task of retinal vessel segmentation. This technique is applied to the Dilate-C model, the network with higher performance in chapter 4. The chapter starts by presenting the motivation for employing test time data augmentation. Next, the experimental setup contains a description of the database, data pre-processing, architectures, training settings and evaluation metrics. Lastly, this chapter presents the results and discussion.

6.1 Motivation

Data augmentation techniques are extensively used in order to increase the amount of training data and, consequently, improve the performance of deep learning models [5]. These techniques are applied in several areas, including medical image processing.

Test time data augmentation takes advantage of the context information encoded in the network, when using data augmentation [47]. This approach consists of applying the transformations used during training to the test set. The final segmentation of a test image commonly results from averaging the probability maps computed for the original image and for the respective transformed copies. Wang et al. [86] applied test time data augmentation in a different form. Instead of averaging the output probability maps, the final segmentation was obtained by majority voting. Although the best way of combining the multiple segmentations isn't defined, the application of this technique proved to be valuable in medical image segmentation tasks, namely, delineation of brain tumor substructures [86, 87] and retinal vessels [47].

In a work not related to test time data augmentation, Xia et al. [88] started by employing three 2D networks to segment abdominal organs in coronal, sagittal and axial views of computed tomography volumes, respectively. The novelty of their work lies on using a 3D U-Net to fuse the stacked 2D segmentations. Fundamentally, this corresponds to utilizing a neural network to merge multiple segmentations of the same example.

In this work, test time data augmentation is applied in two forms. The probability maps of the original

image and the respective rotated copies are merged by averaging. The second form was based on the work of Xia et al. [88] and consists of using a network to learn how to combine the multiple segmentations. Although different types of networks were tested, only the results for ConvLSTM are presented. The use of ConvLSTM allows the segmentation to be recalibrated taking into account the multiple segmentations and the context infomation.

6.2 Experimental Setup

This section contains the implementation details. The deep learning models were implemented in Python using Keras [75] with TensorFlow [76] backend.

6.2.1 Databases

The DRIVE database [77] was utilized to evaluate the models. A more detailed description of this database can be found in section 4.2.1. In summary, DRIVE contains 40 images equally divided into two sets, training and test. The manual segmentations from the first human observer were used as ground truth. As the database doesn't provide a validation set, two images from the training set, subjects 34 and 40, were randomly chosen.

6.2.2 Pre-processing and patch extraction

Regarding pre-processing, the green channel is extracted from the retinal fundus images and normalized using the statistics, mean and standard deviation, of the training set. In the training phase, 500 patches are randomly selected from each training image to be used in the learning process. During training and test, each patch is replicated three times, originating 4 equal patches that are rotated 0 (no rotation), 90, 180 and 270 degrees, respectively. It should be noted that the rotation operations are equal to the ones applied to the training patches of the Dilate-C model.

6.2.3 Network Architectures

The test time data augmentation approaches were implemented within the network as shown in Figure 28 and the Dilate-C model was used as baseline. As mentioned in chapter 4, the Dilate-C model was trained with batch size of 4 and its output has two feature maps. Thus, the output of this network must suffer some transformations in order to apply test time data augmentation. The transformations are applied by the layer denominated as *Patches preparation*. First, it removes the first feature map, which contains the probabilities regarding the background class. Second, it rotates each element of the batch in order to align the four feature maps. Lastly, it does alterations on the dimensions, according with the following layer. If the next layer is convolutional, it permutes two dimensions, batch size and channels. If the next layer is a ConvLSTM, a new dimension is created, replacing the batch size.

In the first approach, Averaging, the probability maps are averaged in order to obtain the final segmentation. This approach was implemented using a 1×1 convolutional layer with pre-defined and equal weights, whose values were set as 0.25, and no training was required. In this network, the *Matching Prob.* layer calculates the complementary probabilities in order to obtain an output with two feature maps, containing the probabilities of each pixel belonging to the background and vessels classes.

The second approach consists of using a ConvLSTM with a 3×3 kernel to learn how to combine the multiple probability maps.

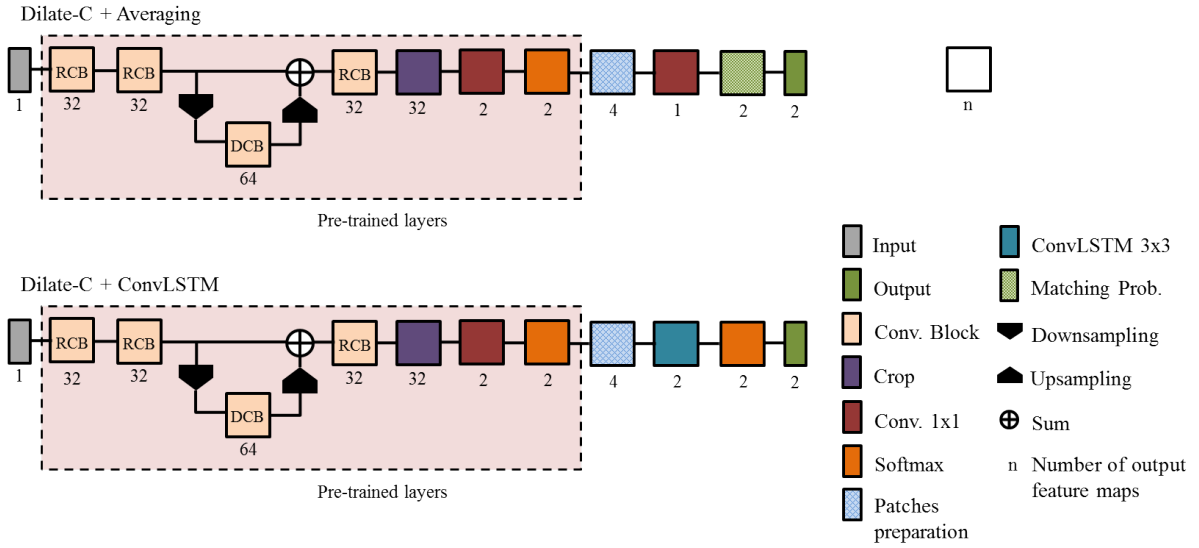


Figure 28: Architectures of the proposed test time data augmentation models, Dilate-C+Averaging and Dilate-C+ConvLSTM models.

6.2.4 Training settings

Although the batch size is 4 at the network input, the insertion of a dimension assures that, in practice, a batch size of 1 is used to train the ConvLSTM. This model is trained for 30 epochs and Glorot uniform [26] was used to initialize its weights. Categorical cross-entropy and Adam [9] were chosen as loss function and optimizer, respectively. The learning rate was set as constant and equal to 5×10^{-5}.

6.2.5 Evaluation Metrics

The models performance was evaluated using multiple metrics, namely, accuracy (Acc), sensitivity (Sens), specificity (Spec) and Mathews Correlation Coefficient (MCC). Section 4.2.5 contains a detailed description of these metrics. The computation of these metrics only took into account the pixels inside the FOV. In contrast with the chapters 4 and 5, the AUC wasn't presented due to technical issues.

6.3 Results and Discussion

In this section, the two approaches used for test time data augmentation and the baseline model are evaluated. This section also presents a comparison between state-of-the-art methods and the proposed approach with better performance.

6.3.1 Application of test time data augmentatation

The results concerning the usage of test time data augmentation are shown in Table 5. Figures 29 and 30 contain segmentations in which the main differences, in relation to the ground truth, are marked.

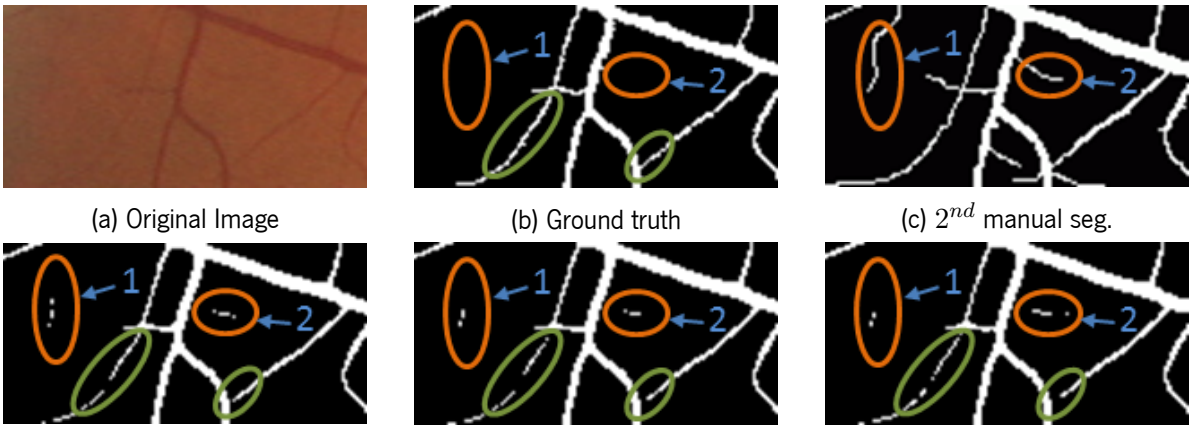
Table 5: Segmentation results before and after applying test time data augmentation on the DRIVE test set. The metrics mean and standard deviation are presented, the later between parenthesis. Values in bold show the best score among all approaches.

Model	Acc	Sens	Spec	MCC
Dilate-C	0.9569 (0.0043)	0.7931 (0.0553)	0.9811 (0.0055)	0.8010 (0.0192)
Averaging	0.9572 (0.0045)	0.7932 (0.0564)	0.9814 (0.0055)	0.8023 (0.0198)
ConvLSTM	**0.9575** (0.0045)	**0.7938** (0.0561)	**0.9817** (0.0054)	**0.8038** (0.0194)

Regardless of the merging approach, the use of test time data augmentation allowed a general increase of performance. The increase of TP wasn't accompanied by the increment of FP, since this technique allowed simultaneously the detection of more vessels and background, which is noticed by the values of Sens and Spec. According to Table 5, the dispersion of results is slightly higher when using test time data augmentation, principally in terms of sensitivity and excluding Spec. Contrary to the quantitative results, test time data augmentation detected less vessels in Figures 29 and 30. Furthermore, comparing Figures 30e and 30f with Figure 30d, the hemorrhage area wrongly segmented as vessel increased or remained unchanged. In the segmentation pieces presented, there are few and small changes consistent with the results and are all related with the detection of less FP. These changes are in Figure 29 arrow 1 and Figure 30 arrow 2 and 3. It should be noted that the selected images mainly contain thin vessels. This may indicate that test time data augmentation could be more beneficial for vessels with large and medium width.

Comparing the two forms of combining the segmentation results, ConvLSTM outperformed the averaging in all metrics. Furthermore, the variation of metrics between subjects is minor when the network executes the task. The use of ConvLSTM allowed the detection of a larger part of a thin and low contrast vessel, only delineated by the 2^{nd} human observer (arrow 2 in Figure 29). Regarding Figure 30, the network detects more vessels and lesion (arrow 1).

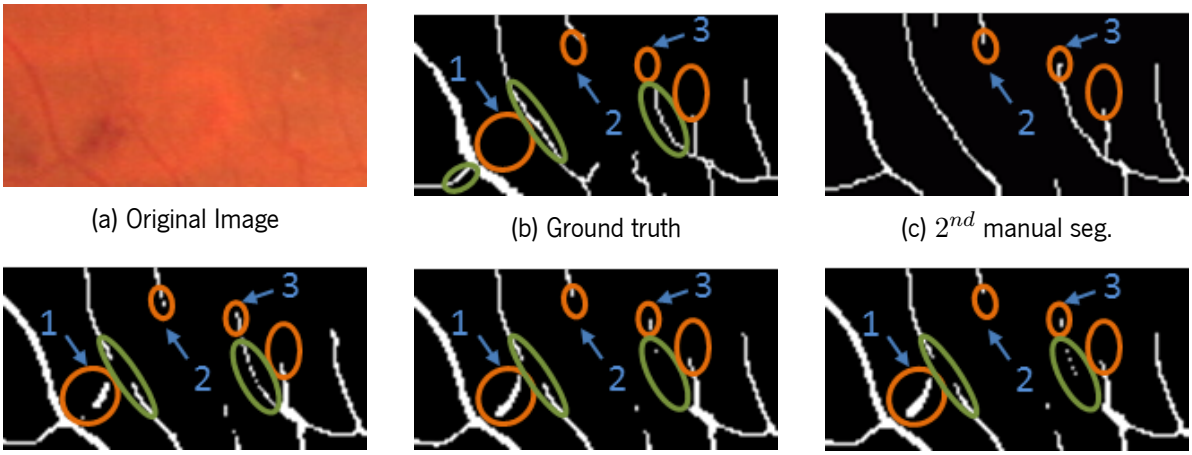
The use of test time data augmentation proved to be beneficial for retinal vessel segmentation, specially when the merging approach was performed by a ConvLSTM.



(a) Original Image (b) Ground truth (c) 2^{nd} manual seg.

(d) Dilate-C (e) Averaging (f) ConvLSTM

Figure 29: Pieces of segmentation results from image 19, regarding the application of test time data augmentation techniques. The increase of FP and FN, in relation to the ground truth, are marked in orange and green, respectively.



(a) Original Image (b) Ground truth (c) 2^{nd} manual seg.

(d) Dilate-C (e) Averaging (f) ConvLSTM

Figure 30: Pieces of segmentation results from image 14, regarding the application of test time data augmentation techniques. The increase of FP and FN, in relation to the ground truth, are marked in orange and green, respectively.

Lastly, it was shown that the combination of multiple segmentations originated by test time data augmentation can be performed by a neural network.

6.3.2 Comparison with the state-of-the-art

The proposed approach for retinal blood vessel segmentation is compared with state-of-the-art methods in Table 6.

The combination of Dilate-C model with test time data augmentation using ConvLSTM allowed to achieve the second best performance in terms of Acc and MCC. In terms of these metrics, Oliveira et al. [47] obtained the highest values. When excluding the use of wavelets as input in the work of Oliveira et al. [47], the proposed method obtains greater Acc. Two approaches, Shin et al. [49] and Oliveira et al. [47], outperformed the proposed method in terms of Sens. Shin et al. [49] detect an immense amount of blood

vessels when compared with the other state-of-the-art methods, namely, the Sens difference between this method and the one that stood in second is greater than 10%. Analyzing the Acc and Spec of this work, it's noted that the number of FPs is also significantly large. In relation to Spec, the proposed method has the same performance as Wang et al. [41] and is surpassed by Yan et al. [45]. These approaches have a smaller Sens and Acc, so the detection of more background negatively affected the detection of vessels.

The test time data augmentation approaches were also applied to the ReNet model, which was presented in the previous chapter. This approach didn't allow any improvements on the performance, therefore the respective results aren't shown. Ultimately, it should be noted that test time data augmentation allowed to achieve better performance in terms of Acc, Spec and MCC, when compared with the ReNet model.

Table 6: Segmentation results of different approaches, including the Dilate-C model combined with test time data augmentation, on the DRIVE test set. Values in bold show the best score among all methods.

Methods		Year	Acc	Sens	Spec	MCC
Unsupervised	Badawi and Fraz [38]	2018	0.9547	0.7898	0.9709	-
	Aguirre-Ramos et al. [39]	2018	0.9503	0.7854	0.9662	-
Supervised	Liskowski and Krawiec [42]	2016	0.9515	0.7520	0.9806	-
	Dasgupta and Singh [43]	2017	0.9533	0.7691	0.9801	-
	Zhang et al. [40]	2017	0.9466	0.7861	0.9712	0.7673
	Mo and Zhang [44]	2017	0.9521	0.7779	0.9780	-
	Yan et al. [45]	2018	0.9542	0.7653	**0.9818**	-
	Hu et al. [46]	2018	0.9533	0.7772	0.9793	-
	Oliveira et al. [47]	2018	**0.9576**	0.8039	0.9804	**0.8054**
	Wang et al. [41]	2019	0.9541	0.7648	0.9817	0.7851
	Guo et al. [48]	2019	0.9561	0.7891	0.9804	0.7964
	Shin et al. [49]	2019	0.9271	**0.9382**	0.9255	-
	ReNet	2019	0.9571	0.8003	0.9803	0.8028
	ConvLSTM	2019	0.9575	0.7938	0.9817	0.8038

6.4 Summary

Data augmentation techniques are extensively used to improve the performance of deep learning models. The information encoded by the network during training can also be used during test. This technique is called test time data augmentation. In practice, the transformations applied to the training data are also applied to the test data, originating multiple segmentations of one data sample. Usually, the final segmentation is obtained by averaging the multiple probability maps.

In this chapter, the proposed approach consisted in using a ConvLSTM to learn how to merge the multiple probability maps, taking into account, not only the pixel but also its context. Furthermore, averaging was also applied. Test time data augmentation was applied to the Dilate-C model and used for retinal

vessel segmentation.

Regardless of the merging approach, test time data augmentation allowed the increase of performance in terms of all metrics. Despite the increased Sens and Spec, these models detected less thin vessels and more hemorrhage. When comparing the two forms of test time data augmentation, the ConvLSTM obtained greater performance and detected more thin vessels. The proposed approach achieved Acc, Sens, Spec and MCC of 0.9575, 0.7938, 0.9817 and 0.8038, respectively. The obtained results are competitive with state-of-the-art methods.

Chapter 7

Study of Deep Layer Agregation architecture for brain tumor segmentation

In this chapter, the Deep Layer Aggregation structures are studied for brain tumor segmentation. First, the motivation for using this architecture is presented. The next section describes the experimental setup, which includes the database, data pre-processing, architectures, training settings and evaluation metrics. Lastly, this chapter contains the results and discussion regarding the implemented architectural changes.

7.1 Motivation

The U-Net [3] has been applied to several medical image segmentation tasks and, in most cases, it has achieved superior performance when compared to other FCN architectures. Notwithstanding, Yu et al. [19] claims that U-Net consists of a shallow aggregation, since the shallowest features are aggregated in a single step, which doesn't allow their refinement and may impair the inference of location. In order to solve this issue, they proposed a structure, denominated iterative deep aggregation (IDA), that progressively fuse the features with different scales. Additionally, they proposed the hierarchical deep aggregation (HDA) structure, responsible for merging features with equal resolution but different complexity, in order to improve recognition. Both structures are presented in Figure 31. The final architecture, resulting from the combination of the mentioned structures, is termed as Deep Layer Aggregation (DLA).

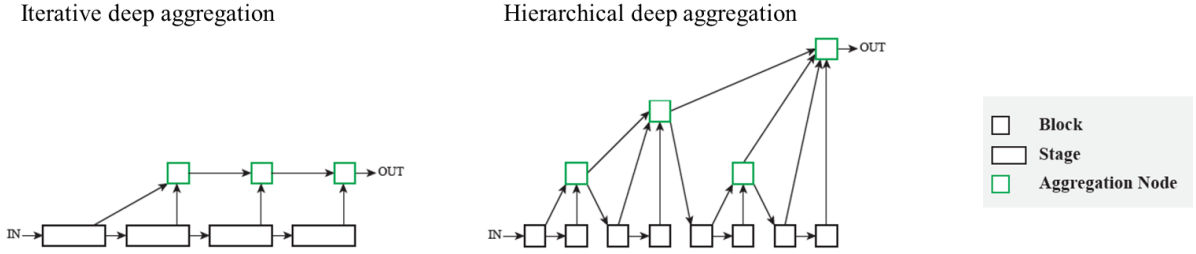


Figure 31: Iterative Deep Aggregation and Hierarchical Deep Aggregation structures. While stages differ in terms of features resolution, blocks have features with equal resolution. Reproduced from Yu et al. [19].

Skip connections are known to be valuable for semantic segmentation since the creation of FCNs and

they are utilized in U-Net and DLA architectures. Unlike U-Net, DLA progressively aggregates low level information, preventing it from skipping directly to the deepest layers of the network. This raises one question: Can a long skip connection improve the DLA architecture?

Dilated convolutions are suitable for replacing downsampling steps, as shown in chapter 4. They allow the increase of context information while keeping the resolution of feature maps. Dilated convolutions can be used in a simpler way. In the works of Devalla et al. [89] and Lopes et al. [4], dilated convolutions are utilized instead of standard convolutions, augmenting the receptive field of the network. Regarding the results, both medical image segmentation tasks obtained a higher performance when this substitution was applied.

In this work, the performances of U-Net and DLA are compared. Then, long skip connections were incorporated into DLA, allowing to evaluate the impact of low level information on segmentation performance. Other two modifications were applied, replacement of standard convolutions with dilated convolutions and substitution of a downsampling step with dilated convolutions. These studies were made for the task of brain tumor segmentation using the BRATS 2017 database.

It should be noted that this architecture was also tested on retinal vessel segmentation, using the DRIVE database. The results aren't presented since training was unstable, despite experimenting diverse values of dropout and L^2 regularization, and it led to poor performance. This may be caused by the considerable depth of DLA, which may result in the lost of details, namely, thin vessels.

7.2 Experimental Setup

This section presents the implementation details. The deep learning methods were implemented using Keras [75] with TensorFlow [76] backend, libraries in Python.

7.2.1 Database

The Brain Tumor Segmentation Challenge (BRATS) 2017 database [57, 90] consists of two datasets, Training and Leaderboard. The Training dataset contains 210 cases of HGG, glioblastoma, and 75 subjects diagnosed with LGG, astrocytoma or oligoastrocytoma. The Leaderboard dataset contains 46 subjects. The MRI sequences available for each subject are T1, T1c, T2 and FLAIR, which have 1 mm^3 resolution. The sequences are aligned and skull-stripped. Different clinical protocols and scanners were used to acquire the mentioned MRI scans. The manual segmentations are only provided for the Training dataset. The experts annotated four regions, namely, peritumoral edema, non-enhancing tumor and necrosis, enhancing tumor and healthy tissues. The labeled data, Training dataset, was divided into three sets, namely, training, validation and test. Each set contains 60%, 20% and 20% of the subjects, respectively.

7.2.2 Pre-processing, patch extraction and post-processing

Three pre-processing techniques were applied to the data, namely, correction of the bias field [91], standardization of the intensity histogram of each MRI sequence [92] and normalization of each sequence

using the statistics, mean and standard deviation, of the training set.

In the training phase, 52 patchs per subject are extracted from the axial view of the MRI sequences.

The training data is augmented through operations of flipping and rotation. There are four possible rotations 0 (no rotation), 90, 180 and 270 degrees. The same patch may be flipped and rotated, experience only one of the transformations or stay unaltered. Throughout training, the original and transformed patchs are randomly presented to the network. The flipping transformation happens according to a binomial distribution with $p = 0.5$. The angle of rotation is randomly chosen, no probability distribution was defined.

The input patch size was 104×104 and the correspondent labeled patch was 32×32.

Regarding post-processing, a morphological filter was applied, using the binary segmentations of the whole tumor originated by Pereira et al. [93]. The connected components that don't intercept the binary segmentation are eliminated.

7.2.3 Network Architectures

The implemented networks are based on the work of Yu et al. [19]. The Deep layer aggregation is composed by two fundamental structures, namely, hierarchical deep aggregation (HDA) and iterative deep aggregation (IDA). The contracting path consists of HDA structures and downsampling operations applied alternately. The HDA merges semantic information with different complexity. In this work, HDAs only have one level, so a convolutional layer is responsible for fusing the features provided by two consecutive convolutional blocks. At each downsampling, the resolution is halved and the number of feature maps is doubled. Then, IDA structures progressively fuse the features with different resolution from the contracting path. Each of these structures have two inputs that are merged by a convolutional layer. In practice, the lower resolution input is first modified to match the second input, namely, its feature map number is halved and it's upsampled. Lastly, a 1×1 convolution followed by softmax is applied to obtain the probability distribution over the classes and compute prediction.

The baseline network along with the HDA and IDA blocks are presented in Figure 32. The HDA block is composed by two convolutional blocks and an aggregation node. In this block, the aggregation node consists of concatenation followed by 1×1 convolution, Batch Normalization (BN) and Rectified Linear Unit (ReLU). The downsampling operation is performed by a convolution with 2×2 kernel and stride 2. The IDA block first acts upon the lower resolution feature map, using 1×1 convolution, BN, ReLU, Dropout and transposed convolution with 3×3 kernel and stride 2. The latter operation is responsible for upsampling. Then, the IDA block contains an aggregation node similar to the one previously described, the only difference is in the convolutional layer which, in this case, has a 3×3 kernel. The convolutional blocks used in the baseline model incorporate two convolutional layers with 3×3 kernel and a residual connection. Each convolutional layer is followed by BN, ReLU and Dropout. These type of convolutional block is denominated as residual convolutional block (RsCB). In Figure 33, the convolutional blocks utilized in this chapter are shown.

Nine models, including the Baseline, were studied for brain tumor segmentation. These models may be organized in three groups, according to their modifications. The first group contains 3 models in

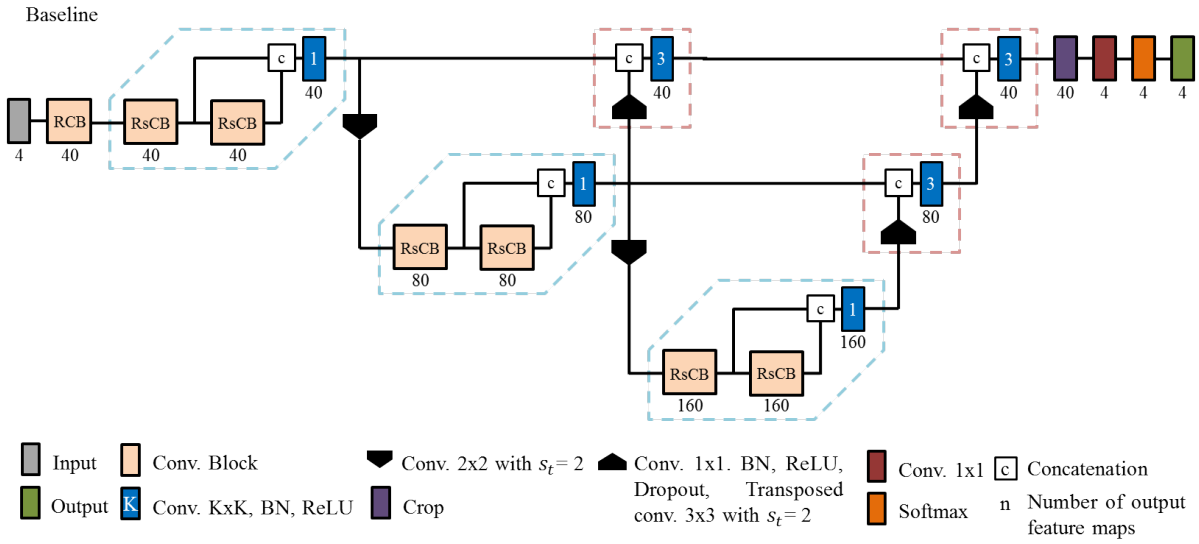


Figure 32: Architecture of the Baseline model. The HDA and IDA blocks are delimited by a blue and pink dashed line, respectively.

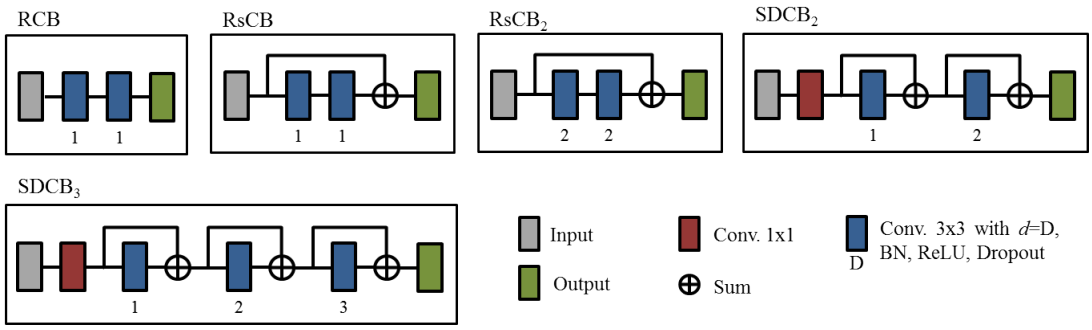


Figure 33: Convolutional blocks.

which one or two long skip connections are incorporated into the Baseline model (Figure 34). In the 4 models of the second group, the residual blocks are partially or totally replaced by blocks containing dilated convolutions, namely $SDCB_2$, $SDCB_3$ and $RsCB_2$, as shown in Figure 35. It should be noted that a simplified version of DCBs, which were proposed in chapter 4, was used due to the elevated number of parameters associated with this architecture. In the last model one downsampling is eliminated and simplified DCBs with maximum dilation rate of 3 are used within the HDA structures (Figure 36).

7.2.4 Training settings

The models were trained for 143 epochs with a batch size of 4. The weights matrix of convolutional layers were initialized with He normal [27]. Categorical cross-entropy and Adam [9] were chosen as loss function and optimizer, respectively. The learning rate was set as constant and equal to 5×10^{-5}. Dropout was applied with a probability of 0.05.

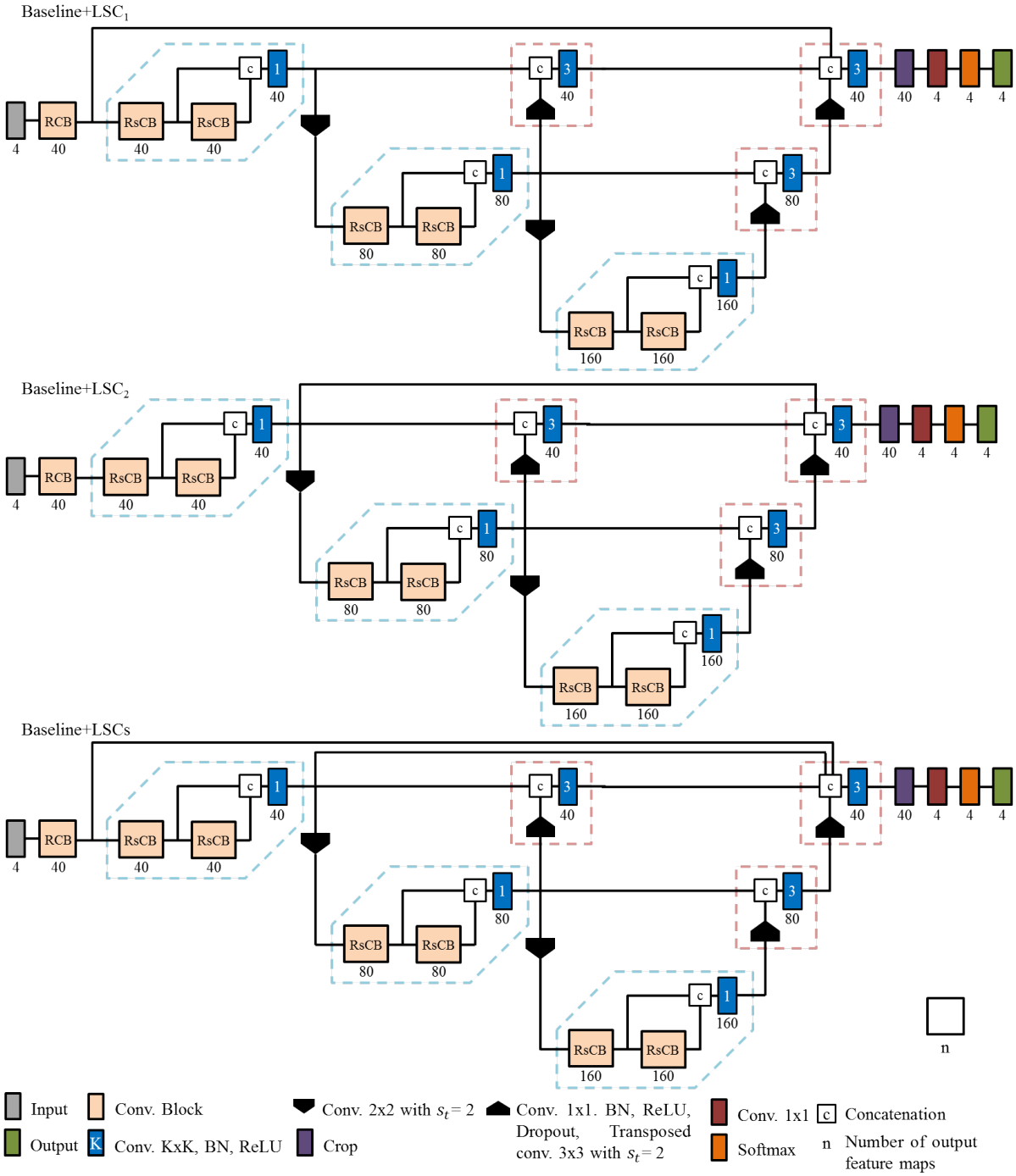


Figure 34: Architectures of the models utilized to study the incorporation of long skip connections (LSCs). The HDA and IDA blocks are delimited by a blue and pink dashed line, respectively.

7.2.5 Evaluation Metrics

The metrics of the Leaderboard set were provided by the organizers of the Challenge. The model's performance is quantified using sensitivity (Sens), Dice Similarity Coefficient, usually denominated as Dice, and robust Hausdorff Distance (HD_{95}).

Sens measures the portion of positives correctly identified, quantifying the model's capacity of detect-

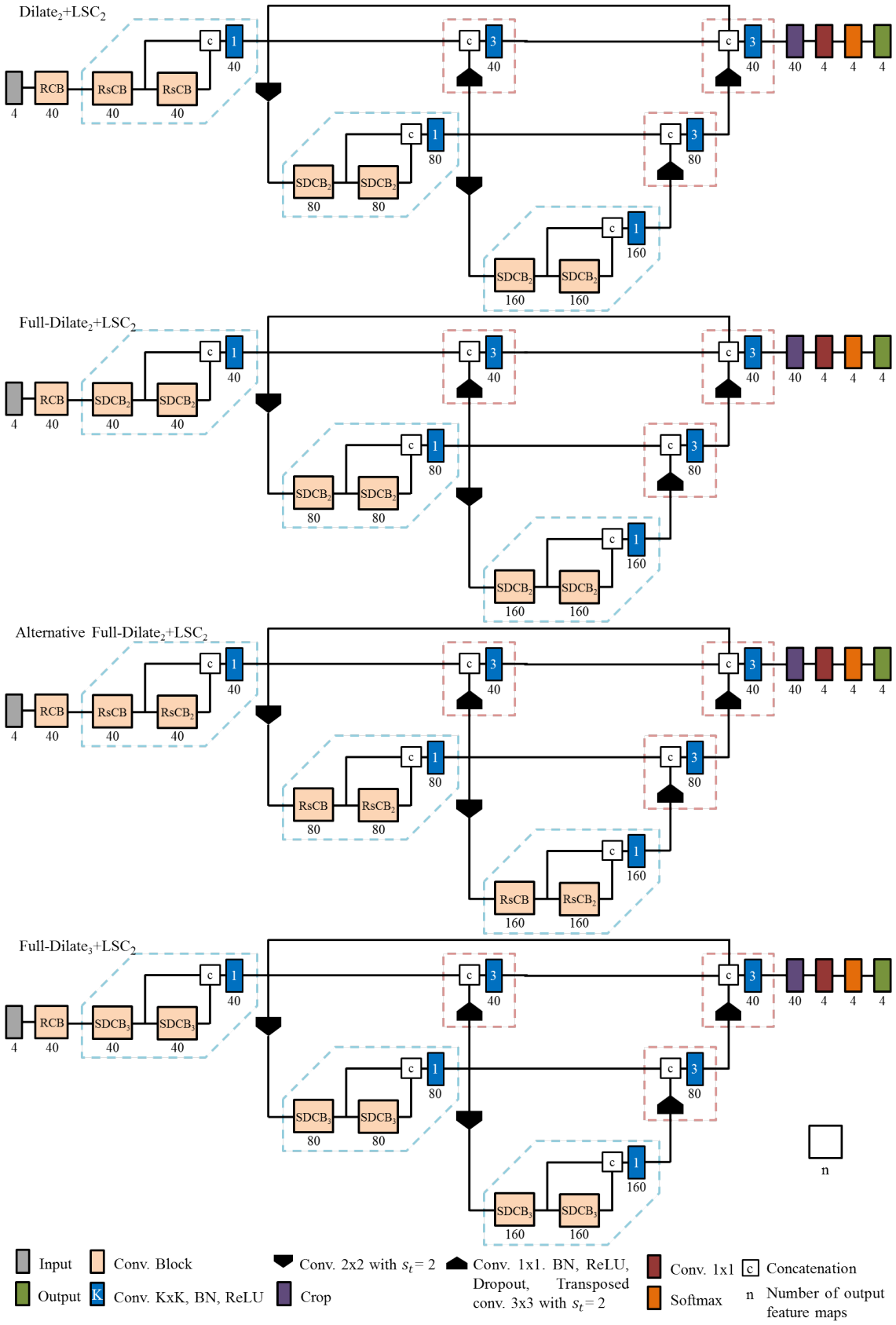


Figure 35: Architectures of the models used to study the replacement of standard convolutions with dilated convolutions. The HDA and IDA blocks are delimited by a blue and pink dashed line, respectively.

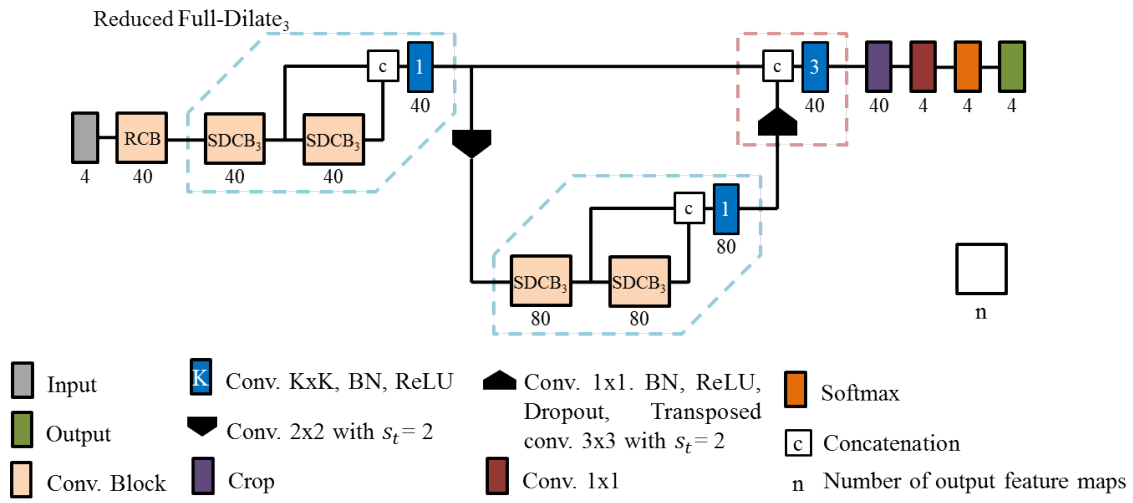


Figure 36: Architecture of the model used to study the replacement of a downsampling step with dilated convolutions. The HDA and IDA blocks are delimited by a blue and pink dashed line, respectively.

ing the structures of interest. As previously stated, this metric is defined as follows [81],

$$\text{Sens} = \frac{TP}{TP + FN} \tag{44}$$

Dice Similarity Coefficient [94] measures the spatial overlap between the output segmentation and the ground truth. It can also be defined as the harmonic mean between precision and Sens. This performance metric is computed as

$$\text{Dice} = \frac{2TP}{2TP + FP + FN} \tag{45}$$

Hausdorff Distance (HD) measures the maximum distance between the surfaces of manual and automatic annotations, represented by (T) and (P), respectively. This distance is calculated as

$$\text{HD} = \max\{\sup_{p \in P} \inf_{t \in T} D(p,t), \sup_{t \in T} \inf_{p \in P} D(t,p)\} \tag{46}$$

where $D(t,p)$ is the least-squares distance distance between t and p, points of each surface. This metric is susceptible to small outlying subregions. The robust Hausdorff Distance corresponds to the 95% quantile of the maximum distance [57].

During metrics computation, the tumor substructures are grouped, forming three mutually inclusive regions, namely, complete, core and enhancing tumor. The complete tumor includes all tumor substructures. The core tumor includes both non-enhancing tumor, enhancing tumor and necrosis. The last region corresponds to the enhancing tumor, as indicated by its designation.

7.3 Results and Discussion

This section starts by comparing two main architectures, U-Net and DLA. Then, the analysis of the modifications employed in the DLA architecture is presented. First, the importance of low level information

for brain tumor segmentation is evaluated. Then, the increase of receptive field and the augmentation of the minimum resolution are examined. Lastly, the best performing approach is compared with state-of-the-art methods.

7.3.1 Comparison between U-Net and DLA architectures

In Table 7, the baseline model, a DLA-based architecture, is compared with an U-Net based model, proposed by Pereira et al. [70]. When DLA structures are employed, the larger tumor regions, complete and core, present better results in terms of Dice and HD_{95}. The core substructure has the highest improvements, Dice and HD_{95} changed by 1.6% and 1.5, respectively. The enhanced tumor region also benefited, the HD_{95} suffered a reduction of about 0.8 while Dice remained unchanged. The DLA model showed better performance on the task of brain tumor segmentation.

It should be noted that the U-Net [70] has three levels, in which 40, 80 and 160 features are extracted by a convolutional layer, just like the baseline model. Furthermore, U-Net training was executed as described in section 7.2.2.

It should also be pointed out that the DLA model contains 1.4 million parameters, while U-Net [70] contains about 1 million. Thus, the number of parameters could be responsible for the higher performance of DLA model. Pereira et al. [70] also trained a wider U-Net , containing aproximately the same number of parameters as the baseline model, but only the results for the test set were presented. Taking into account that the wider U-Net achieved slightly worse performance on the test set when compared with the original U-Net, the higher performance obtained by the DLA model is not related with the higher number of parameters.

Table 7: Segmentation results obtained by U-Net and DLA baseline models on the Leaderboard set. Values in bold show the best score among all approaches. Sens values weren't available in Pereira et al. [70].

Model	Dice			HD_{95}		
	Comp.	Core	Enh.	Comp.	Core	Enh.
Baseline	**0.8916**	**0.7743**	**0.7190**	**6.43**	**9.58**	**4.91**
U-Net [70]	0.8890	0.7580	**0.7190**	6.58	11.1	5.74

7.3.2 Incorporation of long skip connections: importance of low level information

The results obtained with the insertion of long skip connections are shown in Table 8.

Incorporation of LSC_2 Analyzing Table 8, the use of the long skip connection allowed the detection of a larger area of enhanced region, since Sens increased in about 2.5%. As this increase was accompanied with a significant augment of Dice (2.7%), the number of false detections remained approximately unchanged. The increase of HD_{95} indicates that the delineation of the enhanced region slightly worsened. Regarding the complete, the same analysis can be done but the increase of Dice and Sens was significantly

lower, about 0.3%. In relation to core, the high augment of TP, increase of Sens by 3%, was achieved at the expense of false detections, noticed in the small decrease of Dice. Observing the value of core HD_{95}, the false detections occur far from the delineation of this region. In general, the low level information is beneficial for all tumor substructures.

Incorporation of LSC_1 The information codified by the first blocks of a network roughly corresponds to the image gradients. In architecture Baseline+LSC_1, the gradient information is assembled with high level information, resulting in the higher detection of active tumor region (enhanced tumor). This elevated improvement of Sens, roughly 2.5%, and the slight augment of Dice (0.5%) indicate the decrease of Spec. Although the number of false detections has enlarged, the Hausdorff distance practically remained unchanged, pointing out that the FPs introduced by the use of the skip connection are situated near the contour of the enhancing tumor. The complete structure has a higher Dice and lower Sens and HD_{95}. This demonstrates the reduction of both true and false detections. The latter are responsible for the improvement of contour definition. Regarding the core substructure, the significant augment of Sens was accompanied by a meaningful increase of false detections, that were detrimental for HD_{95}. The use of this skip connection was, in general, valuable for the task of tumor segmentation.

Comparison of LSCs Comparing the two low level informations and observing the values of Hausdorff distance, it's noted that the gradients of the image give relevant information regarding the location of tumor substructures border. However, the detection of the regions, in terms of Dice and Sens, is favored when using the shorter skip connection, i.e., the information of deeper layers. Considering this observation and the fact that the HD_{95} margin between these models is small, the model Baseline+LSC_2 presents better performance.

Junction of LSCs In order to try to take advantage of the positive aspects of each skip connection, Baseline+LSC_1 and Baseline+LSC_2 models were merged. Thus, Baseline+LSCs model contains both skip connections transferring low level information, with different complexities, to a deeper layer. As expected, the combination of skip connections allowed the increase of Sens of all substructures, specially core (4%) and enhanced (4.5%). However, the performance in terms of Dice and HD_{95} massively decreased, also, for the three subregions. Taking this into account two conclusions may be drawn. First, the augment of true detections was achieved at the expense of false detections. Second, the FP are located far from the delineation. It should be noted that the three sets of feature maps are merged with only a convolutional layer with a 3×3 kernel, which may be insufficient to correctly extract all relevant information.

7.3.3 Incorporation of dilated convolutions

In Table 9, the results regarding the usage of dilated convolutions are presented.

Incorporation of $SDCB_2$'s Dilated convolutions, with maximum d equal to 2, were inserted in the lower levels of the $Dilate_2$+LSC_2 model to increase the receptive field. This increase led to an augment of

Table 8: Segmentation results concerning the incorporation of long skip connections obtained on the Leaderboard set. Values in bold show the best score among all approaches.

Model	Dice			Sens			HD_{95}		
	Comp.	Core	Enh.	Comp.	Core	Enh.	Comp.	Core	Enh.
Baseline	0.8916	**0.7743**	0.7190	0.8939	0.7597	0.7503	6.43	**9.58**	**4.91**
Baseline+LSC_2	**0.8944**	0.7740	**0.7460**	0.8967	0.7803	0.7747	6.51	11.18	4.96
Baseline+LSC_1	0.8926	0.7685	0.7238	0.8886	0.7714	0.7742	**6.31**	10.90	4.93
Baseline+LSCs	0.8761	0.7331	0.7123	**0.9106**	**0.7996**	**0.7965**	11.74	18.64	12.46

Sens (1%) and reduction of Dice (3.6%) and HD_{95} of the inner subregion. Although the increase of true detections was accompanied with a great augment of false detections, these FPs are placed closer to the delineation of the enhancing tumor. Regarding the core, all metrics improved, specially the HD_{95} by 2.2. This indicates that this model more easily detects and outlines this region, when compared with the Baseline+LSC_2 model. In relation to complete tumor, the only metric that improved was HD_{95}. Sens had the higher modification and it only decreased by 0.7%. This decrease of TP caused the reduction of Dice. In general, the augment of context information was beneficial for the segmentation of tumor substructures, specially, the core region.

Extension of $SDCB_2$'s usage The insertion of dilated convolutions improved the performance of the model, so its use was extended. In Full-$Dilate_2$+LSC_2 model, dilated convolutions were used in all levels, further increasing the receptive field. The augment of context information improved the performance of all metrics for all tumor substructures, except the Sens of core, which remained unaltered. Analyzing the enhancing region, the increase of Dice by 2.7% was mainly caused by the reduction of false detections, since Sens only had a small increase (0.2%). This improvement also contributed to approximate the generated delineation to the real one. A similar analysis can be done to the core structure, the number of FPs significantly reduced. Regarding the complete tumor, the augment of Sens was much higher than the increase of Dice. Therefore, the number of false detections enlarged but they are located close to the boundary. It should be noted that the performance in terms of complete's Dice and Sens, previously reduced from Baseline+LSC_2 to $Dilate_2$+LSC_2 model, was totally recovered in Full-$Dilate_2$+LSC_2 model. Furthermore, the Dice of enhanced was largely recovered.

Replacing $SDCB_2$ with $RsCB_2$ Instead of one block containing convolutions with different dilation rates (Full-$Dilate_2$+LSC_2 model), within a block of Alter-Full-$Dilate_2$+LSC_2 model the convolutions have the same dilation rate. This alteration diminished the performance in terms of all metrics for the core and enhanced substructures. The decrease in Dice, minimum of 2.4% was more pronounced than the reduction of Sens, about 1%, which indicate the augment of FP. The contour definition, represented as HD_{95}, was deteriorated by the modifications in the number of TP and false detections. The segmentation of the region of the brain affected by the tumor slightly improved in all metrics. The use of RsCB and $RsCB_2$ blocks, instead of $SDCB_2$, shown to be detrimental for core and enhanced and marginally beneficial for complete tumor. The difference between these models is in the hierarchical blocks. Merging features with

a higher difference of receptive field seems to be a harder task.

Increment of dilation rate Full-Dilate$_3$+LSC$_2$ model contains dilated convolutions with maximum $d = 3$ in order to obtain a higher receptive field. Regarding the enhanced tumor, the augment of context information caused the decrease of performance in terms of all metrics. Dice reduced by 1.8% while Sens only diminished by 0.4%, so a smaller area of enhanced was detected and the number of false detections enlarged. These modifications led to the deterioration of the boundary of the active tumor region. Analyzing the core substructure, Sens increased by 2%. The augment of TP was achieved at the expense of false detections since, even with this augment, Dice decreased by 1.2%. Furthermore, the FP are located far from the contour of this region. Full-Dilate$_3$+LSC$_2$ model obtained the best value of Dice and HD$_{95}$ for the complete tumor. The reduction of TP was accompanied with the reduction of false detections, which is noticed in the increase of Dice, and consequently improved the complete tumor delineation. Basically, the largest structure, complete, was the only region that beneficed by the use of $d = 3$.

Table 9: Segmentation results concerning the incorporation of dilated convolutions obtained on the Leaderboard set. Values in bold show the best score among all approaches.

Model	Dice			Sens			HD$_{95}$		
	Comp.	Core	Enh.	Comp.	Core	Enh.	Comp.	Core	Enh.
Baseline+LSC$_2$	0.8944	0.7740	**0.7460**	0.8967	0.7803	0.7747	6.51	11.18	4.96
Dilate$_2$+LSC$_2$	0.8933	0.7898	0.7084	0.8897	0.7891	0.7833	6.42	8.94	4.84
Full-Dilate$_2$+LSC$_2$	0.8944	**0.8051**	0.7353	0.9023	0.7891	**0.7859**	6.79	**8.34**	**4.76**
Alter-Full-Dilate$_2$+LSC$_2$	0.8948	0.7810	0.7046	**0.9071**	0.7800	0.7752	6.61	9.27	5.39
Full-Dilate$_3$+LSC$_2$	**0.8958**	0.7926	0.7171	0.8923	**0.8098**	0.7813	**6.28**	9.36	5.74

7.3.4 Reduction of the number of levels

The elimination of a downsampling step together with the substitution of RsCBs with SDCB$_3$'s in the Baseline model gave rise to Red-Full-Dilate$_3$ model. Both models have similar receptive fields, but the reduction of resolution is smaller in the latter. These models are compared in Table 10. Regarding the enhancing region, these modifications resulted in a relevant increase of Sens and Dice, more specifically, by 1.7%. This way, the area of detected enhanced region increased while preserving the number of false detections. The contour definition of this region practically remained unchanged since the HD$_{95}$ augment by 0.20. Red-Full-Dilate$_3$ model detects more tumor core, in other words, the Sens of this substructure increased in 2.4%. Although the number of TP increased, the value of Dice suffered a reduction. So the number of false detections considerably augmented. The unchanging of Hausdorff distance indicates that the introduced FP aren't located any further from the core surface. In relation to the complete tumor, all metrics slightly deteriorate. The increase of resolution had practically neutral effects on the segmentation of core and complete regions. It should be noted that the segmentation of the smallest structure, the enhancing tumor, meaningfully improved.

Table 10: Segmentation results concerning the reduction of levels obtained on the Leaderboard set. Values in bold show the best score among all approaches.

Model	Dice			Sens			HD_{95}		
	Comp.	Core	Enh.	Comp.	Core	Enh.	Comp.	Core	Enh.
Baseline	**0.8916**	**0.7743**	0.7190	**0.8939**	0.7597	0.7503	**6.43**	**9.58**	**4.91**
Red-Full-Dilate$_3$	0.8907	0.7736	**0.7289**	0.8923	**0.7840**	**0.7676**	6.88	**9.58**	5.14

7.3.5 Comparison with the state-of-the-art

In Table 11, the proposed approach is compared with state-of-the-art methods for brain tumor segmentation.

Table 11: Segmentation results of different approches on BRATS 2017 Leaderboard set, first the ensemble melhods and, then, the single model approaches. Values in bold show the best score among all approaches. Underlined values shown the best score among single model approaches.

Methods	Dice			Sens			HD_{95}		
	Comp.	Core	Enh.	Comp.	Core	Enh.	Comp.	Core	Enh.
Isensee et al. [64]	0.8960	0.7970	0.7320	0.8960	0.7810	**0.7900**	6.97	9.48	4.55
Wang et al. [66]	**0.9050**	**0.8378**	**0.7859**	-	-	-	**3.89**	**6.47**	**3.28**
Kamnitsas et al. [65]	0.9010	0.7970	0.7380	0.8950	0.7620	0.7830	4.23	6.56	4.50
Mlynarski et al. [72]	0.9000	0.8080	0.7720	-	-	-	-	-	-
Soltaninejad et al. [62]	0.8600	0.7800	0.6600	0.8300	0.7200	0.5700	7.61	8.70	3.76
Jesson and Arbel [63]	0.8990	0.7510	0.7130	**0.9040**	0.7200	0.7320	4.16	8.65	6.98
Islam and Ren [67]	0.8790	0.7810	0.7010	0.8650	0.7210	0.7300	9.11	11.38	11.92
Zhou et al. [69]	0.9039	0.8279	0.7784	-	-	-	-	-	-
Chen et al. [71]	0.8968	0.7984	0.7205	-	-	-	-	-	-
Pereira et al. [70]	0.8950	0.7980	0.7330	-	-	-	5.92	8.95	5.07
Full-Dilate$_2$+LSC$_2$	0.8944	0.8051	0.7353	0.9023	**0.7891**	0.7859	6.79	8.34	4.76

Comparing the Full-Dilate$_2$+LSC$_2$ model with the other single model approaches, it obtained the best score in Sens of core and enhanced regions and in HD_{95} of core tumor. The proposed approach is outperformed by Zhou et al. [69] in Dice of core and enhanced tumor, reaching the second best score in these metrics. It should be noted that the approach proposed by Zhou et al. [69] consists in a 3D FCN, taking advantage of context information in the three views, with a set of layers specialized in each binary task. They also utilized curriculum learning during training. Regarding the robust Hausdorff distance of the enhancing region, the proposed method achieved the second lowest value. Although Soltaninejad et al. [62] has the best contour delineation for enhancing tumor, the values of Dice and, specially, Sens of this tumor substructure are 7% and 20% smaller when compared with Full-Dilate$_2$+LSC$_2$ model. This work also obtained the second highest Sens of complete tumor, which differs in 0.17% from the method of Jesson and Arbel [63]. They employed a 3D FCN with a multi-scale weighted loss, whose sample weights were changed according to a curriculum. In terms of HD_{95} of complete, the proposed approach stood in third, being surpassed by Jesson and Arbel [63] and Pereira et al. [70]. Lastly, the Dice of the complete tumor is the fifth highest value, but it's similar to the one obtained in the works of Pereira et al. [70] and Chen et al. [71].

The ensemble methods consist on using multiple models to obtain the final segmentations. As different models usually make different errors, the ensemble performs significantly better than its members [5]. Thus, an ensemble method have advantage over a single model approach. Nevertheless, the proposed approach achieved similar results in terms of Dice and Sens when compared with the ensemble methods. Furthermore, when taking into account the work of Isensee et al. [64], the proposed method achieved better performance in the majority of the metrics. It should also be noted that the proposed approach surpassed the method of Wang et al. [66] in terms of Hausdorff distances and Dice of complete, when a single model is used for prediction. Analyzing the values of HD_{95}, Full-Dilate$_2$+LSC$_2$ model and ensemble methods principally differ in terms of delineation of tumor substructures.

7.4 Summary

U-Net has achieved high performance on various medical image segmentation tasks. In this architecture, the shallowest features are aggregated in a single step, which may impair segmentation. In contrast, the DLA architecture contains IDA structures that allow the progressive refinement of shallow features, being responsible for spatial fusion. This architecture also contains HDA structures that execute semantic fusion, improving recognition. Additionally, skip connections and dilated convolutions have shown to be beneficial for semantic segmentation.

In this chapter, deep learning methods were utilized to segment brain tumor substructures. First, the comparison between U-Net and DLA was presented. Then, two long skip connections were inserted into the DLA architecture, separately and together, forming three distinct networks. Lastly, dilated convolutions were incorporated into the network for two purposes, replacing standard convolutions and a downsampling step. Dilated convolutions were utilized in two configurations, in which one consists in a simplified version of the DCB block presented in chapter 4.

The DLA baseline model achieved higher performance in all tumor subregions when compared with the U-Net architecture. When one long skip connection was incorporated into the baseline model, the results improved, showing that low level information is beneficial for brain tumor segmentation. In contrast, the junction of both skip connections into the same model deteriorated brain tumor segmentation. Comparing the models containing one long skip connection, it was observed that the low level information provided by deeper layers, Baseline+LSC$_2$ model, was more advantageous for the segmentation task. The insertion of simplified DCB blocks with d equal to 2, replacing residual blocks in HDA, also allowed obtaining better tumor segmentations. On the one hand, when dilated convolutions were utilized only in the deeper levels, the most benefited region was the tumor core and the complete tumor segmentation was damaged. On the other hand, utilizing dilated convolutions at all levels, Full-Dilate$_2$+LSC$_2$ model, allowed recovering the performance regarding the complete region, without deteriorating the core tumor segmentation. A model with simplified DCB blocks with d equal to 3 was also tested and it was only beneficial for the detection of the complete tumor. Regarding the substitution of a downsampling step for dilated convolutions, it was noted that the segmentation of the enhancing tumor, which is the smallest region, improved when compared with the baseline model. Thus, Full-Dilate$_2$+LSC$_2$ model, which contains one long skip

connection and dilated convolutions with d equal to 2 in the HDA blocks, obtained better performance considering all tumor substructures. The proposed approach achieved Dice scores of 0.8944, 0.8051 and 0.7353 for complete, core and enhanced regions, respectively. Regarding Sens, the model obtained 0.9023, 0.7891 and 0.7859 for each tumor subregion, namely, complete, core and enhanced. In terms of HD_{95}, Full-Dilate$_2$+LSC$_2$ model achieved the following values, 6.79, 8.34 and 4.76, for complete, core and enhanced regions, respectively. The proposed approach for brain tumor segmentation is competitive with state-of-the-art methods, principally when compared with other single model approaches.

Chapter 8

Conclusion

This last chapter starts by presenting the main conclusions of this work. Then, some topics related with the developed work are pointed out as possible future researches.

8.1 Main Conclusions

The objective of this dissertation was to study deep neural network architectures for the segmentation of retinal vessels and brain tumors, which are extremely distinct structures. The analysis of retinal vessels characteristics allows diagnosing, screening and evaluating several opthalmic and cardiovascular diseases. Brain tumor segmentation is essential for patient monitoring and treatment planning. When done manually, these segmentation tasks are time-consuming and prone to inter-rater variability. Several automatic methods were proposed for each task, but they still have many flaws.

In this work, at first, several architectures based on the U-Net [3] were employed for the task of retinal vessel segmentation, in order to compare two downsampling operations and to study the affect of replacing a downsampling step with dilated convolutions. Taking into account that this kind of convolution can produce gridding artifacts, it was incorporated to the networks in a specific form that was based on the works of Wang et al. [74] and Yu et al. [73] and denominated as DCBs. Regarding the study of downsampling operations, strided convolution achieved superior performance when compared with max pooling, the most commonly used operation. When qualitatively evaluated, the usage of strided convolution was advantageous for identifying thin vessels and distinguishing between hemorrhagic lesions and retinal vessels. The incorporation of dilated convolutions, substituting a downsampling operation, was also beneficial for the task in hand in terms of performance. Furthermore, a qualitative analysis demonstrated that more thin vessels where detected when this modification was employed. This indicates that a high reduction of resolution impair the detection of thin structures. Thus, the model containing a strided convolution and a DCB with maximum dilation rate of 3 obtained the greatest performance. This final approach was compared with the state-of-the-art methods, showing competitive results.

Second, a RNN-based layer, ReNet [82], was incorporated into a pre-trained model, retrieving long term dependencies between pixels within the same patch and augmenting the receptive field of the network. This approach was evaluated for retinal vessel segmentation and obtained a higher performance when

compared with the model utilized as baseline. The qualitative analysis revealed the detection of more thin vessels and a negative consequence, the exacerbation of a hemorrhage. The probability maps obtained with the two models were also examined. The ReNet layer mainly altered the classification of pixels with probability close to 0.5. Moreover, some pixels classification was incorrectly modified, which indicates that some regions of the image haven't benefited from the increase of context.

Third, test time data augmentation allows taking advantage of the information encoded during training, when using operations to increase the amount of training data. In this work, it was applied for retinal vessel segmentation. The multiple probability maps obtained with this technique were merged in two ways and both enabled an increment of performance, specially the novel strategy. The qualitative analysis showed that this technique was harmful for thin vessels and hemorrhagic lesion, in spite of the augment of sensitivity and specificity. The most successful form consisted on a trained ConvLSTM with a 3×3 kernel, which determines the pixel final classification using the multiple probabilities of the pixel and the pixel neighborhood. This mode detects more thin vessels than applying averaging. Furthermore, test time data augmentation with ConvLSTM is more valuable for retinal vessel segmentation than the ReNet layer.

Lastly, architectures based on DLA [19] were evaluated for brain tumor segmentation. After comparing U-Net with DLA architectures and verifying that a superior performance was obtained by the last, three studies were executed. Regarding the first study, the incorporation of one long skip connection into DLA was beneficial for the task, specially when using the information provided by deeper layers. However, when two long skip connections were integrated in the same model, its performance dropped substantially. In the next study, the insertion of simplified DCB blocks with dilation rate of 2, replacing all residual blocks allowed to obtain the model with best performance. Additionally, when using a dilation rate of 3, the segmentation of the whole tumor improved but the remaining tumor substructures were deteriorated. This study indicates that the receptive field of the network should be modified according to the size of the structure of interest. The final study consisted on replacing a downsampling step by dilated convolutions. This substitution improved the results regarding the enhanced region, showing that the segmentation of small structures is benefited by higher resolutions. The proposed approach was competitive with the state-of-the-art methods for brain tumor segmentation, principally when they consisted on single models.

8.2 Future Work

New deep neural network architectures are constantly proposed for medical and non-medical image segmentation. These architectures progressively improved several tasks of medical image segmentation, including retinal vessel and brain tumor segmentation. Despite this, there is still room for improvement.

Taking into account the work conceived in this dissertation, various topics can be studied in the future.

First, the methods utilized to segment retinal vessels were only qualitatively evaluated regarding their capability to detect thin vessels. An quantitative analysis is needed to better understand the impact of each modification in the detection of thin vessels. A similar analysis should be made for retinal lesions, which are commonly and wrongly identified as vessels by deep learning models.

Second, features created with Stationary Wavelet Transform were shown to improve the segmentation

of retinal vessels in the work of Oliveira et al. [47]. The deep leaning models proposed in this work only utilized the green channel of the retinal fundus image as input. The inclusion of new features, as wavelets, may improve the performance of the models.

Third, the ReNet layer and the test time data augmentation techniques, already applied for retinal vessel segmentation, should be tested for brain tumor segmentation, using as base the Full-Dilate$_2$+LSC$_2$ model.

Fourth, transfering low level information to deeper layers in DLA showed to improve brain tumor segmentation. However, the model containing two long skip connections, Baseline+LSCs, obtained a poor performance. In this model, an attention mechanism, as the one proposed by Pereira et al. [70], should be employed to select the relevant features at the critical point.

Fifth, the recent approaches for brain tumor segmentation have a simpler pre-processing technique, each modality of each patient is independently normalized. Although no results were presented, some models were implemented with normalized data and showed promising results.

Lastly, He et al. [95] analyzed the residual convolutional block, composed by ReLU, BN and convolution, in terms of its components order. Although no results were presented, this study was started using the DLA architecture and it showed promising results for brain tumor segmentation.

Bibliography

[1] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

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

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

[4] Ana P Lopes, Alexandrine Ribeiro, and Carlos A Silva. Dilated convolutions in retinal blood vessels segmentation. In *2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG)*, pages 1–4. IEEE, 2019.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[6] Kevin Patrick Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

[8] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[10] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285v2*, 2018.

[11] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990.

[12] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, pages 2528–2535. IEEE, 2010.

[13] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

[14] Yi-Tong Zhou and Rama Chellappa. Computation of optical flow using a neural network. In *IEEE International Conference on Neural Networks*, volume 1998, pages 71–78, 1988.

[15] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[17] Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Advances in neural information processing systems*, pages 1945–1953, 2017.

[18] Karishma Pawar and Vahida Z Attar. Assessment of autoencoder architectures for data representation. *Deep Learning: Concepts and Architectures*, 866:101, 2019.

[19] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[22] Alex Graves. *Supervised sequence labelling with recurrent neural networks*. Springer, 2012.

[23] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[25] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[26] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[28] Stephen J Ryan, Andrew P Schachat, Charles P Wilkinson, David Hinton, and Peter Wiedemann. *Retina*. Saunders, 5th edition edition, 2012.

[29] Pearse A Keane and Srinivas R Sadda. Retinal imaging in the twenty-first century: state of the art and future directions. *Ophthalmology*, 121(12):2489–2500, 2014.

[30] Caroline R Baumal and Jay S Duker. *Current Management of Diabetic Retinopathy*. Elsiver, 2018.

[31] Eddie YK Ng, U Rajendra Acharya, Jasjit S Suri, and Aurelio Campilho. *Image analysis and modeling in ophthalmology*. CRC press, 2014.

[32] Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. Blood vessel segmentation methodologies in retinal images–a survey. *Computer methods and programs in biomedicine*, 108(1):407–433, 2012.

[33] Michael D Abràmoff, Mona K Garvin, and Milan Sonka. Retinal imaging and image analysis. *IEEE reviews in biomedical engineering*, 3:169–208, 2010.

[34] TJ MacGillivray, Emanuele Trucco, JR Cameron, Baljean Dhillon, JG Houston, and EJR Van Beek. Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions. *The British journal of radiology*, 87(1040):20130832, 2014.

[35] Jasem Almotiri, Khaled Elleithy, and Abdelrahman Elleithy. Retinal vessels segmentation techniques and algorithms: a survey. *Applied Sciences*, 8(2):155, 2018.

[36] Atul Kumar. *Retina: Medical and Surgical Management*. Jaypee Brothers Medical Pub, 2018.

[37] Shahzad Akbar, Muhammad Sharif, Muhammad Usman Akram, Tanzila Saba, Toqeer Mahmood, and Mahyar Kolivand. Automated techniques for blood vessels segmentation through fundus retinal images: A review. *Microscopy research and technique*, 82(2):153–170, 2019.

[38] Sufian A Badawi and Muhammad Moazam Fraz. Optimizing the trainable b-cosfire filter for retinal blood vessel segmentation. *PeerJ*, 6:e5855, 2018.

[39] Hugo Aguirre-Ramos, Juan Gabriel Avina-Cervantes, Ivan Cruz-Aceves, José Ruiz-Pinales, and Sergio Ledesma. Blood vessel segmentation in retinal fundus images using gabor filters, fractional derivatives, and expectation maximization. *Applied Mathematics and Computation*, 339:568–587, 2018.

[40] Jiong Zhang, Yuan Chen, Erik Bekkers, Meili Wang, Behdad Dashtbozorg, and Bart M ter Haar Romeny. Retinal vessel delineation using a brain-inspired wavelet transform and random forest. *Pattern Recognition*, 69:107–123, 2017.

[41] Xiaohong Wang, Xudong Jiang, and Jianfeng Ren. Blood vessel segmentation from fundus image by a cascade classification framework. *Pattern Recognition*, 88:331–341, 2019.

[42] Paweł Liskowski and Krzysztof Krawiec. Segmenting retinal blood vessels with deep neural networks. *IEEE transactions on medical imaging*, 35(11):2369–2380, 2016.

[43] Avijit Dasgupta and Sonam Singh. A fully convolutional neural network based structured prediction approach towards the retinal vessel segmentation. In *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pages 248–251. IEEE, 2017.

[44] Juan Mo and Lei Zhang. Multi-level deep supervised networks for retinal vessel segmentation. *International journal of computer assisted radiology and surgery*, 12(12):2181–2193, 2017.

[45] Zengqiang Yan, Xin Yang, and Kwang-Ting Cheng. Joint segment-level and pixel-wise losses for deep learning based retinal vessel segmentation. *IEEE Transactions on Biomedical Engineering*, 65(9): 1912–1923, 2018.

[46] Kai Hu, Zhenzhen Zhang, Xiaorui Niu, Yuan Zhang, Chunhong Cao, Fen Xiao, and Xieping Gao. Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function. *Neurocomputing*, 2018.

[47] Américo Filipe Moreira Oliveira, Sérgio Rafael Mano Pereira, and Carlos Alberto Batista Silva. Retinal vessel segmentation based on fully convolutional neural networks. *Expert Systems with Applications*, 2018.

[48] Song Guo, Kai Wang, Hong Kang, Yujun Zhang, Yingqi Gao, and Tao Li. Bts-dsn: Deeply supervised neural network with short connections for retinal vessel segmentation. *International journal of medical informatics*, 126:105–113, 2019.

[49] Seung Yeon Shin, Soochahn Lee, Il Dong Yun, and Kyoung Mu Lee. Deep vessel segmentation by learning graphical connectivity. *Medical image analysis*, page 101556, 2019.

[50] Jacquelyn L Banasik and Lee-Elen C Copstead. *Pathophysiology*. Elsevier, 6th edition edition, 2018.

[51] Stefan Bauer, Roland Wiest, Lutz-P Nolte, and Mauricio Reyes. A survey of mri-based medical image analysis for brain tumor studies. *Physics in Medicine & Biology*, 58(13):R97, 2013.

[52] Philip B Gorelick, Fernando Testai, Graeme Hankey, and Joanna M Wardlaw. *Hankey's Clinical Neurology*. CRC Press, 2nd edition edition, 2014.

[53] David N Louis, Hiroko Ohgaki, Otmar D Wiestler, Webster K Cavenee, Peter C Burger, Anne Jouvet, Bernd W Scheithauer, and Paul Kleihues. The 2007 who classification of tumours of the central nervous system. *Acta neuropathologica*, 114(2):97–109, 2007.

[54] David N Louis, Arie Perry, Guido Reifenberger, Andreas Von Deimling, Dominique Figarella-Branger, Webster K Cavenee, Hiroko Ohgaki, Otmar D Wiestler, Paul Kleihues, and David W Ellison. The 2016 world health organization classification of tumors of the central nervous system: a summary. *Acta neuropathologica*, 131(6):803–820, 2016.

[55] Lisa M DeAngelis. Brain tumors. *New England Journal of Medicine*, 344(2):114–123, 2001.

[56] Benjamin M Ellingson, Martin Bendszus, Jerrold Boxerman, Daniel Barboriak, Bradley J Erickson, Marion Smits, Sarah J Nelson, Elizabeth Gerstner, Brian Alexander, Gregory Goldmacher, et al. Consensus recommendations for a standardized brain tumor imaging protocol in clinical trials. *Neuro-oncology*, 17(9):1188–1198, 2015.

[57] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[58] Martin J van den Bent, Jeffrey S Wefel, D Schiff, Martin JB Taphoorn, K Jaeckle, L Junck, T Armstrong, A Choucair, Adam D Waldman, Thierry Gorlia, et al. Response assessment in neuro-oncology (a report of the rano group): assessment of outcome in trials of diffuse low-grade gliomas. *The lancet oncology*, 12(6):583–593, 2011.

[59] JW Henson, S Ulmer, and GJ Harris. Brain tumor imaging in clinical trials. *American Journal of Neuroradiology*, 29(3):419–424, 2008.

[60] PY Wen, DR Macdonald, DA Reardon, TF Cloughesy, AG Sorensen, E Galanis, J Degroot, W Wick, MR Gilbert, AB Lassman, C Tsien, T Mikkelsen, ET Wong, MC Chamberlain, R Stupp, KR Lamborn, MA Vogelbaum, MJ van den Bent, and SM Chang. Updated response assessment criteria for high-grade gliomas: response assessment in neuro-oncology working group. *Journal of Clinical Oncology*, 28(11):1963–1972, 2010.

[61] Benjamin M Ellingson, Patrick Y Wen, and Timothy F Cloughesy. Modified criteria for radiographic response assessment in glioblastoma clinical trials. *Neurotherapeutics*, 14(2):307–320, 2017.

[62] Mohammadreza Soltaninejad, Lei Zhang, Tryphon Lambrou, Guang Yang, Nigel Allinson, and Xujiong Ye. Mri brain tumor segmentation and patient survival prediction using random forests and fully convolutional networks. In *International MICCAI Brainlesion Workshop*, pages 204–215. Springer, 2017.

[63] Andrew Jesson and Tal Arbel. Brain tumor segmentation using a 3d fcn with multi-scale loss. In *International MICCAI Brainlesion Workshop*, pages 392–402. Springer, 2017.

[64] Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein. Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. In *International MICCAI Brainlesion Workshop*, pages 287–297. Springer, 2017.

[65] Konstantinos Kamnitsas, Wenjia Bai, Enzo Ferrante, Steven McDonagh, Matthew Sinclair, Nick Pawlowski, Martin Rajchl, Matthew Lee, Bernhard Kainz, Daniel Rueckert, et al. Ensembles of multiple models and architectures for robust brain tumour segmentation. In *International MICCAI Brainlesion Workshop*, pages 450–462. Springer, 2017.

[66] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *International MICCAI Brainlesion Workshop*, pages 178–190. Springer, 2017.

[67] Mobarakol Islam and Hongliang Ren. Class balanced pixelnet for neurological image segmentation. In *Proceedings of the 2018 6th International Conference on Bioinformatics and Computational Biology*, pages 83–87. ACM, 2018.

[68] Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta, and Deva Ramanan. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *arXiv preprint arXiv:1702.06506*, 2017.

[69] Chenhong Zhou, Changxing Ding, Zhentai Lu, Xinchao Wang, and Dacheng Tao. One-pass multi-task convolutional neural networks for efficient brain tumor segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 637–645. Springer, 2018.

[70] Sérgio Pereira, Adriano Pinto, Joana Amorim, Alexandrine Ribeiro, Victor Alves, and Carlos A Silva. Adaptive feature recombination and recalibration for semantic segmentation with fully convolutional networks. *IEEE transactions on medical imaging*, 2019.

[71] Shengcong Chen, Changxing Ding, and Minfeng Liu. Dual-force convolutional neural networks for accurate brain tumor segmentation. *Pattern Recognition*, 88:90–100, 2019.

[72] Pawel Mlynarski, Hervé Delingette, Antonio Criminisi, and Nicholas Ayache. 3d convolutional neural networks for tumor segmentation using long-range 2d context. *Computerized Medical Imaging and Graphics*, 73:60–72, 2019.

[73] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[74] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460. IEEE, 2018.

[75] Keras: The python deep learning library. `https://keras.io/`. Acessed: 2018-11-28.

[76] Tensorflow™. `https://www.tensorflow.org/`. Acessed: 2018-11-28.

[77] Joes Staal, Michael D. Abràmoff, Meindert Niemeijer, Max A. Viergever, and Bram van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(1):501–509, 2004.

[78] AD Hoover, Valentina Kouznetsova, and Michael Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical imaging*, 19(3):203–210, 2000.

[79] Frederic Zana and J-C Klein. Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE transactions on image processing*, 10(7):1010–1019, 2001.

[80] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations*, 2017.

[81] Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. Evaluation measures for models assessment over imbalanced data sets. *J Inf Eng Appl*, 3(10), 2013.

[82] Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, and Yoshua Bengio. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393*, 2015.

[83] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 41–48, 2016.

[84] Adriano Pinto, Sergio Pereira, Raphael Meier, Victor Alves, Roland Wiest, Carlos A Silva, and Mauricio Reyes. Enhancing clinical MRI perfusion maps with data-driven maps of complementary nature for lesion outcome prediction. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 107–115, 2018.

[85] Adriano Pinto, Richard McKinley, Victor Alves, Roland Wiest, Carlos Alberto Silva, Mauricio Reyes, et al. Stroke lesion outcome prediction based on MRI imaging combined with clinical information. *Frontiers in Neurology*, 9:1060, 2018.

[86] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In *International MICCAI Brainlesion Workshop*, pages 61–72. Springer, 2018.

[87] Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *International MICCAI Brainlesion Workshop*, pages 311–320. Springer, 2018.

[88] Yingda Xia, Lingxi Xie, Fengze Liu, Zhuotun Zhu, Elliot K Fishman, and Alan L Yuille. Bridging the gap between 2d and 3d organ segmentation with volumetric fusion net. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 445–453. Springer, 2018.

[89] Sripad Krishna Devalla, Prajwal K Renukanand, Bharathwaj K Sreedhar, Giridhar Subramanian, Liang Zhang, Shamira Perera, Jean-Martial Mari, Khai Sing Chin, Tin A Tun, Nicholas G Strouthidis, et al. Drunet: a dilated-residual u-net deep learning network to segment optic nerve head tissues in optical coherence tomography images. *Biomedical optics express*, 9(7):3244–3265, 2018.

[90] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4: 170117, 2017.

[91] Nicholas J Tustison, Brian B Avants, Philip A Cook, Yuanjie Zheng, Alexander Egan, Paul A Yushkevich, and James C Gee. N4itk: improved n3 bias correction. *IEEE transactions on medical imaging*, 29(6):1310, 2010.

[92] László G Nyúl, Jayaram K Udupa, and Xuan Zhang. New variants of a method of mri scale standardization. *IEEE transactions on medical imaging*, 19(2):143–150, 2000.

[93] Sérgio Pereira, Victor Alves, and Carlos A Silva. Adaptive feature recombination and recalibration for semantic segmentation: application to brain tumor segmentation in mri. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 706–714. Springer, 2018.

[94] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3): 297–302, 1945.

[95] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.