Universidade do Minho
Escola de Engenharia

Alexandrine Ribeiro

Study of Attention Mechanisms and Ensemble Methods for Medical Image Semantic Segmentation

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

Trabalho efetuado sob a orientação do(a)
Professor Doutor Carlos Alberto Silva

novembro de 2019

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Acknowledgments

I would first like to express my deepest gratitude to my supervisor Professor Carlos A. Silva. I am very thankful for the dedication, knowledge sharing, and for steering me in the right direction.

I would also like to extend my sincere thanks to Adriano Pinto for providing me with support and patience throughout the duration of this dissertation. For all this, I wish him all the best.

Finally, I must express my very profound gratitude to my family and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of writing this dissertation. This accomplishment would not have been possible without them. Thank you.

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 Mecanismos de Atenção e Métodos de Ensemble para Segmentação Semântica de Imagem Médica

Atualmente, o desenvolvimento dos cuidados médicos aliado ao aperfeiçoamento das técnicas de imagiologia médica garantem uma melhor capacidade de diagnóstico e uma melhor identificação dos problemas de saúde de difícil tratamento. Nos últimos anos, a segmentação automática de imagens médicas provou ser um método viável e robusto para superar a necessidade de recursos humanos e as variabilidades inter- e intra-especialista associadas à segmentação manual. Os métodos de aprendizagem profunda estão fornecendo soluções interessantes com boa precisão no contexto da segmentação de imagens médicas, e são vistos como fundamentais para futuras aplicações no setor da saúde.

A presente dissertação tem como finalidade o desenvolvimento de métodos de Deep Learning para dois problemas de segmentação automática de imagem médica, nomeadamente, a segmentação de vasos sanguíneos da retina e segmentação de tumores cerebrais. O cancro e as doenças oculares afetam uma grande parte da população. A maioria das pessoas tem problemas de visão em alguma fase da vida, e os tumores cerebrais destacam-se dos demais tipos de cancro por terem elevada taxa de mortalidade.

Para a segmentaçao de vasos da retina o método proposto é baseado na média estocástica dos pesos. Os métodos de *ensemble* têm demonstrado melhorar o desempenho em várias aplicações mas, simultaneamente, aumentam a complexidade do treino e o tempo necessário para a previsão. Assim, exploramos duas técnicas alternativas de *ensemble*, que mitigam estas limitações explorando o espaço de pesos da rede para compor o *ensemble*. O método proposto foi avaliado em três bases de dados públicas e amplamente reconhecidas na área da segmentação de vasos da retina: DRIVE, STARE e CHASE_DB1. Os resultados mostram que o método explora com sucesso o espaço de peso da rede, encontrando uma solução com melhor generalização. Em resumo, o método proposto obteve resultados competitivos, concluindo que este é capaz de melhorar o desempenho na segmentação dos vasos retinianos.

Adicionalmente, foi proposto um mecanismo de atenção com contexto de escala adaptativa para a segmentação de tumores cerebrais O método proposto é baseado numa abordagem de segmentação hierárquica utilizando redes totalmente convolucionais e blocos de atenção que capturam informaçao multi-escala através do processamento de ramos paralelos com diferentes escalas. O bloco de atençao proposto é modular, podendo facilmente ser incorporado noutras arquiteturas existentes para aumentar o seu poder de representação. Para avaliaçao da metodologia proposta recorreu-se da base de dados pública *Multimodal Brain Tumor Segmentation Challenge 2017: O MICCAI BraTS 2017* que permitiu a validação dos resultados realizada pela sua plataforma *online*. Os resultados mostram os benefícios desse mecanismo de atenção, sendo as melhores configurações do bloco de atenção competitivas com o estado-da-arte na segmentação de tumores cerebrais.

Palavras-chave: Mecanismos de Atenção, Métodos de *Ensemble*, Segmentação de Tumores Cerebrais, Segmentação dos Vasos da Retina.

Abstract

Study of Attention Mechanisms and Ensemble Methods for Medical Image Semantic Segmentation

Nowadays, the development of medical care and the improvements in medical imaging techniques ensure a better diagnosis capability and a better identification of health problems of difficult treatment. Time is a critical factor for medical diagnosis, and early detection and evaluation can potentially add years to a patient's life. Over the past years, automatic medical image segmentation has proven to be a viable and robust method to overcome the large costs of human resources and the intra- and inter-rater variabilities associated with manual segmentation. Deep Learning methods are providing exciting solutions with good accuracy in the context of medical image segmentation, and they are seen as key for future applications in the health sector.

This dissertation focuses on Deep Learning-based methods for automatic segmentation in two medical imaging tasks, namely, retinal blood vessel segmentation and brain tumor segmentation. Cancer and ocular diseases affect a large part of the population. Most people have eye problems at one time or another in life, and brain tumors stand out by having one of the highest mortality rates among cancers.

In particular, we employed alternative ensemble techniques, Stochastic Weight Averaging and Fast-Stochastic Weight Averaging for retinal blood vessel segmentation. Ensemble methods have shown to improve the performance in several applications, but at the same time increasing the complexity of the training and the time required for the prediction. So, we investigate these alternative ensemble techniques, which mitigate those limitations by exploring the weight space of the network to compose the ensemble. The proposed method was evaluated on three publicly available databases, widely used in the retinal vessel segmentation area: DRIVE, STARE, and CHASE_DB1. The results showed that stochastic weight averaging successfully explores the network's weight space, finding a solution with better generalization. To sum up, the proposed method achieved competitive results, concluding that it is capable of improving the generalization performance in retinal vessel segmentation.

Additionally, we proposed an attention mechanism with adaptive-scale context for brain tumor segmentation. We employed an encoder-decoder fully convolutional network based hierarchical approach coupled with an attention block that captures multi-scale information by processing parallel branches with different scales. Our attention block is modular, being able to be easily incorporated into other existing network architectures to increase their representational power. The method was tested with the publicly available database from Multimodal Brain Tumor Segmentation Challenge 2017: MICCAI BraTS 2017 allowing also an evaluation performed by its online platform. The results showed the benefits of this attention mechanism, being the best configurations of the attention block competitive with the state-of-the-art for brain tumor segmentation.

Keywords: Attention Mechanisms, Brain Tumor Segmentation, Ensemble Methods, Retinal Vessel Segmentation.

Table of Contents

List of Figures

List of Tables

Acronyms

AD Alzheimer’s Disease. 7

Adam Adaptive Moment Estimation. 35, 36

AMD Age-related Macular Degeneration. 6

ANN Artificial Neural Network. 21, 24

BN Batch Normalization. 34

CNN Convolutional Neural Network. 17, 18, 24, 26, 30, 31, 72

FCN Fully Convolutional Network. viii, 31, 44, 45, 54, 55, 58–61, 63, 67–72

FGE Fast Geometric Ensembling. 15, 16

FLAIR T2-weighted Fluid-Attenuated Inversion Recovery. 10, 12, 63

FOV Field of View. 8

GAP Global Average Pooling. 58

GRU Gated Recurrent Units. 17

HGG High Grade Gliomas. 9, 60, 63

LGG Low Grade Gliomas. 9, 10, 60, 63

LR Learning rate. 34, 36, 37, 63

LSTM Long Short-term Memory. 17

MLP Multilayer Perceptron. 22

MRI Magnetic Resonance Imaging. 2, 3, 9, 10, 59, 62–64, 71, 72

NMT Neural Machine Translation. 16, 17, 19

RMSProp Root Mean Square Propagation. 35

RNN Recurrent Neural Network. 17, 39

ROI Region of Interest. 61, 72

SGD Stochastic Gradient Descent. 36

SWA Stochastic Weight Averaging. vi, 16, 38, 42–44, 47–51, 54, 70, 71

T1 T1-weighted. 10, 12, 63

T1c Post-contrast T1-weighted. 10, 12, 63

T2 T2-weighted. 10, 12, 63

WHO World Health Organization. 8, 9

Chapter 1

Introduction

This chapter presents the motivation and main objectives associated with the work developed in this dissertation. In addition, the contributions resulting from this work are also pointed out and the structure of the following chapters is clarified.

1.1 Context and Motivation

The development and proliferation of medical imaging technologies is revolutionizing medicine. The revolutionary capabilities of new medical imaging modalities and computing power have opened new windows for medical research and clinical diagnosis. Medical imaging allows physicians to peer non-invasively into the human body and glean potentially life-saving qualitative and quantitative information. Indeed, medical imaging provides increased information about diseases or anatomic structures that is not available with conventional diagnosis methods, and paves the way to develop our understanding of physiological processes. Thus, it is a valuable tool that is complementary to but not compete with the traditional methods [1, 2]. The role of medical imaging has expanded beyond the simple visualization and inspection of anatomic structures. It has become a tool for widespread applications, for example, intra-operative navigation, surgical planning and simulation, radiotherapy planning, tissue classification, and for tracking the progress of diseases [1].

The segmentation of anatomical structures plays a crucial role in extraction of information to many diagnostic and surgical procedures. However, the complex nature of medical images makes it a difficult task. In fact, medical segmentation structures are often non-rigid, vary in location, shape and size, have indistinct and disconnected boundaries, and differ from patient to patient [3]. This makes manual segmentation a dull, time-consuming task and prone to both inter-rater and intra-rater variability [4]. Hence, it is not reproducible. For these reasons, automatic and semi-automatic segmentation algorithms have been extensively explored for many years. However, the problem remains challenging, with no general and unique solution [1].

The retina is a membrane that covers the interior of the eye and is responsible for converting light into electrical signals that are processed by the visual cortex of the brain [5]. It is the only part of the human body in that the bloodstream can be observed directly and non-invasively. Thus, retinal assessment is a straight way to identify not only ocular but also systemic and neurodegenerative diseases. Current systems for fundus image analysis require to segment the retinal vascular tree as an intermediate step. Therefore, retinal vessel segmentation methodologies are seen as an increasingly important tool [6].

Several types of cancer affect millions of people worldwide, every year. Among these, brain tumors have high mortality rates, especially gliomas, which are the most common ones. In clinical practice, MRI has become the standard imaging technology for assessing these tumors [7]. MRI sequences provides volumetric data with different contrasts. Thus, it requires more time for its analysis by an experienced radiologist [8]. Therefore, proper, reliable and fully automatic brain tumor segmentation methods are imperative.

1.2 Objectives of the dissertation

The potential of automatic medical image analysis is undoubtedly vast. This type of system could give clinical units the ability to analyze large sets of images, with significant reductions in time and financial costs. Then, robust automatic segmentation approaches may open the door to the creation of new screening systems that can completely redesign the diagnosis and treatment of several diseases.

In this work, we seek to explore Deep Learning-based methodologies, and develop reliable automatic solutions to the relevant tasks of retinal vessel segmentation and brain tumor segmentation.

The development of this procedures combines several areas of knowledge, such as, image processing and machine learning, emphasizing ensemble methods and attention mechanisms.

1.3 Contributions from this dissertation

Throughout this dissertation were developed some methods that, to the best of our knowledge, can be considered original contributions, namely:

- Exploration of an ensemble strategy based on weight averaging to improve the performance of retinal vessel segmentation algorithms. We used a fully convolutional neuronal network based segmentation system, where fusion of information at different scales is useful for dealing with the multi-scale nature of the retinal vasculature.

- A new attention mechanism that adaptively selects the context by processing several parallel branches with different scales. More particularly, the combination of a hierarchical FCN-based brain tumor segmentation approach, to tackle the data imbalance problem with, the proposed attention block to enhance the more discriminative features.

1.4 Structure of the dissertation and General Overview

The remaining chapters of this dissertation are organized as follows. In Chapter 2 we start by providing a medical perspective on the human visual process, then we evolve to the description of the main diseases that manifest through the retinal vascular tree, and its appearance in fundus images. In this chapter, we also make a clinical contextualization on gliomas and their appearance on MRI images. Then, in Section 2.3, we delve into the problems of automatic segmentation of retinal vessels and brain tumors. It is described why it is needed to automate these tasks, and the main challenges regarding each task. In Chapter 3, we review some of the main state-of-the-art works in ensemble learning and attention mechanisms, focusing on image processing methods. Chapter 4 introduces the theoretical concepts associated with Machine Learning methods, with special emphasis on convolutional neural networks, ensemble methods, and attention mechanisms. Chapter 5 and Chaper 6 describe the phases of the proposed automatic segmentation methodologies, also the main results are reported and compared with the state-of-the-art methods, for retinal vessel segmentation and brain tumor segmentation, respectively. Finally, Chapter 7 highlights the main conclusions of the work developed and suggests possible future research directions.

Chapter 2

Medical Perspective

This chapter provides a medical perspective on two different medical problems. We start by describing, concisely, the human visual system, with focus on the retina as a potential biomarker of numerous diseases. Then, we present brain tumors, emphasizing gliomas. Among brain tumors these are the most common and aggressive neoplasms, gathering substantial research attention.

Image analysis of both, retina and gliomas, are the main purpose of the approaches developed in this dissertation, specifically retinal vessel segmentation and glioma segmentation. So, we also discuss the structure and characteristics of the retinal vascular tree and gliomas and relate it with their appearance in medical imaging techniques.

2.1 The Human Vision

Out of all the five human senses, the vision seems to be the most important for us. Most of the signals we collect to perceive the environment that surrounds us and react to their stimuli are processed through our eyes. Humans are unique in their reliance on vision as the predominant sense and this is reflected in how complex our eyes are relative to animals. In fact, the eyes have the highest level of development of the body's sensory organs and a larger part of the brain is allocated to sight than to hear, touch, smell or taste combined [5].

Figure 1 represents a cross-section of the human eye, where it is possible to identify the different structures. When a ray of light hits the eye, it travels successively trough the cornea, aquous humour, the pupil, lens and vitreous humour until it converges on a specif spot at the back of the eye, the retina. The cornea, at the front of the eye, and the lens, located right behind the pupil, work in tandem to focus the ray onto the retina [5]. The retina is a layered nervous tissue membrane that covers the back of the eye and it is where the raw sight data from the light collected from the eyes begins to be translated into useful visual information. The retina contains two types of photoreceptors: rods and cones [5, 10]. The light focused on the retina triggers these photoreceptors and they operate as transducers that convert light stimuli into nerve impulses or electrical signals. Rods are more sensitive to light but cannot distinguish between colors, which cones are designed to. Rods are responsible for the scotopic vision. They are easily activated by few photons, being the ones active in dark environments. Cones are activated in bright environments,

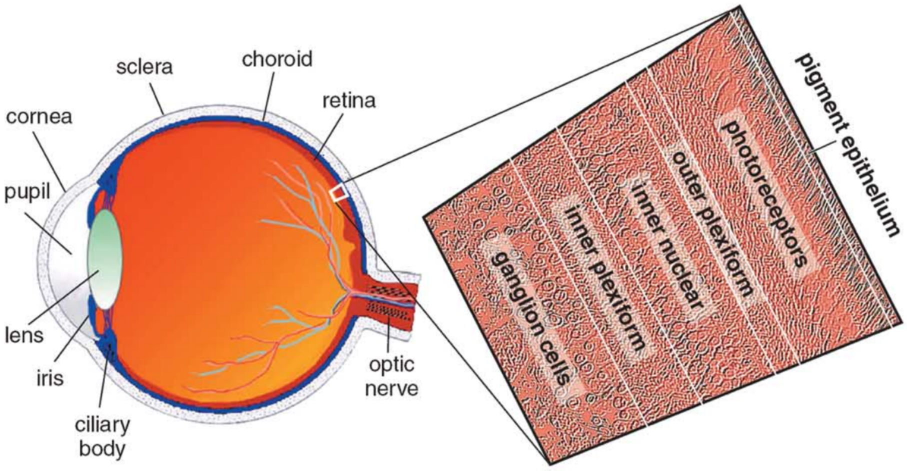


Figure 1: Cross-sectional view of the human eye anatomy and retinal layers. Adapted from [9].

triggering photopic vision and allowing color perception. There are about 120 million rods and up to 7 million cones, in one eye. Rods are distributed around the periphery of the retina and contribute to peripheral vision. In turn, most cones are concentrated in the macula, where the central vision forms. In the center of this region a depression is found, the fovea, where the light rays responsible for maximum visual acuity converge [5].

The optic nerve is responsible for transmitting all visual information from the retina to the brain. It extends the optic disc, which is the eye's blind spot as a result of the absence of photoreceptors. The optic disc is also the entry point for the major blood vessels that supply the retina. The retina is irrigated by two distinct blood systems because its metabolic demands are not uniform. One blood network meets about one third of the necessities, feeding the inner part, and it is visible in fundus images. The other supplies the remaining portion, irrigating the outside and it is commonly undetectable [5, 10].

2.1.1 The Retina as a Biomarker

Several diseases can affect the functional and structural balance of the eye. It is straightforward to realize that ocular pathologies may cause structural changes and injury in the eye. But, multiple systemic and neurodegenerative diseases can induce changes in ocular structures, as well, resulting in a broad spectrum of clinical signs. These, coupled with imaging developments, make it possible to interpret these signs as biomarkers, which might be useful for purposes of diagnosing, patient screening and clinical study [11, 12].

Retinal assessment is a straight way to identify ocular problems. When analyzing retinal images, special attention should be given to the optic nerve state, the condition of the macula, and the appearance of the vascular structures, specifically, vessel geometry, bifurcation angle, tortuosity (the frequency of more or less steep curves in the vessels) and, venular and arteriolar diameters (Fig.2) [11].

Next, we describe some of the most relevant diseases that manifest through the retina and identify the main clinical signs associated with each of them.

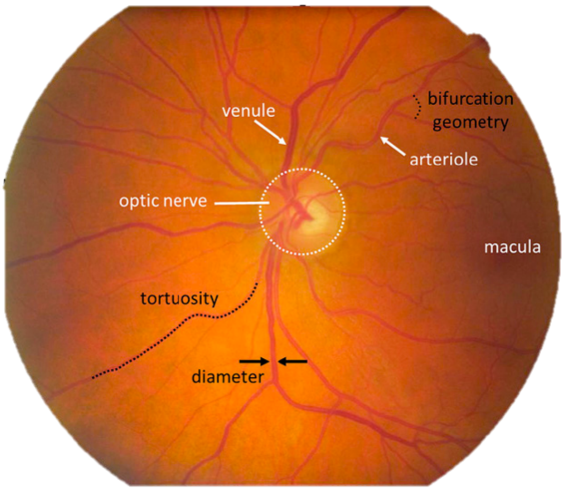


Figure 2: Features of interest in a retinal fundus image. Adapted from [11].

2.1.1.1 Ocular Pathologies

Age-related Macular Degeneration (AMD) and glaucoma are two ocular pathologies responsible for countless cases of blindness [12].

AMD is a disease of the elderly involving the macula. The main cause of AMD is the accumulation of protein and fat (drusen) in the Brunch membrane, a thin cell layer under the retina. In the early stages of the disease, the presence of drusen is the most visible feature. At a later stage, pathological vessels appear in the macula region. These neovascularizations cause a sudden decrease in visual acuity, and distortions due to fluid and blood leakage [12, 13].

Glaucoma damages the optic nerve, mainly affecting the peripheral vision. It generally happens when fluid builds up in the eye increasing the pressure, injuring the optic nerve. There are two major types of glaucoma: primary open-angle and angle-closure glaucoma. The first, happens gradually, the eye does not drain fluid as well as it should. The later occurs when the drain space between the iris and cornea becomes too narrow, and it can end up blocking the area of the eye where the aqueous humor drains from the front of the eye [12, 14].

2.1.1.2 Systemic Diseases

Systemic diseases involve many organs or the whole body. Many of these also affect the eyes. Here, we will only address diabetes and hypertension, as these are some of the most relevant systemic pathologies that manifest through the retina.

Diabetic people can have an eye condition called diabetic retinopathy. At the capillary level, there is proliferation and aggregation of endothelial cells with membrane thickening, size reduction and changes in elasticity, consistency and permeability, causing a decrease in blood supply and generating hypoxia. Multiple vascular changes are visible. In the early stage, microaneurysms, hemorrhages, exudates and cottony spots are perceptible. A state of ischemia is reached, with the accentuation of the capillary block. As the disease progresses, it enters in a proliferative stage, and, in an attempt by metabolism to restore blood perfusion, new vessels appear around the occluded areas. These neovasus are fragile and may cause bleeding in the vitreous cavity, leading to a marked and sudden decrease in visual acuity. If left unchecked, the vessels may develop into fibrovascular scars and subsequent retinal detachment. At the same time, as vascular leakage increases, macula thickening may occur, which is the main indicator of macular edema, that - although not visible in fundus images - is the leading cause of blindness in patients with diabetic retinopathy. Like diabetes, diabetic retinopathy has no cure [12, 15].

Hypertension, also known as high blood pressure, is a chronic cardiovascular disease in which the blood pressure is persistently elevated. The most noticeable effect of hypertension in the eye is hypertensive retinopathy, a condition that encompasses several vascular changes in the retina. At an early stage, elevated blood pressure leads to retinal arterioles constriction and subsequent narrowing. If the blood pressure remains chronically high, structural changes in the arterioles drive to compression of the venules with the appearance of arteriovenous notches. At a more severe stage, there is progression to a state where bleeding and cotton spots are observed and, ultimately, to a malignant state, with papilledema and macular edema, which can lead to blindness. In this case too, the lesions caused by the disease are irreversible and no cure is known [16, 17]

2.1.1.3 Neurodegenerative Diseases

The retina is an attractive source of biomarkers for neurodegenerative pathologies since it shares many features with the brain. Some even describe the retina as a window to the brain, given that it is more accessible for imaging [11].

Alzheimer's disease (AD) is the most common cause of dementia. It causes problems with memory, thinking and behavior. Patients with AD, commonly exhibit reduced vessel diameters (venular and arteriolar), reduced optimality of the branching geometry, reduced complexity of the branching pattern and less tortuous venules. Additionally, there is a reduction in the number of optic nerve head axons and a decrease in the thickness of the peripapillary and macular retinal nerve fiber layer. One of the earliest symptoms of AD could be the thinning of the retinal ganglion cells layer and visual spatial impairment. Studies also suggest an increased incidence of glaucoma pathology in AD patients [11, 18].

2.1.2 Retinal Imaging

The retina is one of the few places in the human body allowing easy non-invasive observation of blood vessels and nerves. Retinal imaging is an integral part of medical examination in ophthalmology. Information in retinal images is useful for screening, diagnosis, monitoring, and treatment of diseases that affect the eye. The main retinal imaging modalities are fundus photography, optical coherence tomography (OCT) and scanning laser ophthalmoscopy (SLO) [6, 11]. Following the line of thought on which this dissertation was developed, it will only be addressed fundus photography, in particular, its functioning and characteristics.

2.1.2.1 Fundus Photography

Fundus photography provides a 2D image of the rear of an eye, the fundus, which includes the retina, the retinal vasculature, the macula, and the optic disc. The retina is photographed, with specialized cameras consisting of an intricate microscope attached to a flash-enabled camera, through the pupil of the patient [19].

The patient is placed with the chin at rest and the forehead in contact with a bar while the operator focuses and aligns the camera before firing the flash and creating the image. This image is an enlarged version of the retinal background with typical viewing angles from $30°$ to $50°$. A larger field of view (FOV) can be achieved by composing multiple images acquired at different fixation points [11, 19]. In addition, higher quality images can be obtained by dilating the pupil with mydriatic drops beforehand [11].

There are various modes of examination in fundus photography. Typically, the retina is examined in full color through white light illumination. However, a filter can be utilized to better observe superficial lesions and some vascular abnormalities. Red wavelengths of light are blocked out creating an enhanced contrast red–free image of retinal blood vessels and other structures. Alternatively, in fluorescein angiography (FA) the patient receives an intravenous injection of a fluorescent dye. This dye fluoresces a different color when light from a specific wavelength reaches it. This allows a high contrast image to be generated, by which a sequence of photographs can be produced that show the movement, and pooling of blood over time, and enabling to identify areas of damage where the dye escapes into the surrounding tissue [6, 11, 19].

2.2 Cancer

Cancer, also known as malignant tumors or neoplasms, is the uncontrolled growth and spread of cells that can affect any part of the body. The growths often invade surrounding tissue and can metastasize to distant sites. Metastases are a major cause of death from cancer. Cancer incidence and mortality are rapidly growing worldwide. According to the World Health Organization (WHO), cancer is the second leading cause of death globally, and is responsible for an estimated 9.6 million deaths in 2018. Globally, cancer is responsible for about 1 in 6 deaths [20].

Within all the cancer types, brain and central nervous system tumors accounted for $296,851$ new cases and $241,037$ deaths in 2018, accounting for about 1.6% of new cancer cases and 2.5% of cancer deaths [21]. As stated by the WHO, brain tumors can be classified into four grades, depending on their aggressiveness. Grades I and II are considered low grade, being the less proliferative and less aggressive tumors. These can progress into grade III or IV, which are malignant tumors and, grow faster and are more proliferative than the low grade ones [7].

2.2.1 Brain Tumors - Gliomas

Brain tumors can have different origins. Gliomas are brain tumors originated in a family of specific brain support cells, the glial cells, such as astrocytes, Schwann cells or oligodendrocytes. The glial cells are very abundant in all the nervous system, peripheral and central. The common property of all the different glial cells is to support and protect nerve cells, both through protection against pathogenic entities and through the ability to supply nutrients and oxygen [22, 23].

As all brain tumors, gliomas can be graded according to their malignancy, following the WHO classification [24]. The determination of the brain tumor grade is imperative for treatment planning and for prognosis. Moreover, the grades can be subdivided into two larger classes: low grade gliomas (LGG), if they are grade II, and high grade gliomas (HGG) for grades III and IV.

- Grade II – Mainly oligodendrogliomas and astrocytomas, they tend to exhibit benign tendencies. The cells are highly dense, but almost normal and growing slow;
- Grade III – Mainly anaplastic gliomas. Malignant tumores without necrotic tissues;
- Grade IV – Glioblastomas. The most common and aggressive malignant gliomas. Usually include regions of hemorrhages, necrosis and fleshy tumor.

Gliomas are composed of an edema and an active zone and, sometimes, a necrotic region. The necrosis is more common in HGG. As for the active zone it can be divided into two distinct tissue classes: the enhancing and the non-enhancing tumor zone. The enhancing regions are a feature of HGG, although it may appear in some LGG. Physiologically, the non-enhancing tumor represents the progression of the active zone, invading the edema, while the enhancing zone corresponds to the nucleus of the active zone. The enhancing regions of a tumor result from a disruption of the blood–brain barrier that leads to blood accumulation. So, these regions are visible after the gadolinium contrast injection. Regardless of the tumor grade, the definition of the active zone and edema are fundamental for diagnosis and subsequent treatment [25]. Additionally, gliomas are a type of tumors that present multi-centrality and multifocality. Thus, their imaging behavior is very varied in terms of size, shape, location and appearance. Multifocality consists of the diffusion of a lesion to other brain areas. Multi-centrality, on the other hand, occurs when several tumor lesions are detected at different brain locations [26].

The increased incidence oh this type of tumor, caused gliomas to be an active area of medical research. The advances in the medical field, which enabled the accessibility to imaging techniques with greater detection capacity of such structures may justify this raise. Also, the increase in average life expectancy due to the treatment and eradication of certain diseases facilitated the progress of new diseases [25].

2.2.2 Imaging in Gliomas

According to the WHO, the diagnosis and prognosis of brain tumors are more robust when a biopsy is performed to the area with signs of tumor [27]. This phase is preceded by a preliminary detection phase, involving imaging techniques, such as, Magnetic Resonance Imaging (MRI), Computed Tomography (CT) and Positron Emission Tomography (PET). MRI has become the standard imaging technique for the assessment of brain tumors [28]. Therefore, in the context of this dissertation, we will only approach the process of detection and diagnosis for MRI.

2.2.2.1 Magnetic Resonance Imaging

MRI is a minimally invasive medical imaging method that ensures high spatial resolution and very good contrast in soft tissues. It operates with magnetic fields that have, so far, no associated degree of harmfulness. Sometimes, MRI is aided by the application of a contrast agent, hence, it is a minimally invasive technique [29].

MRI allows us to evaluate the pathology and anatomy of the brain. Additionally, MRI images are three-dimensional. So, in a prior examination of the tumor, it allows one to determine its location for biopsy or surgery planning, observe the tumor compartments and evaluate the mass effect on the normal brain tissues [30]. Moreover, it is possible to acquire various sequences that result in images with distinct tissue contrasts. In fact, since different structures and tumor tissues are prominent for different MRI sequences, this characteristic is essential for glioma analysis.

The consensus for standard brain tumor imaging includes the following MRI sequences: T1-weighted (T1), 3D isotropic 1 mm post-contrast T1-weighted (T1c), T2-weighted (T2), and T2-weighted Fluid–Attenuated Inversion Recovery (FLAIR) [31]. T1 has a very good contrast for normal brain tissues, such as white matter and gray matter. Gliomas might appear as abnormal zones. Compared to the other sequences, most of the time, T1 does not bring relevant information for brain tumor examination. Often, in glioblastomas, necrosis appears as a hypointense region inside the enhancing tumor in the T1c sequence. Enhancing regions observation is facilitated in the subtraction of T1c and T1 (T1c – T1) so, these MRI sequences shall be acquired with the same parameters. T1 and T1c sequences are not optimal for edema observation and, may not be efficient for assessing non-enhancing tumor, commonly LGG. The whole tumor area, including edema, is hyperintense in T2 and FLAIR. Thus, both T2 and FLAIR sequences are important to distinguish tumor from healthy regions. Normally, T2 is the elected sequence for assessing the non-enhancing active tumor and edema. But, FLAIR is advantageous to evaluate tumors placed near the sulci and the ventricles because the intensities of the cavities filled with cerebrospinal fluid are suppressed [31–33].

2.3 Medical Image Segmentation

Medical imaging techniques provide the ability to non-invasively assess body structures. Futhermore, images are a valuable source of information enabling an improved diagnosis, patient screening, treatment and surgery planning. For these purposes, segmentation is a mandatory step. Image segmentation is one of the most interesting and challenging problems in medical imaging applications specifically.

2.3.1 Retinal Vessel Segmentation

Currently, systems for fundus image analysis require to segment the retinal vascular tree as an intermediate step [6], as blood vessels can provide several useful features for numerous applications. Specifically, retinal vessel segmentation can be used to characterize pathological alterations associated with ophthalmic and systemic diseases [34], as landmarks for multimodal image registration [35], and for localizing other anatomical elements of the retina [36].

Manual segmentation of the retinal blood vessels is a difficult, time–consuming and prone to error task even for the most experienced specialists. Additionally, it is subject to inter– and intra–rater variability, as well. In Fig. 3 we can verify the variability between two human raters, namely in the segmentation of thin vessels. Hence, it is not reproducible. For this reason, faster and automatic segmentation strategies have gained importance in therapeutic and/or surgical planning. However, in clinical practice, the inclusion of automatic systems is still unusual. Although, several studies have demonstrated potential to reduce the variability associated with manual segmentation with robust results [6, 37, 38].

Automatic retinal vessel segmentation remains a challenging task due to difficulties in image acquisition coupled with variations in vessel characteristics. During image acquisition, fundus images often inherited artifacts because of low-quality acquisition devices. In addition, images often contain inadequate contrast between vessels and background and uneven background illumination [37]. Regarding vessel properties, the limitations include the merging of close vessels, difficulties segmenting vessels at regions of bifurcations and crossovers and in the presence of vessel central light reflex. Additionally, the distinction between vessels and other ocular structures can be problematic, resulting in false vessel detection at the optic disc and pathological regions. Also,the segmentation of small thin vessels in the retina is extremely challenging [39].

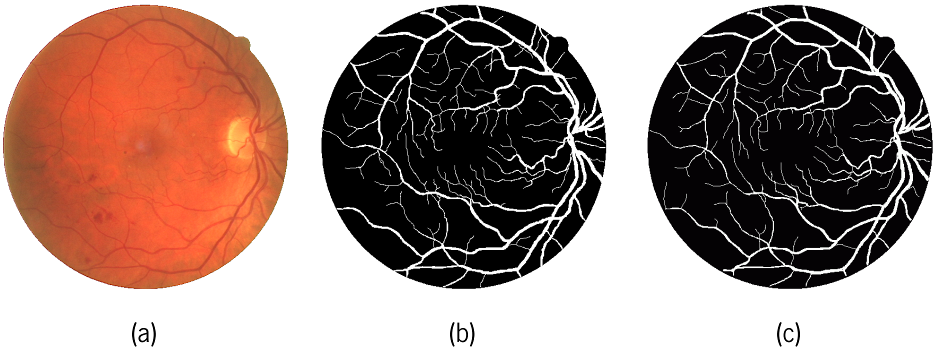


Figure 3: Example of retinal vessel segmentation. (a) Fundus RGB image; (b) 1^{st} human observer segmentation; (c) 2^{nd} human observer segmentation.

2.3.2 Gliomas Segmentation

In manual tumor segmentation, it is required an experienced radiologist to manually delimit the regions of the tumor. The difficulty of the task lies in the peculiarities in terms of tissue imaging behavior that makes the discrimination of the tissue boundaries meticulous and prone to errors. Also, it is time demanding and has intra–rater and inter–rater variability [8, 40]. For this reason, in clinical practice, semi-automatic methods are being utilized. Even though these procedures hasten the image annotation, they still require an operator so the variability problem remains [28]. Therefore, proper and reliable, but also, fully automatic segmentation methods are imperative.

Fully automatic segmentation of brain tumors is a complex task. Gliomas vary in size, shape, appearance and location. Thus, to identify the distinct regions of these neoplasms, different MRI sequences must be taken into account [8].

In addition to the tumor features, MRI entails some obstacles. First, MRI is not calibrated so image intensities do not have a fixed meaning. In fact, intensities may vary if two images from the same patient are acquired in distinct acquisition sites or with different equipments or only at different moments [41, 42]. Second, acquisition artifacts, due to, for example, the bias field or sampling, cause an intensity inhomogeneity in the images. Thus, identical tissues may have different intensity in different locations [43]. At last, images may have different contrasts, as the acquisition parameters and protocols vary over clinics [31].

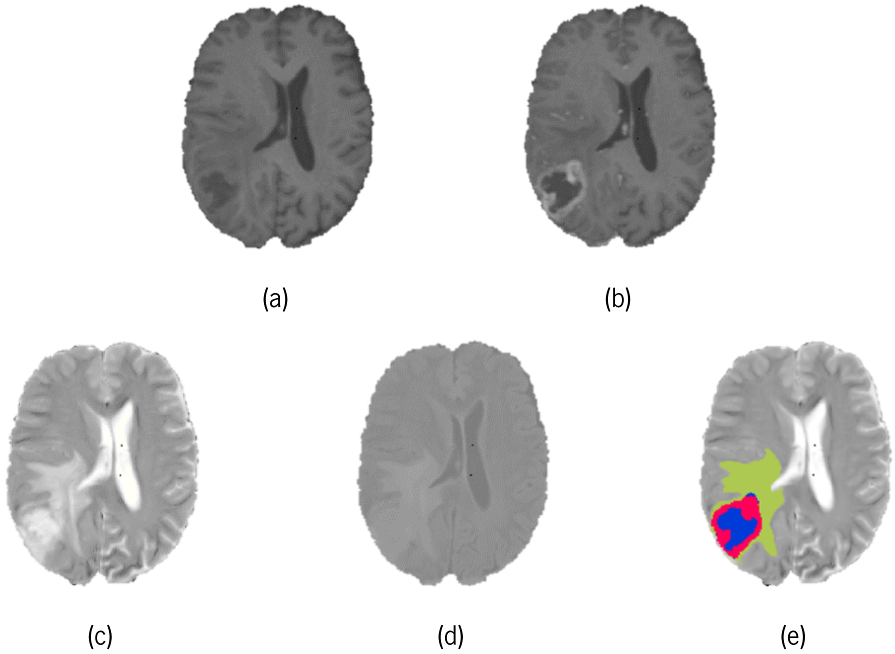


Figure 4: MRI acquisition of a patient with glioma. Sequences: (a) T1; (b) T1c; (c) T2; (d) FLAIR. (e) Manual segmentation. Colors identify different tumor regions: blue – necrosis and non-enhancing tumor, red – enhancing tumor, and green – edema.

Experts [32] defined a set of tumor labels that should be identified, when present, in glioma segmentation: the label 1 corresponds to necrotic tissue, label 2 refers to edema, non-enhancing and enhancing tumor correspond, respectively, to labels 3 and 4. Fig. 4 shows an example of a brain with glioma. The mass effect is clear in T1 sequence (Fig. 4a), through the deformation of the ventricles. T1c (Fig. 31a) evidences the enhancing tumor with necrotic tissue inside it. All the tumoral tissues are hyperintense in both T2 (Fig. 31b) and FLAIR (Fig.4d) sequences. The non-enhancing active tumor demonstrates the need for multiple sequence acquisitions, as this region is difficult to identify. It can be perceived as a low intensity in-between the edema and enhancing tumor in the T2 sequence (Fig. 31b).

2.4 Summary

Cancer and ocular diseases affect a large part of the population. In fact, most people have eye problems at one time or another in life. Moreover, cancer is a disease which impact on society has increased substantially in the past years.

Regarding the human eye, the retina plays a vital role in vision. It converts the information gathered by the photoreceptor cells into electric signals that the brain can process. Nowadays, ophthalmology has several techniques that allow the acquisition of retinal images. The retinal blood vessels are key to the evaluation of various pathological manifestations that can be caused by eye diseases but may also represent side effects from systemic or neurodegenerative diseases. Fundus images are widely recognized as one of the most valuable diagnostic and monitoring tools, as they allow the observation of the retina. Automatic segmentation of the retinal vascular tree is an intermediate step in the analysis of retinal images.

In the specific case of nervous system cancers, it has been found that their considerable worldwide incidence is largely explained by the growing expression of brain tumors called gliomas. Magnetic Resonance Imaging stands as the standard imaging technique for the assessment of these neoplasms. However, glioma image analysis is a challenging task, as the appearance and structure of the tumoral tissues is extremely heterogeneous. Automatic glioma segmentation is an open problem of clinical relevance, once it may help with a faster and more precise assessment, as well as better diagnosis and treatment planning.

For all this, it is not surprising, that several works of glioma and retinal vessel segmentation have been proposed over the years. This is, precisely, one of the focuses of this work.

Chapter 3

State of the Art

In this chapter the state of the art regarding ensemble learning and attention mechanisms is presented. Special emphasis is given to different ensemble strategies, where we divide ensemble learning methods into three groups, according to the approach used to build the ensemble: ensembling by varying the input data, models or combination procedures. Regarding the attention mechanisms, we present some of the most relevant works, with focus on those applied in the computer vision field. In addition, we describe some works using Fully Convolutional Networks in the context of medical image segmentation.

3.1 Ensemble Learning

The concept of Ensemble Learning emerged as another form of regularization by combining multiple hypotheses that explain the training data [44]. These methods can be shown both theoretically and empirically to outperform single predictors on a wide range of tasks. Today, ensembles are considered as the state-of-the-art approach for a cornucopia of machine learning challenges as reported in an extensive comparison of 179 classifiers from 17 families using 121 datasets from UCI[1], as well as various real-life challenges [45].

The field of ensemble learning is well studied and there are many variations on this theme. Different methods construct the ensemble of models in different ways [44, 46]. The most common ways of building an ensemble are by varying one of these three major elements of the ensemble method:

- Training data: vary the data used to train each model in the ensemble;
- Ensemble models: vary the choice of the models used in the ensemble;
- Combinations: vary the way that outcomes from ensemble members are combined.

In the following subsections, the most relevant methods of ensembling are described, divided into three different ways to create an ensemble.

[1] UCI machine learning repository: http://archive.ics.uci.edu/ml

3.1.1 Ensembling by Varying Input Data

The data used to train each member of the ensemble can be varied. The simplest way to vary the training data would be to use k-fold cross-validation. In this procedure, k non-overlapping subsets are used to train k different models. At each time, one subset of the data is used as test set and the rest is used as training set. The method has a low bias as each example in the dataset is only used once in the test dataset to estimate model performance. The test performance may then be estimated by taking the average performance across the k trials [46].

Bagging (short for bootstrap aggregating) developed by Breiman [47], is another popular technique that allows the same kind of training algorithm, model and objective function to be reused. In particular, bagging consists of constructing several distinct datasets, then training a network using these datasets. Each dataset is created by sampling with replacement from the original database. By sampling with replacement, if each dataset has the same number of data as the original one, some observations may be repeated in each dataset. Thus, every single resampled dataset is unique with a high probability of duplicated examples. Bagging was designed for use with decision trees, creating a particular instance of the random forest technique. Typically a large number of decision trees are used, such as hundreds or thousands [44].

3.1.2 Ensembling by Varying Models

The optimization problems that neural networks try to solve are so challenging that there are many different solutions to map inputs into outputs. Training the same system on different data will lead to different models but may not substantially improve generalization, as mistakes made by the models may still be too highly correlated [48]. An alternative approach for constructing an ensemble might be to vary the member models. This can be done by combining models from different topologies or using a completely different learning algorithm or objective function, or simply by training the same model under different conditions, e.g., with different hyperparameters [44]. It is generally believed that the ensemble members should be as accurate as possible and as diverse as possible, to get a robust ensemble [49].

Training ensembles of many diverse models takes a great amount of computational resources and time. When working with deep neural networks, where training a single model may take days to train, these factors are prohibitive [49]. In these cases, another alternative may be to periodically save models during the training run, called a snapshot, collecting networks from different weight space points, and then produce an ensemble with the saved models. This provides the benefits of having multiple models trained on the same data, although collected in a single training process. One such example is Snapshot Ensembles proposed by Huang et al. [50]. Multiple neural networks are ensembled at no additional training cost by training a single network, converging to several local minima along its optimization path and taking model snapshots on the distinct minima. At test time, their predictions are averaged to compute the ensemble. Another example is Fast Geometric Ensembling (FGE) designed by Garipov et al. [51]. This method is closely related to Snapshot Ensembles [50], however, it aims to find diverse networks without leaving the region that corresponds to the low test error of a pre-trained model, using shorter cycle lengths. The insight of the FGE was that there exist connected paths of low loss between sufficiently different models, it is possible to travel along those paths in small steps and the models encountered along will be different enough to allow ensembling them with good results.

3.1.3 Ensembling by Varying Combinations

The most prevailing approaches for combining the predictions of the ensemble members are simple averaging [47], weighted averaging [48], plurality voting and majority voting [52]. Simple averaging [47] of the predictions from each model is the simplest way to combine predictions, where each member contributes equally to predictions. This can be improved by using a weighted average of the outputs of component models [48]. Weighted averaging allows the contribution of each model to the ensemble to be weighted proportionally to its trust or performance on a hold-out dataset. The combination of ensemble members can be done by using voting methods, such as, plurality and majority voting [52]. The plurality method is the simplest form of voting. Every classifier has one vote, which it can cast for any one candidate predictions. The prediction with the highest number of votes wins. The major drawback of plurality voting is the risk of a win on a small number of votes and, so there is a high probability of an erroneous winner. To overcome this problem, in majority voting the possibility that achieves more than half the votes, i.e., the majority, prevails. This method only chooses a prediction in the case of majority, so the majority of the classifiers has to be wrong in order to produce an error.

There are many other more complex approaches for combining predictions. One further step in complexity involves training an extra model to learn how to best combine the ensemble members predictions. This model can be a learning algorithm that considers an input sample in addition to the contributions from each ensemble model to learn how to best combine the predictions. This generic method of learning a new model is called stacking, or stacked generalization [53]. There are more elaborated procedures for stacking models, such as boosting [54, 55] where ensemble members are incrementally added to correct the errors of previous models. This scheme tends to be likely to overfit the training data. Additionally, the augmented complexity means this method is less often used with large models.

Another somewhat different technique is to combine the weights of multiple neural networks with the same structure. The weights of multiple networks can be averaged, to hopefully result in a new single model that has better overall performance than any original model. This approach is called model weight averaging. The first and simplest implementation of weight averaging was introduced by Polyak and Juditsky [56] and involves computing the average of weights of the models over the last few training epochs. Recently, Izmailov et al. [57] proposed Stochastic Weight Averaging (SWA). In this methodology, the average of models corresponding to the end of the learning rate cycles is computed, aiming to achieve a solution with better generalization than the original model. This method approximates the FGE approach with a single model, and thus inference can be faster. Posteriorly, Athiwaratkun et al. [119] proposed fast-SWA motivated by the observation that, in SWA, the weight average is updated only once per cycle, which implies that in order to collect a considerable amount of weights, additional training cycles are required. Fast-SWA method extends SWA aiming to overcome this constrain. It captures the networks weights every j epochs, in each learning rate cycle, and averages them. By averaging multiple networks within each learning rate cycle, fast-SWA should further accelerate convergence to an optimal point.

3.2 Attention Mechanisms

The attention mechanism was born in the context of neural machine translation (NMT) using Seq2Seq models to help memorize long source sentences [58]. As the name suggests, Seq2Seq models take as input a sequence of words and generate an output sequence of words. Normally, Seq2Seq have an encoder-decoder architecture where both the encoder and decoder are recurrent neural networks, i.e. using long short-term memory (LSTM) or gated recurrent units (GRU). The problem with Seq2Seq models is the design of the fixed-length context vector that is built out of the encoder's last hidden state. Thus, often it has forgotten the first part once it completes processing the whole input. To solve this problem, Bahdanau et al. [58] implemented an attention mechanism by using a bidirectional LSTM for input and by introducing an alignment model, a matrix of weights connecting each input location to each output location. The alignment between the input and target sequences is learned and controlled by the bypassed context vector. This strategy enabled modeling of dependencies without regard to their distance in the input or source sequences.

Given the significant improvement by attention in NMT, it soon got expanded into the computer vision field where it has been popular in various tasks, such as, image captioning [59, 60], visual question answering [61], image classification [62–66] and image segmentation [67].

Motivated by the high computational cost associated with the application of CNNs to large images, Mnih et al. [62] explored a location-based attention mechanism for image classification. In their work, the main idea is to take inspiration from how the human eye works. The model is a recurrent neural network (RNN) that processes inputs sequentially, addressing different locations within the images one by one, and progressively combines information from these glimpses to create an internal representation of the input. Based on its internal representation of the input and the requirements of the task, the Recurrent Attention Model (RAM) outputs the next location to attend to. While this scheme refers to a hard attention mechanism, the model relies on reinforcement learning to train the non-differentiabilities, where a policy chooses the chain of glimpses that maximizes classification accuracy. Later, Ba et al. [63] extended the research work of Mnih et al. [62] to identify multiple objects using RNN and visual attention. They proposed a deep recurrent-based attention framework, the deep recurrent attention model (DRAM), that at each step processes a multi-resolution crop of the input image. The internal representation of the image is updated with the information from the glimpse, and the network selects the next location and the next possible object in the sequence. The process ends when the model determines that there are no more objects to recognize. DRAM biggest drawback is its limitation to datasets with a natural label ordering.

Another problem is that CNNs remain short on the ability to be spatially invariant to the input data in a computationally efficient manner. To address this problem, Jaderberg et al. [64] came up with Spatial Transformer Networks, wich are CNNs that contain one or more Spatial Transformer Modules (STM). These seek to incorporate spatial attention to make the network spatially invariant to its input data, because objects in images are usually at different scales, taken from random viewpoints and at random positions. STM explicitly allows the spatial manipulation of data within the network by rotating, cropping, scaling and/or warping an input image or a feature map in order to focus on the target object and to remove rotational variance. For that, it learns what affine transformation ought to be applied conditioned on input.

Still in the context of image classification, experiments on generating attention-aware features were conducted by Wang et al. [65]. The authors proposed the Residual Attention Network, which is a CNN built by stacking multiple residual blocks with attention modules. The stacked arrangement enables mixed attention mechanisms, as different modules capture different types of attention. Each attention module is branched off into trunk and mask branches. The trunk branch performs feature processing through residual units, and the mask branch adopts a bottom-up top-down structure, inspired by [68], that aims at enhancing trunk branch features. The gathering of two different learning strategies into the attention module enables fast feed-forward processing and top-down attention feedback.

CNNs make use of convolutional filters to extract hierarchal information from images. Lower layers find trivial pieces of context like edges, while upper layers can detect faces, text or other complex geometrical shapes. All of this works by fusing the channel and spatial information together within local receptive fields. In order to boost the representational power of a network, Hu et al. [66] introduced Squeeze-and-Excitation Networks (SENets), in the context of image classification, by proposing a building block, the SE block, for CNNs is a lightweight gating mechanism that improves channel interdependencies at almost no computational cost. When creating the output feature maps, standard networks weight each of its channels equally. The SE block is all about changing this by adding a content-aware mechanism to weight each channel. They get a global understanding of each channel by squeezing each feature map to a single numeric value and then scaling it based on its importance.

In the context of image semantic segmentation, Chen et al. [67] proposed an attention model that learns to weight the multi-scale features using soft attention as a scale selection mechanism. They feed multiple resized versions of the input images to Fully Convolutional Networks with shared weights and then merge the resulting features, with a weighted sum of feature maps, for pixel-wise classification.

3.2.1 Attention in Medical Imaging

In recent past, in the context of medical image segmentation, different attention strategies have been proposed using Fully Convolutional Networks.

Qin et al. [69] proposed an autofocus attention module that adaptively selects the optimal receptive field. The multi-scale information is captured by processing several parallel branches of convolutional layers with different dilation rates. The amount of focus to give to each scale is obtained by fusing the outputs from the parallel dilated convolutions with an element-wise weighted summation. Also, their proposal is scale-invariant, as the parallel filters in each module share parameters making each branch search for similar patterns. However, the increased representational power provided by the attention module comes with higher memory and computational requirements.

Oktay et al. [70] utilizes a novel soft-attention gate to enhance the region of interest. The proposed attention gate is incorporated as a soft mask into the standard U-Net architecture [71] to highlight important features passed through the skip connections. Nevertheless, this mechanism scales the same regions across all feature maps.

Roy et al. [72] adopted the feature recalibration strategy proposed by Hu et al. [66] to learn attention at the channel and spatial levels. Feature map recalibration is computed through linear contraction to lower dimensions followed by restoration of the number of feature maps. At each level, a single attention map is inferred for all feature maps. Their blocks are generic network components that learn to selectively emphasize discriminative features and suppress less useful ones by using global information. Recently, Pereira et al. [73] suggested a segmentation adapted block to overcome the drawbacks of the recalibration blocks. Their proposal collects contextual information while maintaining the spatial meaning and still considering the relations across channels, as context is fundamental to evaluate the spatial importance of features, in semantic segmentation tasks.

3.3 Summary

Ensemble methods have been widely studied in recent decades. Proof of this is the large number of works described in the literature. Depending on how each method is developed, the various state-of-the-art works may fall into one of three categories. Ensembling by varying the input data is based on training each individual model with different training data. Ensembling by varying models consists of training base models with different architectures and/or with different settings (optimizer, hyperparameters, or loss functions). Finally, ensembles can be obtained by varying the way that outcomes from ensemble members are combined.

Attention mechanisms allow systems to improve their ability to extract the most relevant information for each piece of the output, by focusing on distinct aspects of the input, thus yielding improvements in the quality of the generated outputs. Additionally, by selecting to only process subsets of the input, attention mechanisms reduce the computational burden of processing high dimensional inputs. Thus, they are now a habitual component of the deep learning toolkit, contributing to impressive results not only in NMT [58, 74] but also in image captioning [59, 60], visual question answering [61], image classification [62–66], and domain adaptation [75], among others. Finally, attention mechanisms have been used in the field of medical imaging. Effectively, this is an active line of research that continues to drive advances in medical imaging segmentation.

Chapter 4

Theoretical Principles

Machine Learning is the basis for all work developed in the scope of this dissertation. Therefore, in this chapter, we present a broad overview on Machine Learning and introduce the scientific background related to the methods conceived in this dissertation. First, artificial neural networks are focused, especially, convolutional neural networks. In line with it, we approach the training of such models. The central points of this dissertation are an ensemble method and attention mechanisms for semantic segmentation of medical imaging. Thus, some concepts about ensembling and attention mechanisms are also presented.

4.1 Machine Learning

Machine Learning, in very simple terms, is a set of methods that learn directly from the input data. Machine Learning algorithms try to learn a function f of the observable variable x. The basic procedure of Machine Learning is to give training data to a learning algorithm. The learning algorithm then originates a new set of rules, based on inferences from the features, in a process called training. During training, the model parameters are optimized to fit the data and learn to perform the task. A learning algorithm could be used to create models for different tasks, by using different training data. The task execution is called test or inference. This, in essence, means that Machine Learning enabled computers to be used for new, complex tasks that could not be manually programmed [44, 76].

4.1.1 Supervised Learning

Machine learning algorithms can be broadly divided into supervised learning and unsupervised learning. Throughout this dissertation, the approaches will be developed in a supervised learning context.

Supervised learning algorithms require a manually annotated set of images to learn a classifier model, i.e., a dataset in which each example is associated with a label or target. It involves observing several examples of a vector x and an associated vector y, then learning to predict y from x by estimating $p(y|x)$. During the training phase, an optimization algorithm analyzes a set of features and class associated with each training example and infers a set of parameters that the classifier uses to handle new examples in the test phase. Since supervised methods are designed based on pre-classified data, their performance is generally better than unsupervised methods [44, 77].

4.1.2 Classification task

In the scope of this dissertation, classification, one of the most common supervised machine learning tasks, will be emphasized. Classification algorithms attempt to estimate a mapping function, f, to assign each element of the input x into one of c classes. Thus, the mapping function can be defined as $f : R^n \rightarrow R^c$, where n is the number of features, and the prediction $\hat{y} \in \{1, ..., c\}$. The number of classes to distinguish from sets the classification mode. If $c = 2$, it is a binary classification problem, otherwise ($c > 2$) it is a multi-class problem. Some models infer a probabilistic prediction, where the output can be interpreted as the probability that a given pixel belongs to each of the classes, rather than directly predicting a class. This soft classification is often not only indispensable but also desirable, as it may be a measure of certainty about a prediction [44, 77].

In the context of medical imaging segmentation, given a set of pixels, i.e., an image or image patch, the goal is to classify each pixel to different classes, such as, different tissues, organs, pathologies, or other biologically relevant structures [78].

4.2 Artificial Neural Networks

The human brain can interpret real-world situations in a way that computers can not. It contains about 86 billions of neurons and more than 100 trillion synapses [79]. These neurons are arranged to form a complex network of nerves able to produce high complexity patterns. The neurons process the information received from our senses and synapses are connections that transmit the information from one neuron to another.

Artificial Neural Networks (ANNs) or Neural Networks are computing systems inspired by the structure of the brain. ANNs are composed of interconnected computational units which operation aims to approximate the behavior of the biological nervous systems [77].

An ANN learns to perform a task by considering examples, through a learning process. Nowadays, ANNs lead the state of the art in several machine learning problems, especially in classification tasks [44].

4.2.1 The Simple Perceptron

In 1957, Frank Rosenblatt [80] invented the perceptron. A perceptron is a computational model of a single neuron, thus, it is a single-layer feed-forward neural network, the simplest neural network possible.

A perceptron takes n inputs, represented by a vector $x \in \mathbb{R}^n$, weighs them separately (according to a weight vector w_1, w_2, ..., w_n), sums them up and passes this sum through a nonlinear function, the activation function, φ, to produce the output, $y \in \mathbb{R}$. An additional parameter, the bias (θ) has the effect of applying an affine transformation to the weighted sum of inputs [77]. The general model takes the form,

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + \theta\right)$$

The perceptron's decision boundary corresponds to a hyperplane in a n-dimensional input space. Thus, it can only divide the input space into two. So, we can conclude that the neuron is only capable of learning problems that are linearly separable [77, 81].

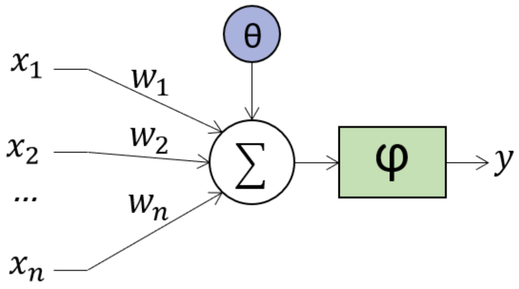


Figure 5: The simple perceptron.

4.2.2 The Multilayer Perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network. It consists of an input layer, an output layer and one or more hidden layers in between. Figure 6 shows a simple feedforward neural network with one hidden layer. If more than one hidden layer exists, it is said to be a deep artificial neural network. MLPs are called feedforward because information flows from the input, through the hidden layers, and finally to the output layer and there are no feedback connections. When feedback connections are included, they are designated recurrent neural networks (RNN) [44, 81].

The idea of using many layers of representations is drawn from neuroscience. We can think of the layer as consisting of several units that act in parallel, each unit resembles a simple perceptron in the sense that it receives input from other units and computes its own activation value, except for the input nodes. A unit in one layer is connected to all units in the previous and following layers, so they are called fully-connected layers [77, 81].

Hidden layers are not visible as a network output, the primary reason why they are referred to as hidden. They are used to handle non-linearly separable relations between the input and output. Each layer takes the former representations and learns to extract more complex and abstract features, before providing them to the next layer, by combining the signal and applying a non-linear activation function to the output of each unit [44]. For a given input $x \in \mathbb{R}^n$, the state of the i-th neuron is computed by $s_i = \varphi\left(\sum_{j \in m} w_{ij} s_j + \theta_i\right)$, where m represents the number of connections reaching the node i, φ the activation function, w_{ij} the weight of the connection between nodes i and j, θ_i the bias in the i-th node and $s_1 = x_1, ..., s_n = x_n$ [81]. The output y, results from a chain of functions corresponding to each layer. The depth of the model is determined by the length of this chain.

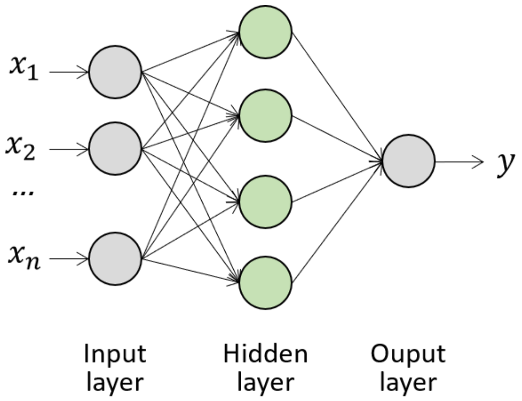


Figure 6: Example of a neural network diagram having a general feed-forward topology. The input, hidden, and output variables are represented by nodes, and the weight parameters are represented by links between the nodes. Note that each hidden and output unit has an associated bias parameter (omitted for clarity).

The input and output sizes are defined according to the problem. The input layer must have n nodes corresponding to n features and the output layer size match up with the classification task: it must have k nodes corresponding to k classes evaluated, $y \in \mathbb{R}^k$, apart from binary classification tasks where, usually, $y \in \mathbb{R}$. The number and size of hidden layers depends much on the problem and can vary even in identical problems. These hyperparameters must be carefully considered since they are of essential importance to the network performance [77, 81].

4.2.2.1 Activations Functions

Activations functions, φ, are non-linear transformations. A neural network without an activation function is essentially a linear regression model. The activation function makes a non-linear transformation of the input data, making the network capable to learn and perform more complex tasks. In a multi-layer ANN the activation function should be differentiable to make the back-propagation mechanism possible. Among these functions, the most used ones are the sigmoid function, the hyperbolic tangent function and the rectifier linear unit (ReLU) function [82].

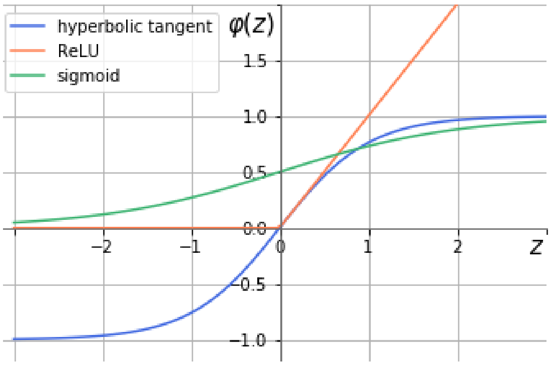


Figure 7: Activation functions.

Figure 7 compares these functions. As can be seen in Fig. 7, hyperbolic tangent and sigmoid functions have limited output ranges of, respectively, $]-1,1[$ and $]0,1[$, so, they saturate. The ReLU function doesn't suffer from this problem, as the gradient is unitary when z is greater than 0, helping to circumvent the phenomenon of vanishing gradient. One of the reasons ReLU (eq.1) is the most popular activation function for deep neural networks currently. Additionally, when z is lower than 0, the resulting activation of the ReLU is zero. This cancels some connections, making them more scattered and accelerating convergence [44].

$$\varphi(z) = max(0, z) \tag{1}$$

4.2.3 Convolutional Neural Networks

In the past few years, convolutional neural networks (CNNs) [83] had groundbreaking results not only applied to computer vision but also in Natural Language Processing. CNNs are a group of feed-forward neural networks that use at least one convolutional layer [44].

CNNs emerged from the idea of weight sharing. In particular, the insight that features are distributed across the entire image and invariant to translation. So, convolutional layers have shared heights and are locally connected so that, regardless of its position, one pattern can be detected. Therefore, CNNs enabled to largely reduce the number of parameters and consequently, increase depth in feed-forward neural networks [83, 84]. This achievement motivated researchers to approach larger networks in order to solve complex tasks, intangible with general ANNs [44].

Typical CNNs consist of two stages. In the first stage, the feature extraction is performed by the convolutional and pooling layers. The second stage, which corresponds to the last CNN layers, is composed of fully-connected layers (or dense layers) responsible for the classification [44]. These entail not only a rise in the number of parameters but also other problems that lead to them being discarded and substituted.

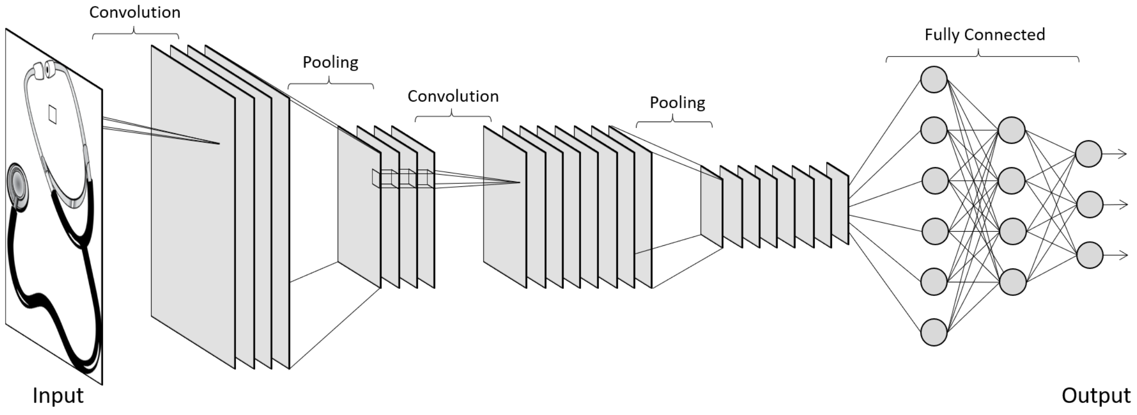


Figure 8: Convolutional Neural Network.

In this section, we start by clarifying what convolution is and the motivation behind convolution in neural networks. Then, we detail convolution variations. Next, we describe pooling operation and conclude by addressing the softmax layer.

4.2.3.1 Convolutional Layer

Key to the convolutional neural network is the convolutional layer, which is specially designed to take advantage of structured inputs, which are organized as grid-like structures, such as, temporal series (1D), images (2D) or videos (3D) [44].

The convolution operation is often interpreted as filtering, where a small matrix of weights, called a kernel, filters the input for certain information. Specifically, a flipped kernel is passed over the input, left to right, top to bottom, systematically overlapping a filter-sized patch of the input data. At each step, a dot product is computed between the filter and the filter-sized patch of the input. The outcome of a convolutional layer is typically referred to as feature map, and it is organized as an image-like grid structure as kernels are generally smaller than the inputs. The distinct feature maps in a stack are called channels. In the specific case of the first layer, the feature maps are the input image, which can have several channels, e.g. CMYK and RBG. Each feature map is the outcome of convolving the input with one single filter, hence, each convolutional layer must have as many filters as input channels [44].

Suppose we have two discrete functions, f and g, defined on t. The discrete convolution of f and g is defined as

$$s[t] = (f * g)[t] = \sum_{-\infty}^{\infty} f[x]g[t-x]. \tag{2}$$

In Deep Learning applications, the data (I) and kernel (K) are, generally, two-dimensional. Thus, the mathematical formulation of 2D convolution is given by

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m, j-n). \tag{3}$$

Note that CNNs often use multichannel convolution and the commutative property exists only if each convolution operation has the same number of output channels as input channels.

In practice, most neural network libraries implement cross-correlation but call it convolution, which is the same as convolution but without flipping the kernel:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(i+m, j+n)K(m,j). \tag{4}$$

This is more straightforward to implement, and it does not affect the performance of the algorithms. The kernel weights are learned during training, thus adding the flip would simply make the algorithm learn the weights in different kernel cells to accommodate the flip [44].

Fig. 9 illustrates a basic example of a convolution operation, where input and kernel only have one channel. Considering the outlined pixels, we have $0 \times 12 + 1 \times 23 + (-1) \times 9 + 1 \times 8 = 22$. Every entry in the output feature map can be interpreted as an output of a convolutional neuron that looks at a small region of the input, its receptive field [44].

A kernel is designed to detect a specific type of feature in the input and discard other information. The systematic application of the same kernel across the entire input allows the filter to discover that feature anywhere in the input, resulting in a high output value [44].

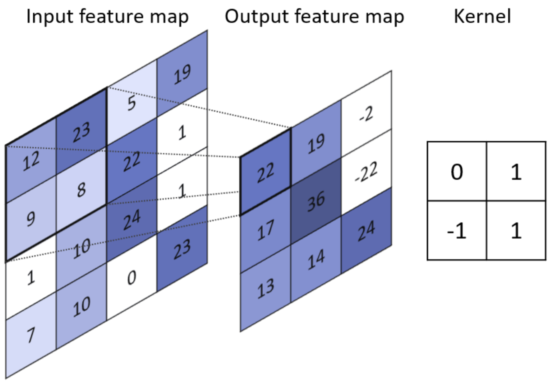


Figure 9: Example of a 2D convolution operation.

As more convolutional layers are added to the networks, deeper networks appear capable of detecting more complex features, such as patterns and objects. Given this fact, the feature engineering step of conventional methods can be obviated, which allows us to look at CNNs as networks capable of learning the best representation of input data. Thus, Deep Learning methods are also seen as a group of representation learning methods [44].

4.2.3.2 Motivation

Convolution leverages three important principles that underlie CNNs success over regular ANNs: sparse interactions, parameter sharing and equivariant representations.

Sparse interactions or sparse weights is accomplished by using kernels smaller than the input. In each layer of a traditional neural network, each output unit is connected with every input unit, with an individual parameter describing each interaction. Contrastingly, in CNNs, each output unit of a convolutional layer interacts only with its receptive field. This means the focus is on the local feature identification. It also means that it bears the advantage of reducing the number of parameters, which leads to fewer memory requirements and fewer operations to compute the output [44].

In a neural network, parameter sharing refers to the use of the same weight more than once. As mentioned above, in a regular ANN each element of the weight matrix matches one interaction between the input and output unit so, it is used only once when computing the output of a layer. In CNNs, each kernel is replicated across the entire input. The replicated matrix of weights shares the same parameters at every input position. This further reduces the memory requirements and does not affect the forward propagation runtime [44].

The convolution's particular way of parameter sharing allows for features to be detected regardless of their position in the input data. Hence, convolutional layers have the property of translation invariance. This is a very important characteristic of convolutional layers which make them better suited for image analysis, because, in natural images, objects may arise in several regions of the images. Convolution is not equivariant to other transformations, such as rotations or alterations in the scale. To handle these, other mechanisms are required [44, 84].

4.2.3.3 Variants and Hyperparameters of the Convolutional Layer

In the context of neural networks, the convolution operation differs from the standard discrete convolution operation [44]. In this section, we describe these differences and address some useful properties used in neural networks.

First, when referring to convolution in the context of CNNs, often it is based on several parallel convolution operations [44]. This is due to the fact that one kernel can only detect a single feature, in multiple locations, and each convolutional layer should be able to extract different features.

Second, as rule, the input data is not a 2D array but a 3D one. We will refer to these multidimensional arrays as tensors. In the 3D input tensor, two of the dimensions represent the spatial coordinates, width and height, and the third is the number of input channels [44].

Third, it is imperative to specify how the borders are treated. By default, the kernel starts at the left of the image with the left side of the kernel sitting on the far left pixels of the image. Thus, the pixels in the corner will only get covered by the kernel once and the center of the kernel will never overlap the corners nor the border pixels. This results in two downsides: reducing the height and width of the output compared to the input and losing information on borders of the input feature map. This can be overcome with padding. Padding, p, is a technique where rows and columns of zeros are added on every side of the input tensor. There are two typical options: valid and same. Valid convolution refers to the initial process described above, thus convolution is computed only in locations where the input totally contains the kernel. Considering $k \times k$ as the kernel size, the output feature maps will be $2((k-1)/2)$ units smaller than the input, which limits the number of convolutional layers. In same convolution mode, zero padding is applied evenly at each side in a way that the output and the input feature maps sizes match. This ensures that each pixel is given the opportunity to be at the center of the kernel and the number of convolutional layers is not limited but may introduce artificial information in the units closer to the borders [44].

Additionally, the distance between two successive kernel positions over the input is called a stride, s. The typical choice of a stride is 1, the kernel is moved across the image left to right, top to bottom, with a one-pixel column change on horizontal movements, then a one-pixel row change on the vertical movements. Nonetheless, it is possible to skip over some positions to reduce the computational cost, at the cost of not extracting the features as finely, setting $s > 1$. In turn, a downsampled output feature map is obtained [44]. An alternative technique to perform downsampling is the pooling operation, described in Section 4.2.3.4.

Last, the kernel size is a key hyperparameter in a convolution. It is easy to realize that smaller filter collects local information whereas bigger filters extract more global, representative and high-level features, so different sized kernels will detect different sized features in the input. In turn, it will result in different sized output feature maps [44]. Note that, in general, square filters with odd sizes are used, for convenience.

To sum up, kernel units are the only parameters automatically learned in a convolutional layer, during the training process. Yet, the number of kernels, the kernel size, padding and stride are hyperparameters that need to be set before the training starts, considering the given task and dataset.

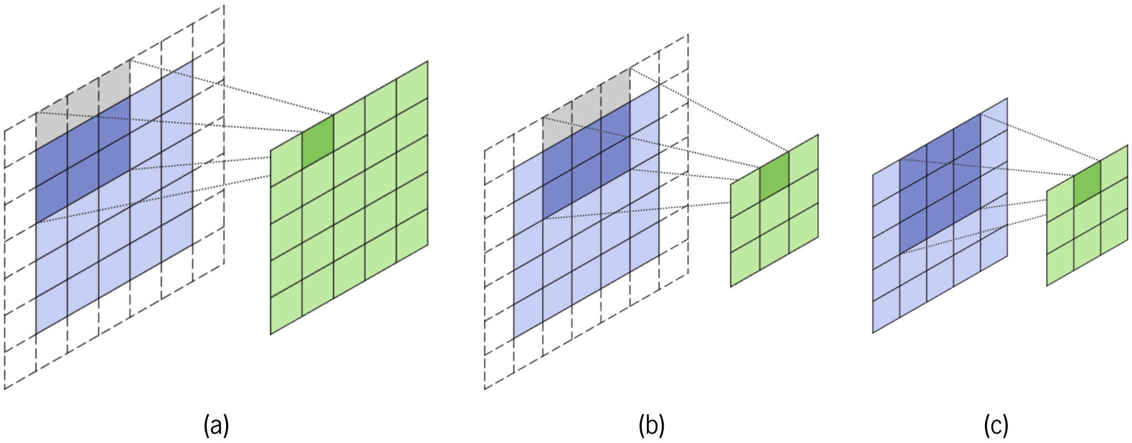


Figure 10: Convolution operation with a 3×3 kernel with different hyperparameters: (a) stride $= 1$ and padding same ($p = 1$); (b) stride $= 2$ and zero padding ($p = 1$); (c) stride $= 1$ and padding valid ($p = 0$).

Figure 10 illustrates convolution operations with different hyperparameters. If S_o is the output feature map size and S_i the input size, then $S_o = (S_i + 2p - k)/s + 1$. So we can validate the output sizes for Fig. 10. Note that, in all the examples, the input is a 5×5 map and the convolution is performed by a 3×3 kernel. So, $S_i = 5$ and $k = 3$. Thus, for the first case (Fig. 10a) $S_o = (5{+}2{\times}1{-}3)/1{+}1 = 5$, for the second (Fig. 10b) $S_o = (5{+}2{\times}1{-}3)/2{+}1 = 3$, and the last (Fig. 10c) $S_o = (5{+}2{\times}0{-}3)/1{+}1 = 3$.

Dilated Convolution Dilated convolutions [85] (or atrous convolutions, from the french expression *convolutions à trous*) were originally developed in an algorithm for wavelet decomposition. Dilated convolutions introduce one additional parameter to regular convolutional layers, the dilation rate (d), which enables to increase the receptive field at the same computational cost. The main idea is to insert holes (zeros) between the kernel elements enlarging the kernel size. A convolutional kernel with size $k \times k$ dilated by a dilation rate, d, has an effective size of $\hat{k} \times \hat{k}$ where $\hat{k} = k + (k–1)(d–1)$ [86]. Implementations may vary, but there are normally $d–1$ spaces between kernel elements such that $d = 1$ corresponds to a standard convolution.

Figure 11 illustrates an example of dilated kernels. A 3×3 kernel with a dilation rate of 2, while only using 9 parameters, will have the same receptive field as a 5×5 kernel.

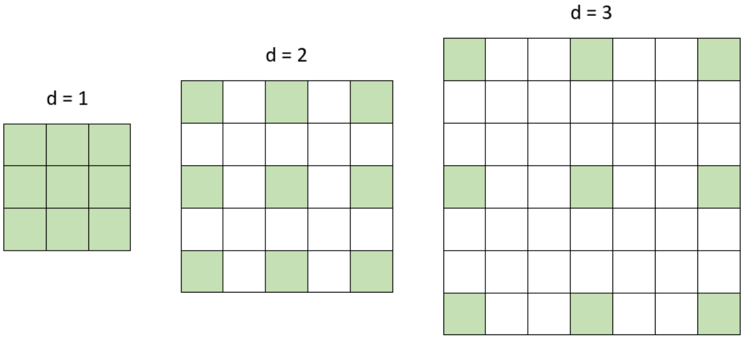


Figure 11: 3×3 convolutional kernels: regular kernel with $d = 1$ (left); dilated kernels with $d = 2$ (center) and $d = 3$ (right).

An inherent problem can occur when using dilated convolutions, the development of gridding artifacts [86, 87]. Since zeros are padded between two pixels in a dilated convolutional layer, the effective pixels that participate in the computation from the $\hat{k} \times \hat{k}$ region are just $k \times k$, with a gap of $d - 1$ between them, so the kernel only covers an area with checkerboard patterns, losing some neighboring information. The problem of gridding worsens when d increases, mostly in higher layers as the receptive field gets wider. The dilated kernel may be too sparse and local information can be completely missing. Additionally, the consistency of local information may be impaired, as the information that contributes to a fixed pixel always derives from a predefined gridding pattern. To obviate gridding artifacts, dilated convolutions with equal d or with common factor relationships should be avoided, and the composition of the layers must be thought through.

Depthwise Separable Convolution Depthwise separable convolutions [88] were initially introduced to reduce the computation in the first few layers of the models.

The standard convolution operation has the effect of filtering and combining features based on convolutional kernels to produce a new representation in one step. Depthwise separable convolutions approximate a standard convolution via the use of factorized convolutions: a depthwise convolution followed by a pointwise one. The depthwise convolution focuses on the spatial relationships, applying a single filter per each input channel. Then, the pointwise convolution, a simple 1×1 convolution, computes a linear combination of the output of the depthwise convolution, generating new features [88, 89]. Fig. 12 illustrates a basic example of a depthwise separable convolution operation, with a 3-channel input.

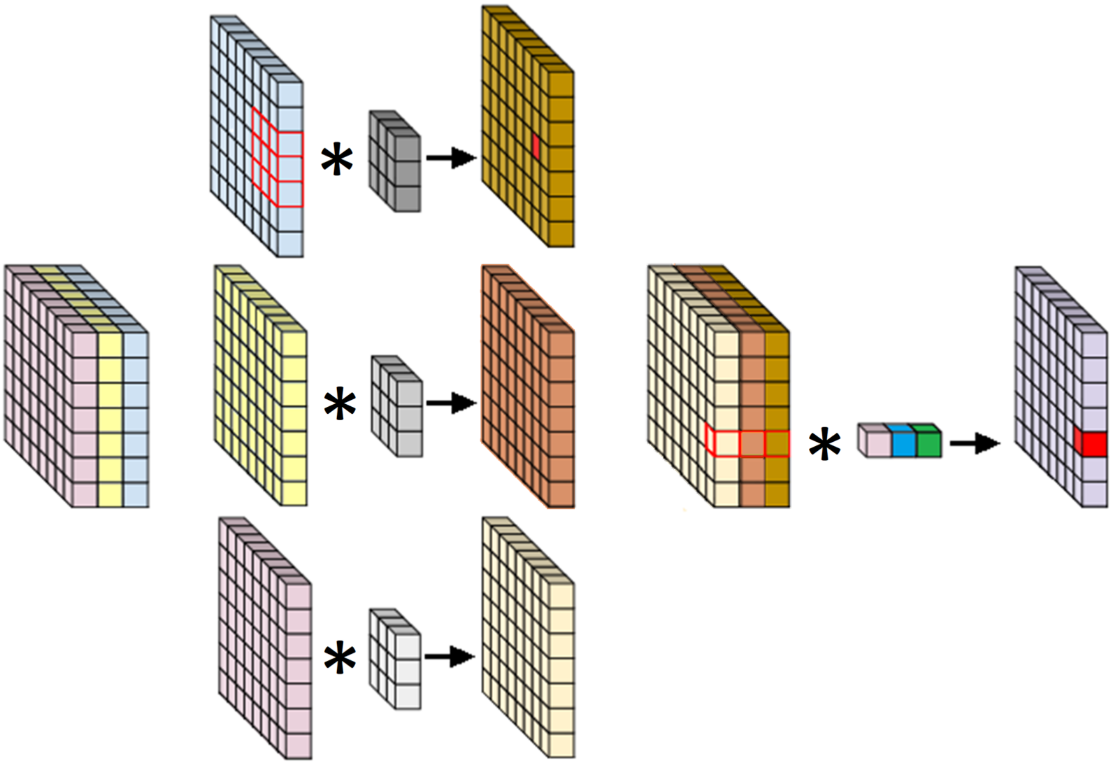


Figure 12: Depthwise separable convolution.

Recently, depthwise separable convolutions have become popular in deep neural networks as they have a lower number of parameters than regular convolutional layers. On that account, they are less prone to overfitting and also require fewer operations to compute, processing data faster. However, depthwise separable convolutions often lead to diminished representational power [89].

Examining the difference between the number of parameters, suppose there is a square filter, where F can be the filter width and height, and inC and $outC$ are, respectively, the number of input and output channels. Assuming padding same, where the spatial size of the output matches the input, in a regular convolution, as each filter is three-dimensional and there is one filter per output channel, there are $F \times F \times inC \times outC$ parameters. In its turn, in depthwise separable convolutions, for the depthwise convolution there are $F \times F \times inC$ parameters and $inC \times outC$ parameters for the pointwise convolution, totaling $(F \times F + outC)inC$ parameters. So, for $F > 1$ and $outC > 1$ and for $F = 1$ and $outC > inC$, it is straightforward that the number of parameters is significantly smaller for depthwise separable convolutions [89].

4.2.3.4 Pooling Layer

In a CNN, a typical layer includes three steps [44]. In the first step, several parallel convolutions are performed producing a set of linear activations. Next, a nonlinear activation function is employed to each activation generating a feature map. Last, a pooling function is used.

The pooling operation summarizes a given feature map region into some statistic. The desired function is computed over a neighborhood defined by a kernel. To reduce the computational cost of the next layers, generally, pooling with stride greater than 1 is employed. Pooling layers help to make CNNs invariant to small input translations. Furthermore, pooling increases the receptive field of the network, especially if stride greater than 1 is applied. Also, the output of the pooling operation is a downsampled feature map, where is created a lower resolution version of the input that maintains the larger or more significant features while discarding possibly vague and fine details, such as noise [44].

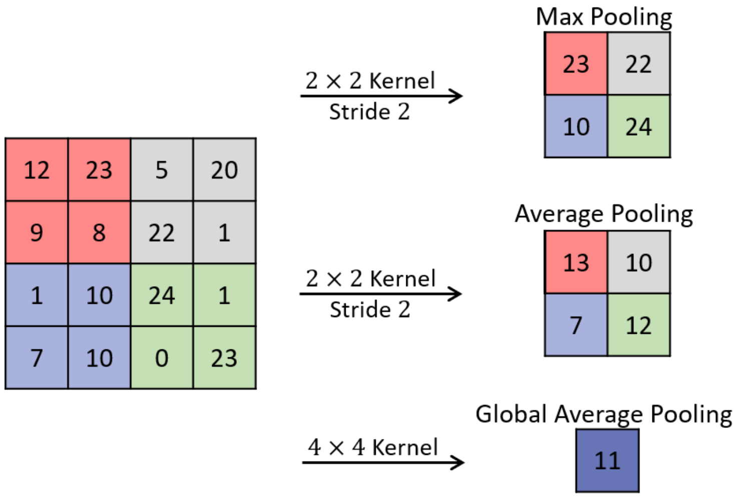


Figure 13: Pooling principle: max pooling (top); average pooling (middle); global average pooling (bottom).

The most commonly pooling function found in CNNs is the max-pooling [90] where the output is the maximum value for each rectangular neighborhood region, (Fig.13 - top). Other popular aggregating functions include the average pooling (Fig.13 - middle) and sum pooling in which the output is the average and the sum of the region values, respectively. There is an extreme type of pooling called global pooling, as the global average pooling [91] (Fig.13 - bottom). Global pooling downsamples an entire input feature map into a single value instead of downsampling kernel-defined regions. Usually, its goal is to fully or partially replace the fully-connected layers. The former is used to transition from feature maps to an output prediction for the network and the latter is used to form a feature vector to be fed to one or more fully connected layers [91, 92]. Generally, global pooling summarizes in the spatial dimensions, i.e., it is applied separately in each channel of the feature maps. For instance, a stack of features maps containing 8 channels of shape $8 \times 36 \times 36$ is pooled into an $8 \times 1 \times 1$ tensor.

4.2.3.5 Softmax Layer

In a classification task, the output layer should assign a probability to each class. Softmax function is most often used as the output of a classifier, to provide the probability distribution over the possible i classes. It can be seen as a generalization of the sigmoid function, which is used to represent a probability distribution over a binary variable, in binary tasks [44]. Formally, the softmax function is given by

$$\text{softmax}\left(y\right)_i = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}}. \tag{5}$$

Softmax normalizes the k outputs of the network into positive values that sum to 1.

If intended, the softmax function can be used inside the network itself, if desired the network to choose between different options for some internal variable [44].

4.2.4 Fully Convolutional Networks

Fully Convolutional Networks (FCNs) [93] - (Fig.14) - are neural networks composed of convolutional, pooling and upsampling layers, without any fully connected layers. They emerged in the context of semantic segmentation, which implies predicting a class for each pixel within the input image, instead of only one label for the whole input as in classification tasks.

The fully connected layers in typical recognition CNNs are fixed, confining the inputs to fixed sizes, and they overlook spatial iinformation. Nevertheless, these layers can be seen as convolutions with kernels that cover the entire input region. In practice, dense layers are replaced by convolutional layers with kernel size 1, once these assign an individual weight to each input unit, as the former. So, CNNs can be transformed into FCNs that accept arbitrary-sized inputs, segment a complete input patch at once, and output classification maps [93].

FCNs have been successfully applied to a number of applications significantly outperforming traditional learning-based approaches while improving computational efficiency. Hence, recently, FCNs have become the architecture of choice for image semantic segmentation. For this reason, FCNs with an encoder-decoder topology will be the base model for the work developed throughout this dissertation.

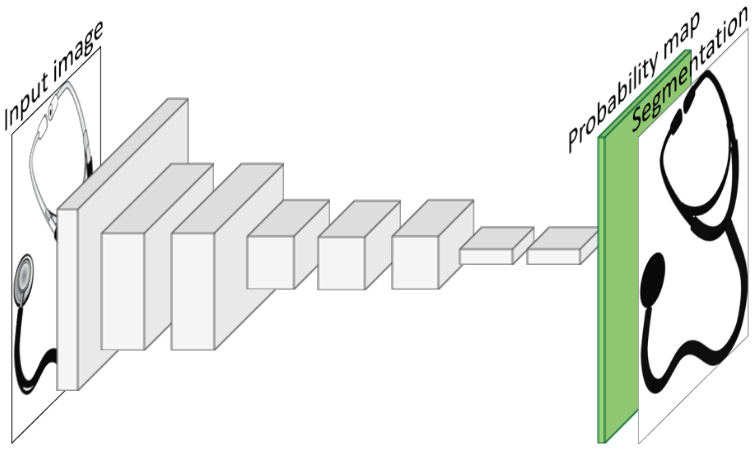


Figure 14: Fully Convolutional Neural Network.

4.3 Gradient-Based Learning

A feed-forward neural network is simply a chain of functions, where each layer represents a function. Thus, if the loss function is differentiable and all layers implement differentiable functions, it is possible to drive learning by gradient descent, through the minimization of the loss function. To do so, the learnable parameters of the model are adjusted in the direction of the negative of the gradient of the loss function, at a given point [44, 84].

In a neural network, the non-linearities cause loss functions to become highly non-convex, with several local minima. This means that neural networks trained with gradient descent drive the loss function to a minimum, without guarantees that it is a global minimum [44].

Computing the gradients may imply a high computational cost. The back-propagation algorithm [94] allows computing such gradients using an inexpensive and simple procedure. Specifically, the output values are compared with the true values to compute the loss. Then, the error is fed back through the network. Using this information, the algorithm adapts the weights of each layer, in order to minimize the loss function [44].

4.3.1 Initialization

Deep Learning algorithms require an initial specification of the parameters, from which to begin the training. Moreover, the weights initialization affects almost all Deep Learning algorithms. Improper initialization can lead to vanishing or exploding gradients driving to unstable training [44]. So, the initialization step can be critical to the ultimate performance of a model.

4.3.1.1 He Normal Initialization

He et al. [95] proposed an initialization scheme tailored for deep neural networks with non-linear activations, like the ReLU function. In He normal initialization, the weights are initialized with a normal distribution centered on zero and standard deviation $\sqrt{2/n}$, where n is the number of input units in the weight tensor. Bias are initialized to zero.

4.3.2 Regularization

In machine learning, regularization is a technique to avoid overfitting. Regularization is often done by adding some constraints, for example, a regularization term, in the model in order to constrain the parameter values. An effective regularizer is said to prevent the parameters to fit so perfectly the training data to overfit it [44].

4.3.2.1 Weight Decay

Weight decay is one of the ways to perform regularization. This regularization scheme decays the weights during learning by adding a parameter norm penalty to the weight update rule. During the process of overfitting the weighs get too big, so, by adding a regularization term, large weights will be driven down, in order to minimize the loss function [44, 77].

4.3.2.2 L^2 Parameter Regularization

In L^2 regularization, also called ridge regression, the regularization term is $\Omega(\theta) = \frac{1}{2} \parallel \theta \parallel_2^2$, which is the sum of the square of all weights, where the vector θ denotes all the parameters. Commonly, the L^2 regularization term is added to the loss function $J(w)$, where the vector w indicates all the weights that should be regularized, penalizing large weight values, so larger weights produce a higher error. The regularized loss function is given by $\tilde{J} = J(w) + \frac{\alpha}{2} w^T w$, where α is a hyperparameter that weights the relative contribution of the regularization. Mathematically, this forces a change in the weight gradients which, in turn, leads to a change in the values used to update the weights and reduces the magnitude of each weight, on each training iteration. As a consequence, this regularization is popularly known as weight decay. While L^2 regularization is the most common form of weight decay, there are other ways to perform weight decay [44].

4.3.2.3 Dropout

Dropout [96, 97] is another approach to regularization in neural networks. The role of dropout is to improve generalization performance by avoiding activations from becoming strongly correlated. Without dropout, each unit is always surrounded by the same neighboring units, during training, and hence, neurons can develop co-dependency amongst each other which curbs the individual power of each neuron. This, in turn, leads to overfitting of the training data because this interdependent learning does not generalize to unseen data. Dropout consists in ignoring units in a neural network, jointly with all its incoming and outgoing connections, during the training phase (Fig.15). By temporarily removing units for the network, dropout constantly compels each unit into new contexts, forcing the neural network to learn more robust features. For each iteration, each unit is randomly dropped out with probability p. In testing phase, the complete network is considered and each activation is reduced by a factor p, to account for the missing activations in the training phase [97].

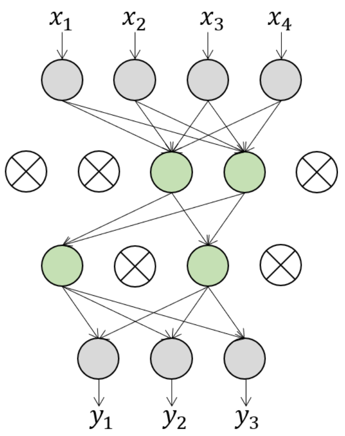


Figure 15: Dropout Effect. Dropped units are marked with an '×'.

Spatial Dropout In the context of CNNs, the concept of spatial dropout arose [98]. The standard dropout implementation disregards spatial correlation between the units that remain active. So, instead of dropping out units individually, spatial dropout ignores entire feature maps. Thus, all units across a feature map are either activated or set to zero, introducing additional constraints into convolutional layers so that each feature map will not be able to rely on other feature maps to explain training data.

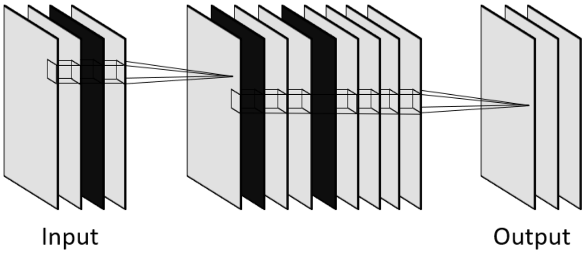


Figure 16: Spatial Dropout. Dropped feature maps are are colored in black.

4.3.2.4 Batch Normalization

Batch Normalization (BN) [99] is a method intended to mitigate internal covariate shift, i.e., the change in the distribution of network activations, in each layer, throughout training as the parameters change. During training, a BN layer calculates the batch mean $\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$ and variance $\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$, normalizes the layer inputs using these statistics $\bar{x_i} = (x_i - \mu)/(\sqrt{\sigma^2 + \epsilon})^1$, and scales and shifts the normalized values in order to obtain the output $y_i = \gamma\bar{x_i} + \beta$, where γ and β are parameters learned during training. Usually, a moving average estimate is computed instead and used during test time.

It turns out that BN makes the networks more robust to higher learning rate (LR) and less careful initializations. Also, the model is reparametrized in a way that noise is introduced, having a regularizing effect and decreasing the need for dropout [44, 99]. Contrastingly, Ioffe [100] pointed out the key drawback of BN: when dealing with small batch sizes, the batch mean and variance used during training are little representative of the training data and a less accurate approximation of the statistics used for inference.

[1] ϵ is a small positive value

4.3.2.5 Data Augmentation

In medical imaging, one major challenge is how to cope with the small data sets and the limited quantity of labeled data. Furthermore, one key characteristic of Deep Learning models is that they can benefit from significant quantities of training data, because the leaning to overfitting is reduced and, thus, the models achieve higher generalization [44].

One way of addressing the problem of the limited amount of data is by using data augmentation in the training set. In the medical imaging field, data augmentation is tipically done with transformations in both, images and ground truth, equally, generating warped versions of the training data that closely resembles the original training example. Rotations, reflections, translations and elastic deformations are commonly used for data augmentation [44].

4.3.3 Loss Function

In the context of an optimization algorithm for neural networks, the function used to assess a candidate solution, i.e., a set of weights, is referred to as loss function. Typically, the optimization algorithm seeks to minimize the loss. The loss is computed by matching the target and predicted values, using the loss function.

4.3.3.1 Categorical Cross-entropy Loss

In a multi-class classification problem, categorical cross-entropy is the default loss function. Categorical cross-entropy is used when a probabilistic interpretation of the output is desired and, when only one class is predicted for each unit. It requires that the output layer is set up with the same number of nodes as classes, and a softmax or sigmoid activation to predict the probability for each class. The measured dissimilarity between the true labels,y, and the probabilistic prediction, $\hat{y}$, is defined as the negative log likelihood:

$$J(\theta) = L(\hat{y}, y, \theta) = -\sum_{i} y_i \log(\hat{y}_i), \tag{6}$$

where i indexes the corresponding class.

From Eq. 6 it is possible to recognize some favorable properties of this loss function. First, if a sample is correctly predicted, it induces small gradients. Additionally, if the prediction is uncertain, the gradients are larger. Also, the more wrong the prediction, higher the gradient, leading to larger updates in the opposite direction to compensate it [44].

4.3.4 Optimization

As mentioned before, optimization is used as a training algorithm in neural networks, so learning can be achieved. In the context of Deep Learning, the optimization algorithm minimizes the loss function, updating the network parameters after every iteration [44].

4.3.4.1 Adam

First published in 2014, Adaptive Moment Estimation (Adam) [101] is one of the most popular gradient descent optimization algorithms. As one of the most important hyper-parameters during learning is, undoubtedly, the learning rate, some approaches try to automatically adapt it during training. Adam is an adaptive learning rate optimization algorithm designed particularly for deep neural networks training.

Adam computes adaptive learning rates for each parameter of the neural network. It can be looked at as a combination of Momentum [102] and Root Mean Square Propagation (RMSProp) [103]. The Momentum algorithm accumulates an exponentially decaying moving average of the gradient of the past steps and continues to progress in their direction, accelerating the search in direction of minima. RMSProp maintains per-parameter learning rates that are adapted based on the average of the latest magnitudes of the gradients using an exponentially decaying average so that it can converge rapidly in a convex basin [44]. Adam algorithm calculates exponential moving averages of the gradient and the squared gradient, respectively, the estimates of the first and second moments. It uses a moving average of the gradient instead of the gradient itself like Momentum, and it takes advantage of RMSProp by storing an exponentially decaying average of past squared gradients. The moments are initialized as zero leading to estimates biased around zero. Adam easily counteracts this initialization by including bias corrections [44, 101].

4.3.4.2 AdamW

As researchers start to apply Adam to train their models, it became apparent that plain Stochastic Gradient Descent (SGD) with Momentum [44] was performing better than Adam, discouraging its use. At the end of 2017, Loshchilov and Hutter [104] reported that the way weight decay is implemented in Adam seems to be wrong, and proposed a way do fix it, which they call AdamW.

When using SGD as the optimizer, the weight decay can be made equivalent to adding a L^2 regularization term to the loss. Due to this equivalence, L^2 regularization and weight decay are often referred to as the same [44]. However, when using any other optimizer, this is not accurate.

Loshchilov and Hutter [104] argue that L^2 regularization is not effective in adaptive algorithms, like Adam. In Adam, the L^2 regularization term is added to the loss function which is derived to calculate the gradients. But, if the weight decay term is added at this point, the moving averages of the gradient and its square keep track not only of the gradients of the cost function but also of the regularization term. In practice, this means that if the gradient of a certain parameter is high or changing much, the corresponding moment is large too and the weight is regularized less than weights with small and steady gradients. Hence, the regularization does not work as intended. Therefore, in AdamW, the weight decay original definition is followed, so it is applied simultaneously with the parameters update.

4.3.5 Learning Rate Scheduler

Training a neural network is a difficult optimization task. The learning rate is the most important hyperparameter to tune for a neural network. The LR decides how much of the loss gradient is to be applied to the current weights to move them in the direction of lower loss. It is well known that a LR too high will make the training algorithm diverge and may render optimization impossible, while if it is too low the learning may be to slow and may never converge or may get stuck on a suboptimal solution [105].

Common wisdom decrees that the LR should be constant or it should be a single value that monotonically decreases during training. Recently, several works agree that rather than a fixed LR value, a non-monotonic LR scheduling provides faster convergence [106] and, also, it has been claimed that the conventional wisdom that high learning rates should not be used may be flawed, and that high learning rates can lead to faster convergence [107]. Often, it is advantageous to start with a high LR and decrease it over iterations, following some schedule [44].

4.3.5.1 Warm Restarts

Loshchilov and Hutter [108] introduced a new LR scheme that combines warm restarts and cosine annealing. The authors found that SGD with warm restarts (SGDR) requires less than half training epochs than learning rate annealing to achieve comparable or better performance. In the proposed strategy, the LR decreases, following a cosine curve, during each cycle, and resets to a larger value when a new cycle starts. The cosine annealing enables the network to rapidly converge to a solution and the high LR values after a restart are used to catapult the parameters out of the minimum they previously converged to, into a different area of the loss surface. The learning rate at the t-th epoch is given by

$$\eta_t = \eta_{min} + 0.5\left(\eta_{max} - \eta_{min}\right)\left(1 + cos\left(\frac{T_{cur}}{T_i}\pi\right)\right), \tag{7}$$

where η_{max} and η_{min} are, respectively, the upper and lower limits of the LR. T_{cur} refers to the number of epochs passed since the last restart, T_i represents the number of epochs between restarts and $T_i = T_{mult} \times T_{i-1}$, meaning the period T_i is increased by a factor of T_{mult} after each restart. It is also possible to decrease η_{max} and η_{min} at every new restart.

4.3.5.2 One-Cycle Policy

Recent works [107, 109], suggest a new training methodology with improved speed and performance. They proposed a slight modification of cyclical learning rate (CLR) in which the LR scheduling consists of only one cycle, hence, the one-cycle policy name. In their setting, they use one cycle that is smaller than the total number of epochs and, for the remaining epochs, decrease the LR several orders of magnitude less than the starting LR. They showed that the one-cycle policy enables an improvement in the accuracy in several classification tasks.

4.4 Ensemble Methods

Ensemble methods are machine learning techniques that combine multiple classifiers to improve predictive performance. The main idea is that individual performances can be leveraged through ensembling as the errors of a single classifier will likely be compensated by the others when combining multiple networks [44]. Each network makes different mistakes hence, the collective prediction produced by the ensemble is less probable to be in error than the prediction obtained by any of the individual classifiers. In other terms, an ensemble will, at least, have as good performance as any of its members, and if the member models make independent mistakes, it will perform considerably better than its members. Therefore, to get a robust ensemble, the ensemble members should be as diverse as possible, and as accurate as possible [44, 46, 52].

Different ensemble methods construct the ensemble in distinct ways. Generally, it takes two steps to build an ensemble: training the base models and combining their predictions, as depicted in Figure 17. Each member of the ensemble can be generated by training a model with different network architecture, for example. Additionaly, the combination of the ensemble members may be done by simple averaging [47] or more complex procedures, such as, training a new algorithm to combine predictions [53]. Thus, composing an ensemble entails much larger computational cost than creating a single model [44, 110].

Next, we will explain one of the major problems ensembles try to mitigate. Additionally, we will summarize the principles of weight averaging, as one of the focuses of this work will be the use of weight averaging as a method to improve network performance.

4.4.1 The Bias-Variance Trade-off

Machine learning models are influenced by an important problem known as the bias-variance dilemma [111]. The conflict lies in trying to simultaneously minimize bias and variance, two sources of error that hamper learning algorithms from achieving good generalization. Bias and variance are side effects of one factor: the complexity of the model. Thus, their relationship is tightly related to the concepts of capacity, overfitting and underfitting [44].

Models with low bias present higher complexity, being able to represent the training data accurately at risk of overfitting by also capturing noise along with the underlying pattern leaving less scope for generalization. Thus, these models are highly sensitive to small fluctuations in the data, having high variance error. By contrast, high-bias models are generally simpler models, with a low number of parameters, that tend to underfit the training data being unable to capture its underlying pattern. However, these may produce lower variance predictions when applied to unseen data. Ideally, one wants to select a model that both accurately captures the essence of its training data, but also generalizes well to new data [44, 111].

One way to address this trade-off is ensembling. By combining multiple models, ensemble methods are designed to improve stability, helping to minimize these sources of error. For example, bagging [47] combines models in a way that lowers their variance, while boosting [54] combines high-bias learners in an

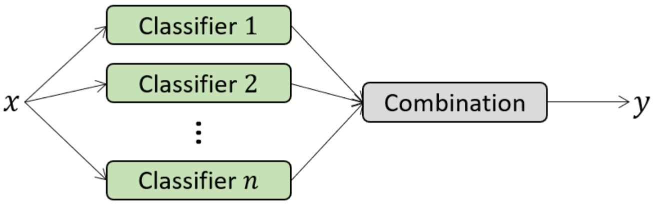


Figure 17: Common ensemble architecture.

ensemble that has reduced bias than the individual models. Likewise, the conventional way of composing ensembles by training multiple models and combine their predictions, illustrated in Figure 17, adds a bias that counters the variance of the individual models, resulting in less sensitive predictions to the specifics of the training data and choice of training scheme of a single training run.

4.4.2 Weight Averaging

There are many sophisticated methods for ensembling models. When dealing with Deep Learning models, the added complexity of training networks means ensembling several networks is less often used. As a consequence, ensemble strategies that use a single model have emerged.

Weight averaging was first regarded in convex optimization of neural networks by Polyak and Juditsky [56]. Learning the weights for deep models requires solving a high-dimensional non-convex optimization problem, where the learning algorithm can bounce around and fail to settle in one solution. To address this issue, one approach is to average the weights collected towards the end of the training process, resulting in a final set of weights that may provide a more stable, and perhaps more accurate performance [44].

This idea was then utilized by Izmailov et al. [57] to produce an ensemble using a single training process, with a procedure called Stochastic Weight Averaging (SWA). SWA is not an ensemble in its conventional understanding, as you get one model at the end of the training. SWA is a training procedure that averages multiple sets of weights collected during training using a learning rate schedule that allows the exploration of regions of the weight space that correspond to networks with high performance. This strategy is based on the observation that, at the end of a learning rate cycle, optimizers converge to points near the boundary of local minima regions. Thus, taking the average of several such points can lean the cost function to a point centered in this region, achieving a solution with better generalization.

Several authors have observed the connection between flatness and generalization. The theory that models with flat minima tend to have good generalization was reinvigorated by Keskar et al. [112] and later observed by Izmailov et al. [57]. The broad justification is that the training loss and test error surfaces are shifted relative to each other and, thus, a solution centered in a wide optima region is more robust, maintaining approximately the performance under small perturbations.

4.5 Attention Mechanisms

In recent years, attention mechanisms are a prevailing component in state-of-the-art models. In broad terms, attention is one component of a network's architecture, and it is in charge of managing and quantifying the interdependencies between the input and output elements [44].

4.5.1 Human Visual Attention

The human visual perception does not tend to process an entire visual space at once. In fact, the feel of observing a whole scene in high resolution is an illusion created by the subconscious part of our brain [113, 114]. Instead, humans make several eye movements, called saccades, focusing attention selectively on parts of the scene to acquire information of the most visually salient or task-relevant areas and combine information from different glimpses of small areas over time to build up an internal representation of the visual space in which the local and global features combined guide object detection and recognition [62, 115]. Motivated by the visual attention mechanism in the human visual system that gives humans the capability to pay attention selectively to the part of the image, instead of processing the whole scene in its entirety, attention mechanisms for neural networks arose [44].

4.5.2 Attention in Neural Networks

Integrating attention mechanisms into artificial neural networks is an active research direction [44]. The idea of attention for Deep Learning networks can be viewed, roughly, as a tool to guide the allocation of available processing resources towards the most informative elements of an input signal [62, 113]. In fact, the purpose of attention mechanisms is to allow the system to focus on the most relevant information allowing modeling of dependencies and disregarding the noise or less discriminative features. The benefits of such mechanisms have been shown in a range of tasks. It is commonly implemented in combination with sequential techniques, e.g., recurrent neural network (RNN), and a gating function, e.g., softmax and sigmoid [44].

Attention mechanisms can be grouped into classes. Thus, next we will summarize the broader categories of attention mechanisms: soft and hard attention, global and local attention, and self-attention.

4.5.2.1 Soft and Hard Attention

Xu et al. [59] first proposed the distinction between soft and hard attention, based on whether the attention has access to the entire input or only to a part of it. In the case of soft attention, the output of the attention module is a weighted sum of the input states. The weights on each location are usually given by a softmax function. The whole model is smooth and differentiable which is fundamental for calculating the error gradients, to back-propagate gradients, and to update parameters, so learning end-to-end is trivial by using back-propagation. Contrastingly, hard attention focuses on the most relevant part of the source sequence: the attention weights are used to select a single state, for example, the one with the highest attention weight. The mandatory selection operation, like the argmax function, is not a continuous function and hence, not differentiable. Therefore, the model requires more complicated techniques to train, such as variance reduction or reinforcement learning. Note that, the distinction between hard and soft attention is purely based on the selection method, it has nothing to do with the attention calculation mechanism.

4.5.2.2 Local and Global Attention

Luong et al. [74] proposed global and local attention, in the context of machine translation. The global attention configuration is similar to the soft attention, while local attention is a blend between hard and soft attention, an improvement over the hard attention to make it differentiable. The idea behind global attention is to use all states to define the attention weights, which can be computationally costly. Conversely, in local attention, the model peaks a position, per target, in the input sequence and the states

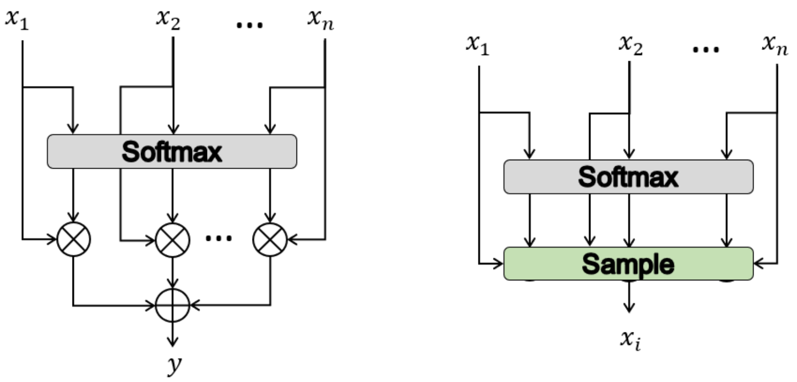


Figure 18: Attention mechanisms: (a) soft attention; (b) hard attention.

that fall within a window centered around the source position are then used to compute the attention weights. The choice of the central position, per target, varies with the assumption of the source and target sequences alignment. This chosen position in the source sequence will determine a window of states that the model attends to, reason why local attention is sometimes referred to as window-based attention.

4.5.2.3 Self-Attention

Cheng et al. [116] used self-attention, also known as intra-attention, to do machine reading. This attention mechanism relates different positions of a sequence in order to compute a representation of the same sequence. We can use self-attention to generate a richer representation of a given input x_i, with respect to all other input units $x_1, x_2, ..., x_n$. It enables us to learn the correlation between current states and the previous states of the sequence. Self-attention has been shown to be effective, also, in machine translation [117] and abstractive summarization [118].

4.6 Summary

Machine Learning models learn to perform a given task from the data. Artificial neural networks lead the state of the art in classification tasks. The term Deep Learning comes when using artificial neural networks with several stacked layers. These are capable of learning directly from the data more complex and discriminative features. In Deep Learning methods, current successful methods are trained by gradient-based learning. Within these, convolutional neural networks stand out for being supervised methods that replace conventional matrix multiplication with convolution operation. Fully convolutional networks are a variant of traditional convolutional networks where dense layers are excluded, allowing you to handle inputs of any size and increase computing efficiency.

The recent noteworthy performances achieved with Deep Learning often come at the cost of more complex models. If used properly, some variations of the regular convolution operation manage to enhance efficiency without reducing effectiveness and, while not increasing the complexity. Of these, dilated convolutions and depthwise separable convolutions stand out. Dilated convolutions are used for aggregating context, increasing the receptive field without increasing the computational cost. Depthwise separable convolutions originated from the idea that depth and spatial dimension of a convolutional filter can be separated. These have smaller number of parameters to adjust, as compared to the regular convolutions, which reduces overfitting while being computationally cheaper.

Ensembe methods are machine learning techniques based on the principle that by combining multiple classifiers, the errors of each model will be mitigated by the others resulting in a classifier whose overall performance is superior to individual models. Since this methods perform remarkably well, strategies based on ensemble methods have reached the top in many medical imaging competitions.

Among the more recent Deep Learning approaches, the incorporation of attention mechanisms often achieve good performance. These mechanisms are loosely based on the human visual attention mechanism, allowing the network to focus on the most relevant features by selecting certain regions of the input and modeling dependencies between features.

Chapter 5

Stochastic Weight Averaging for Retinal Vessel Semantic Segmentation

As seen in Chapter 4, ensemble methods are often used to enhance the performance of classifiers. In this section, a retinal vessel semantic segmentation system using ensemble methods is presented. We focus our work on stochastic weight averaging (SWA) and fast stochastic weight averaging (fast-SWA). First, the ensemble procedures employed are explained. Afterwards, we describe the steps of the automatic segmentation strategy applied, the databases and evaluation metrics used to validate the segmentation system. At last, we validate the SWA hyperparameters for retinal vessel segmentation, and, having finished our ablation study, we compare our results with state-of-the-art works.

5.1 Methodology

In this work, we consider two ensemble approaches. In this section, we will introduce the methodology for retinal vessel segmentation using stochastic weight averaging.

5.1.1 Stochastic Weight Averaging

Recently, procedures have been proposed to benefit from the advantages of ensembling while reducing the time to produce a set of models [50, 51, 57]. In this work, we adapted the work of Izmailov et al. [57] for our semantic segmentation task, applying SWA for retinal vessel segmentation.

SWA overcomes the drawback of the higher amount of computation needed when ensembling multiple models. Starting from a pre-trained network, SWA explores the weight space with the goal of finding a better performing solution centered in a region of high-performing networks, by using a cyclical learning rate scheduler. The use of a learning rate scheduler that skips directly from minimum to maximum learning rates is important for purposes of exploration of the loss surface. During training, SWA captures the networks weights corresponding to the minimum value of learning rate in each cycle and averages them. The model obtained with this running weight average is then used to compute predictions.

5.1.2 Fast Stochastic Weight Averaging

Motivated by the observation that in SWA the weight average is updated only once per cycle, which implies that to collect a considerable amount of weights, additional training cycles are required, Athiwaratkun et al. [119] proposed fast-SWA. This method extends SWA and changes it by averaging multiple network weights within each learning rate cycle. Fast-SWA captures the networks weights every j epochs and averages them in order to further accelerate convergence to an optimal point. Analogously to SWA, the predictions are computed using the averaged model. Note that some of the models included in fast-SWA may have higher error rates than those averaged in SWA because they are obtained when the learning rate is not minimum.

5.1.3 Batch normalization

We compute the Batch normalization statistics for the SWA and fast-SWA networks, as described in Garipov et al. [51], because these are not calculated during training since the final weights result from averaging weights throughout training. In practise, we run one extra pass over the data, after the training is concluded, to collect the running mean and standard deviation for each layer of the network with the averaged weights.

5.1.4 Computational Complexity

During SWA and fast-SWA training, it is only required to update the running weight average and to store a copy of the averaged weights. The operation to update the average takes the form

$$w_{SWA} \leftarrow \frac{w_{SWA} \cdot n_{models} + w}{n_{models} + 1}, \tag{8}$$

which is essentially a weighted sum of two networks. Note that, these operations are applied at most once per epoch. Also, when training is complete, only the model with the weight average is saved. Therefore, the computational requirements of SWA and fast-SWA procedures only involve the training time needed, since the memory overhead is negligible.

5.2 Experimental Details

In this section, we describe the steps of our automatic segmentation strategy, the databases, evaluation metrics and training setting used to validate the vessel segmentation system. We finish by describing the software and hardware used for the development of the proposed methods.

5.2.1 Retinal Vessel Segmentation Setup

An overview of the different steps that define the vessel segmentation algorithm is presented in Figure 19. Two phases can be highlighted: pre-pocessing and classification. In accordance with the structure of the proposed methodology, the practical implementation of each of the constituent phases will be described.

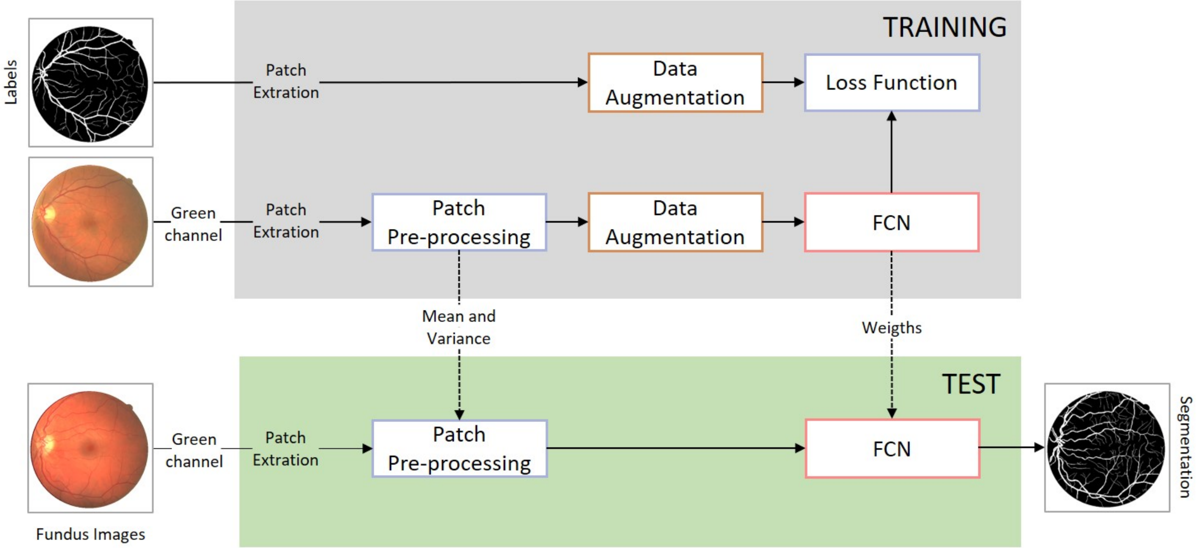


Figure 19: Overview of the proposed automatic retinal vessel segmentation system.

5.2.1.1 Pre-processing

Retinal fundus images are acquired in color and can be split into three channels according to the RGB system: red, green and blue. The red and blue channels are typically noisier and have a smaller dynamic range. On the other hand, the green channel shows the best contrast between vessels and background [120]. Therefore, throughout this work, only the green channel was used, along with most of the state-of-the-art methods. After extracting the patches from the green channel, each patch was normalized to have zero mean and unit standard deviation. This normalization allows the intensities to be confined to a more uniform range and helps to accelerate convergence [81].

5.2.1.2 Data Augmentation

One strategy was used to generate new artificial patches by transforming the original green channel patches. There is a need to match the annotated patch to its corresponding image patch so the transformation was applied to both. The strategy was to augment the training set by introducing rotations, which are quite common when performing data augmentation. Only rotations at multiple angles of $90°$ were used to avoid interpolation. In particular, each patch can be subjected to rotations of $90°$, $180°$, $270°$ or no rotation. The choice of rotation angle was modeled using a multinomial distribution with $p = 0.25$.

5.2.1.3 Baseline Architecture

The baseline FCN architecture utilized for SWA experiments is shown in Fig. 20. The implemented network is inspired by an encoder-decoder architecture with long skip connections proposed by Ronneberger et al. [71].

The network consists of regular convolutional blocks (RCBs), dilated convolutional blocks (DCBs) and a final convolutinal block (Fig. 20 - B3 block). The RCB is composed of two 3×3 convolutional layers without dilation, followed by Batch Normalization and ReLU, the former layer is still followed by dropout. The DCB was designed based on the work of Yu et al. [87] and Wang et al. [86]. DCBs are comprised of a 1×1 convolution followed by a set of 3×3 convolutional layers with different dilation rates, with subsequent Batch Normalization, ReLU and dropout. The initial layers present residual connections, and the dilation rate increases from 1 to 3. In the final layers, the dilation rate value is progressively lower, decreasing from 2 to 1. In the contracting path, after the second RCB, 2×2 convolution with stride 2 is applied to generate higher-level feature maps and the number of feature maps is doubled. In the expanding path, the feature maps are upsampled and their number is halved. Then, the lower level feature maps are added to higher-level ones, through a long skip connection. This encoder-decoder architecture with dilated convolutions enables the network to capture both context and precise location. The dilated convolutions in the encoder increase the receptive field while maintaining the spatial resolution and removing the need for further downsampling operations. At the final layer, the feature maps are cropped, and a 1×1 convolution followed by softmax is used to map each feature vector to the number of classes and compute prediction (Fig. 20 - B3 block). The input for the FCN is image patches extracted from the green channel of fundus images.

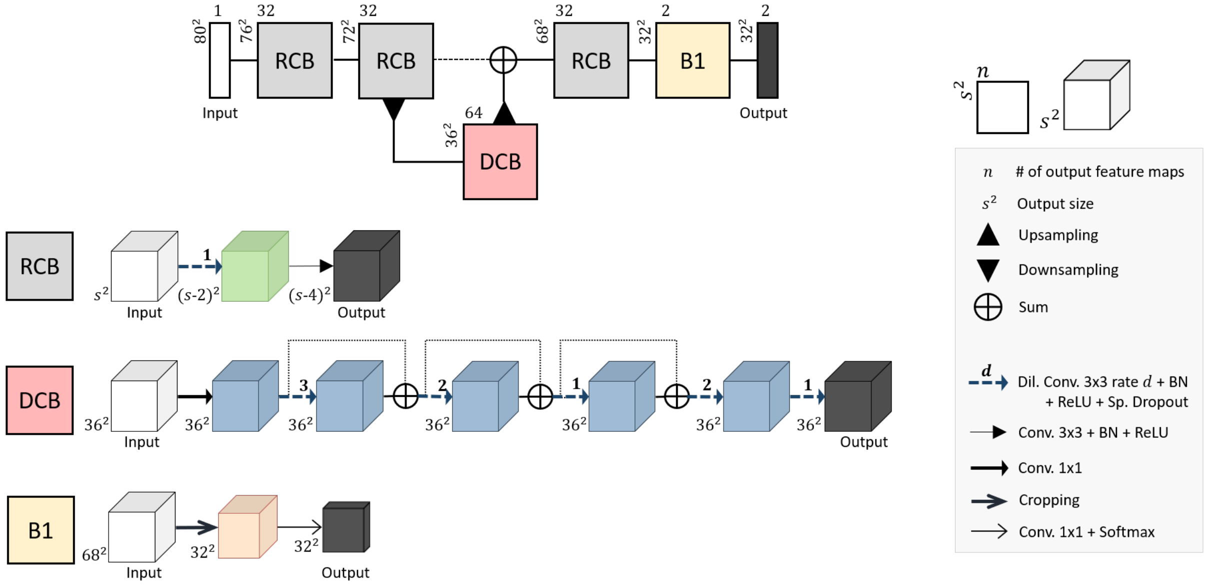


Figure 20: Architecture of the baseline FCN for retinal vessel segmentation.

5.2.2 Databases

During the course of this project, the proposed method was validated for retinal vessel segmentation. To do so, the method was tested with the publicly available DRIVE database [121]. Afterwards, to verify the robustness of the method, it was evaluated in other two publicly available databases: STARE [120] and CHASE_DB1 [122]. The DRIVE, STARE and CHASE_DB1 databases have been established to enable comparative studies on segmentation of blood vessels in retinal images.

5.2.2.1 DRIVE

DRIVE database Staal et al. [121] contains 40 fundus images with a resolution of 565×584 pixels and 8 bits per channel, 7 show signs of ocular pathologies. DRIVE is splitted in training and test sets, each of them with 20 images. From the total 20 training subjects, we used 18 randomly selected as training set, the 2 remaining for validation, and the performance was evaluated in the test set. There are two manual segmentations available from two independent human observers for DRIVE. For performance evaluation, the annotations of the first human observer were used as ground truth.

5.2.2.2 STARE

The STARE database (Hoover et al. [120]) contains 20 images with a resolution of 700×605 and 8 bits per channel, 10 of which showing pathological signals. For STARE, an explicit division in train and test sets is not available. In light of this, the models were trained using a stratified k-fold cross-validation, where the global set was divided into k equal-sized folds. STARE images were divided in 5 folds of 4 images. We employed the folds stratification following Oliveira et al. [138] which ensures the data is evenly splitted across the folds. In each fold, half of the images show pathological signals. The cross-validation procedure is repeated k times, yielding one fold as the test set, and the remaining k–1 folds for training and validation, being 14 and 2 images used for training and validation, respectively. To obtain the model's performance the cross-validation results are averaged. The networks were trained from scratch in each fold. There are two manual segmentations available from two independent human observers for STARE also. The annotations from Hoover were selected for STARE. We used the binary masks created by Oliveira et al. [138] as the masks for STARE images are not publicly available.

5.2.2.3 CHASE_DB1

CHASE_DB1 database (Owen et al. [122]) has 28 images, each with a resolution of 999×960 pixels, collected from both the eyes of 14 children. An explicit division in train and test sets is not available. Thus, the models were trained using a stratified k-fold cross-validation, where the global set was divided into k equal-sized folds. For CHASE_DB1 images we settled $k = 4$, having 4 folds of 7 images. We employed the folds stratification following Oliveira et al. [138], which ensures the data is evenly splitted across the folds. In each fold, in the CHASE_DB1 case, 3 images of one eye and 4 images of the other were included. The cross-validation procedure is repeated k times, yelding one fold as the test set, and the remaining k–1 folds for training and valitation, namely, 19 images for training and 2 for validation. To obtain the model's perfomance the cross-validation results are averaged. The networks were trained from scratch in each fold. There are two manual segmentations available from two independent human observers for each database. For performance evaluation, the manual segmentations of the first human observer were used as ground truth for CHASE_DB1. The binary masks for CHASE_DB1 images are not available, so we used the binary masks created by Oliveira et al. [138].

5.2.3 Training Settings

In this section, we present the implementation details for the proposed models.

5.2.3.1 Baseline Network

The baseline models were trained for 54 epochs with a batch size of 4. He normal [95] was used to initialize the weights of the convolutional layers. Categorical Cross Entropy and Adam [101] were selected 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 [108]. In the first cycle, with a length of 4 epochs, the learning rate and 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.

5.2.3.2 Stochastic Weight Averaging

For the SWA and fast-SWA training setup, we use the cosine annealing learning rate scheduler with periodic restarts [108] with Adam [101] optimizer, and Categorical Cross Entropy loss function. SWA and fast-SWA procedures were computed over the pre-trained baseline model. The model weights corresponding to the best epoch are utilized as initialization.

For SWA, we capture the networks weights corresponding to the minimum values of each learning rate cycle, as shown in orange in Figure 21, whereas, for fast-SWA, we average the models every $j = 2$ epochs, as shown in green in Figure 21. For SWA and fast-SWA, we set the cycle length fixed to 4 and kept it constant throughout both ensemble methods, so that the methods exploit the weight space with relatively-small steps that are still sufficient to get diverse networks [51].

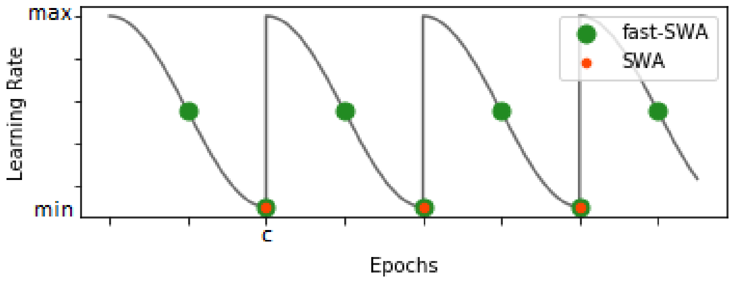


Figure 21: Cosine annealing learning rate scheduler and SWA and fast-SWA averaging strategies.

5.2.4 Evaluation Metrics

As already mentioned, in a problem of image segmentation the objective is to classify each pixel as one of the classes defined for the problem. In the case of vessel segmentation, the pixels can be defined as vessel or background. When comparing segmentations with manual annotations, each pixel can be framed in one of four groups: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Based on these groups, it is possible to define several recurring metrics, including sensitivity (Sens), specificity (Spec), accuracy (Acc), and the Matthews Correlation Coefficient (MCC).

For evaluation of vessel segmentation, we use Acc, Sens, Esp, MCC and AUC. These metrics are computed by comparing the segmentation results with the ground truth, only for the pixels inside the field of view. In general, Sens translates the ability of the model to detect vessels, while Spec allows us to understand the extent to which this detection capacity is obtained at the expense of the inclusion of false detections. In turn, Acc summarizes the overall performance in segmenting all the pixels [123]. The MCC is a measure of the quality of a binary classification. It can give more insight into the evaluation when the sample sizes of the classes are skewed, which is the case in vessel segmentation, where the number of non-vessel pixels is higher than the number of vessel pixels [124]. These quantitative measures share the disadvantage of relying on the threshold used to generate the segmentations from the probability maps. Additionally, the area under the Receiver Operating Characteristic curve (ROC cuve) - the AUC - can be reported. The ROC curve plots the fraction of pixels correctly classified as vessel, namely the true positive rate (TPR), versus the fraction of background pixels wrongly classified as vessel, the false positive rate (FPR), as the threshold value is varied. The AUC, as the name implies, is the area below this curve, so it does not depend on the threshold used to obtain each sensitivity/specificity pair, encompassing all possible thresholds. The closer the ROC curve approaches the top left corner, the better is the performance of the system and higher the AUC value [125].

5.2.5 Software and Hardware

The functions used throughout the various steps of this work were developed in Python [126]. More specifically, the proposed methods were implemented using Keras[1] with TensorFlow[2] backend.

The tests were performed on a computer with the Linux Mint 18 operating system, equipped with an NVIDIA GeForce GTX 1070 GPU, an Intel® CoreTM i7-6850k 3.6 GHz processor and 128 GB of RAM.

5.3 Results and Discussion

In this section, the start by validating some hyperparameters for SWA setting. For this purpose, the DRIVE database was used, and Acc and AUC metrics were used as criteria for selecting the best models as they concise network performance. Then, we compare the performance of SWA facing fast-SWA. Finally we apply our systems to the other databases, namely, to STARE and CHASE_DB1, and we analyze the obtained results in the light of the state of the art. Only the best systems obtained for SWA and fast-SWA were tested on STARE and CHASE_DB1.

[1] Keras: The python deep learning library in https://keras.io/

[2] https://www.tensorflow.org/.

5.3.1 SWA Learning Rate

The first study conducted to tune SWA for retinal vessel segmentation aims to understand the impact of the learning rate range on SWA. The hyperparameters of the learning rate scheduler are key to SWA as they affect the exploration of the weight space, which is more important than the accuracy of individual networks collected.

As referred before, the objective of SWA is to explore the local minima region to achieve a better solution in the same basin of attraction, not to find a different minima of the initial model. Hence, on one hand, it is instinctive to lower the maximum learning rate to assure the training loss does not escape to a different local minima. On the other hand, the use of larger learning rate values enable to find multiple faraway points in weight space. In fact, when the distance between the averaged points is large the contribute of averaging is the most prominent [119].

The experimental results of using different learning rate ranges in SWA are shown in Table 1. First, the maximum and minimum learning rate were fixed in 1×10^{-3} and 1×10^{-5}. Then, fixing the minimum learning rate at 1×10^{-5}, we lowered the maximum learning rate to 1×10^{-4}. Analyzing Table 1, when the maximum learning rate decreased the Sens droped but the Spec, MCC and AUC increased. Regarding Acc, the tests achieved equal value. The optimum point was achieve when the maximum learning rate was set to 1×10^{-4} and the minimum to 1×10^{-5}. This learning rate range achieved the highest Acc, MCC and AUC values. Hence, we set the minimum learning rate to 1×10^{-5} and the maximum was reduced to 1×10^{-4} and kept constant across cycles in all SWA and fast-SWA experiments.

Table 1: Segmentation performance of different learning rate ranges in SWA on DRIVE test set. The metrics mean and standard deviation are presented, the later between parenthesis. Bold values show the best scores.

Model	Learning Rate		Acc	Sens	Spec	MCC	AUC
	maximum	minimum					
Baseline	1×10^{-3}	1×10^{-5}	0.9569 (0.0043)	0.7931 (0.0553)	**0.9811 (0.0055)**	0.8010 (0.0192)	0.9800 (0.0056)
SWA	1×10^{-3}	1×10^{-5}	**0.9570 (0.0041)**	**0.8022 (0.0535)**	0.9798 (0.0057)	0.8027 (0.0181)	0.9800 (0.0056)
(16 epochs)	1×10^{-4}	1×10^{-5}	**0.9570 (0.0041)**	0.8013 (0.0530)	0.9801 (0.0056)	**0.8029 (0.0180)**	**0.9802 (0.0055)**

5.3.2 SWA duration

The second study conducted sought to adjust the SWA duration. It is easy to speculate that model performance and SWA duration are directly related, since duration varies the number of points averaged and these points and their location change the position of the final model on the loss surface. So, in theory, the averaged points will cover more loss surface area as the duration of the SWA increases, and the final model will be more centered in the local minima, leveraging overall performance.

The effects of varying the SWA duration, between 8 and 24 epochs, are shown in Table 2. All tests were performed under the same conditions, with the only source of variability being the training duration. Analyzing Table 2, there is no seeming relationship between SWA duration and performance variation. Overall, the metrics are quite similar. As a matter of fact, the results do not change as the SWA duration varies, in both Acc and AUC mean values. This shows that the algorithm is robust to SWA duration. The best performance is obtained when running SWA for 16 epochs. It achieves the highest global Spec and MCC performance and equals the other tests in terms of Acc and AUC.

Table 2: Segmentation performance of different SWA duration on DRIVE test set. The metrics mean and standard deviation are presented, the later between parenthesis. Bold values show the best scores.

SWA duration	Acc	Sens	Spec	MCC	AUC
24 epochs	**0.9570 (0.0041)**	**0.8017 (0.0528)**	0.9799 (0.0056)	0.8026 (0.0179)	**0.9802 (0.0055)**
20 epochs	**0.9570 (0.0041)**	0.8015 (0.0529)	0.9800 (0.0056)	0.8027 (0.0180)	**0.9802 (0.0056)**
16 epochs	**0.9570 (0.0041)**	0.8013 (0.0530)	**0.9801 (0.0056)**	**0.8029 (0.0180)**	**0.9802 (0.0055)**
12 epochs	**0.9570 (0.0042)**	0.8010 (0.0531)	0.9800 (0.0056)	0.8025 (0.0182)	**0.9802 (0.0055)**
8 epochs	**0.9570 (0.0042)**	0.8011 (0.0531)	0.9800 (0.0056)	0.8025 (0.0182)	**0.9802 (0.0055)**

5.3.3 SWA and Fast-SWA

After validating some SWA hyperparameters, we ran fast-SWA for a maximum of 16 epochs. As can be seen in Table 3, both ensemble methods, SWA and fast-SWA, were able to improve the performance of the baseline network, increasing all metrics but Spec in all experiments. This suggests that fast-SWA, similarly to SWA, successfully explores the minima region of the baseline model. In fact, while collecting model weights for averaging corresponding to higher learning rates, fast-SWA continues to successfully improve the base system, achieving the highest Acc and AUC.

Comparing SWA and fast-SWA results, we verify that, for our architecture, by updating the average multiple times during a single cycle, fast-SWA converges substantially faster to a minimum than the original SWA. In fact, the overall performance of fast-SWA at epoch 8 is quite similar to SWA at epoch 16, having the same Acc, Spec and AUC, but falling short in terms of Sens and MCC, which indicates that SWA was capable of detecting more vessel pixels. The best ensemble variant is achieved with fast-SWA procedure, trained for 16 epochs, as it gathers the highest Acc and AUC.

Table 3: Segmentation results of each SWA and fast-SWA test on DRIVE. The metrics mean and standard deviation are presented, the later between parenthesis. Bold values show the best scores.

Method		Acc	Sens	Spec	MCC	AUC
Baseline		0.9569 (0.0043)	0.7931 (0.0553)	**0.9811 (0.0055)**	0.8010 (0.0192)	0.9800 (0.0056)
SE		0.9570 (0.0043)	0.7971 (0.0542)	0.9807 (0.0054)	0.8019 (0.0189)	0.9802 (0.0056)
SWA	(16 epochs)	0.9570 (0.0041)	**0.8013 (0.0530)**	0.9801 (0.0056)	**0.8029 (0.0180)**	0.9802 (0.0055)
fast-SWA	(16 epochs)	**0.9571 (0.0043)**	0.7973 (0.0536)	0.9807 (0.0055)	0.8024 (0.0187)	**0.9803 (0.0055)**
	(8 epochs)	0.9570 (0.0042)	0.8008 (0.0529)	0.9801 (0.0056)	0.8028 (0.0181)	0.9802 (0.0055)

The Acc, Sens, Spec and AUC values of the best case for the DRIVE database were 0.9679, 0.9093, 0.9759 and 0.9901, while those of the worst case matched 0.9488, 0.7538, 0.9820, and 0.9764, respectively. From Figure 22, where we compare the best and worst case obtained for DRIVE, it is noteworthy that there is a predominance of thin vessels in the case with the lowest results (Fig. 22 - second row) and that the our system has failed in detecting these, which may have caused the low Sens value due to the

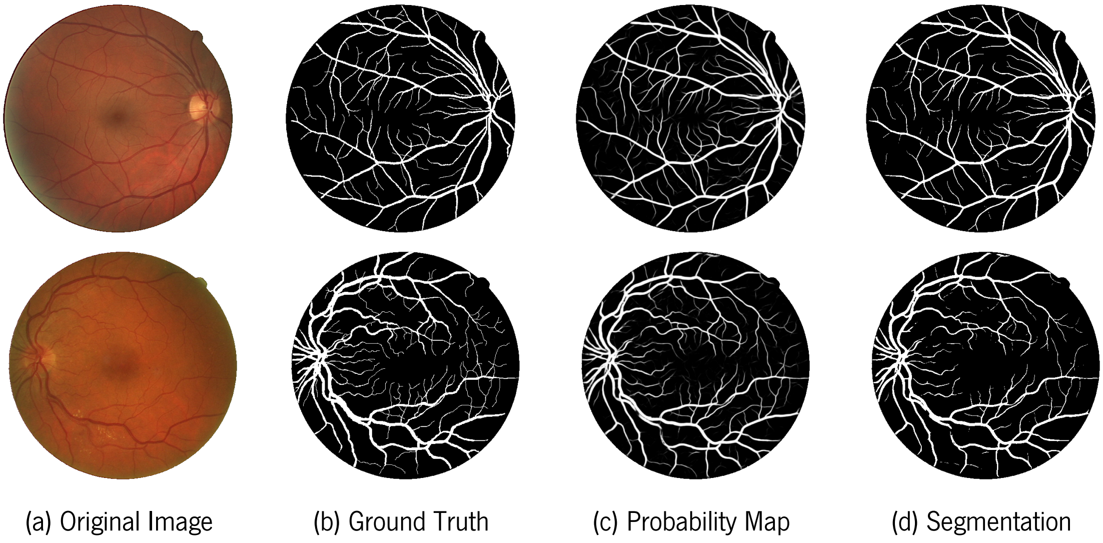

(a) Original Image (b) Ground Truth (c) Probability Map (d) Segmentation

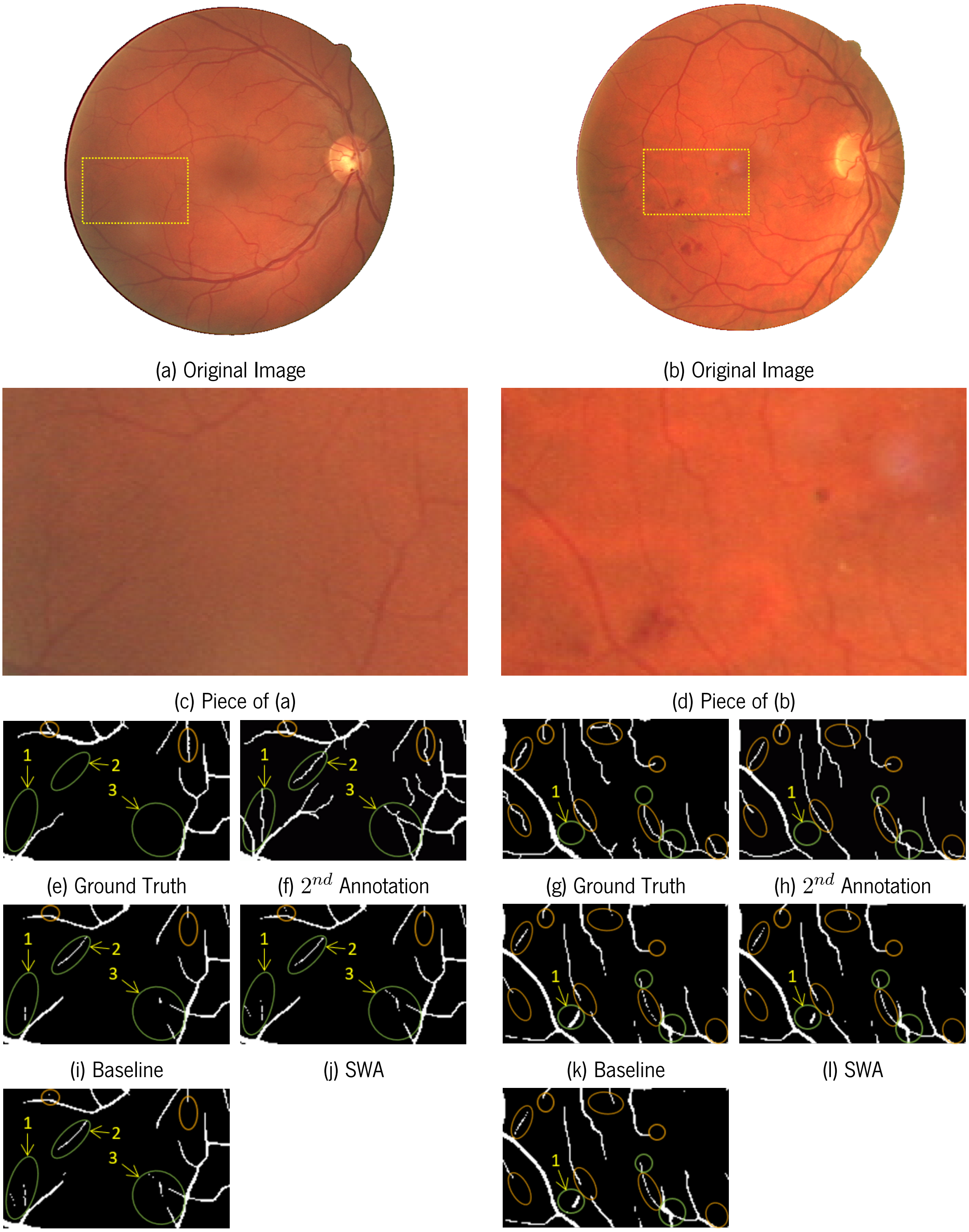
Figure 22: Segmentation examples for DRIVE database for the final system, Baseline + fast-SWA. The first row shows the best case, while the second presents the worst one.

decrease in TP, and may have negatively affected the remaining metrics. In parallel, we notice that the probability map correctly marks some of these small vessels that do not appear in the final segmentation, showing that our strategy could benefit from some tuning of the threshold.

Observing Figure 23(left), it is possible to note that the vessels annotated only by the 2^{nd} human observer (Figure 23(left) - arrows 1, 2 and 3) are barely visible due to their characteristics: narrow width and low contrast. Moreover only the annotations from the 1^{st} observer were used to train the models. However these vessels are somewhat detected by the baseline model. Furthermore, the SWA and fast-SWA procedures were capable of improving this detection. This indicates that some of the FP, responsible for the decrease of Spec, are actually real vessels. Analysing Figure 23(right), it is perceivable that the hemorrhage present in the image, marked with the arrow 1, is partially segmented as vessel by the baseline model but SWA and fast-SWA lessen its effect, decreasing the FP. Also, it can be noted that the SWA and fast-SWA models, when compared with the model model, detect more thin vessels. To sum up, SWA and fast-SWA have shown to improve detection of vessels, especially of thin vessels, and to minimize false detections of the baseline model.

5.3.4 Snapshot Ensembling and Stochastic Weight Averaging

Additionally, we evaluated the ensembling approach proposed by Huang et al. [50] in our baseline model to compare its performance with the improvement of SWA and fast-SWA over the baseline system. We evaluate this alternative using the strategy proposed by Huang, collecting all the models corresponding to the lowest learning rate value of each training cycle of the baseline model. Analysing Table 3, SE improved the baseline perfomance. In fact, SE achieved similar AUC, and better Acc and Spec than SWA. However, SE stood below fast-SWA in terms of Acc, Sens, MCC and AUC, and equaled fast-SWA in terms of Spec. Thus, the overall improvement of SE is smaller than the improvement made by fast-SWA. Furthermore, in SE it is mandatory to store and make predictions for several networks prior to averaging for the final prediction, while the stochastic weight averaging methods are more straightforward, it is only required to store one model and to compute predictions for it.



(a) Original Image

(b) Original Image

(c) Piece of (a)

(d) Piece of (b)

(e) Ground Truth

(f) 2^{nd} Annotation

(g) Ground Truth

(h) 2^{nd} Annotation

(i) Baseline

(j) SWA

(k) Baseline

(l) SWA

(m) Fast-SWA

(n) Fast-SWA

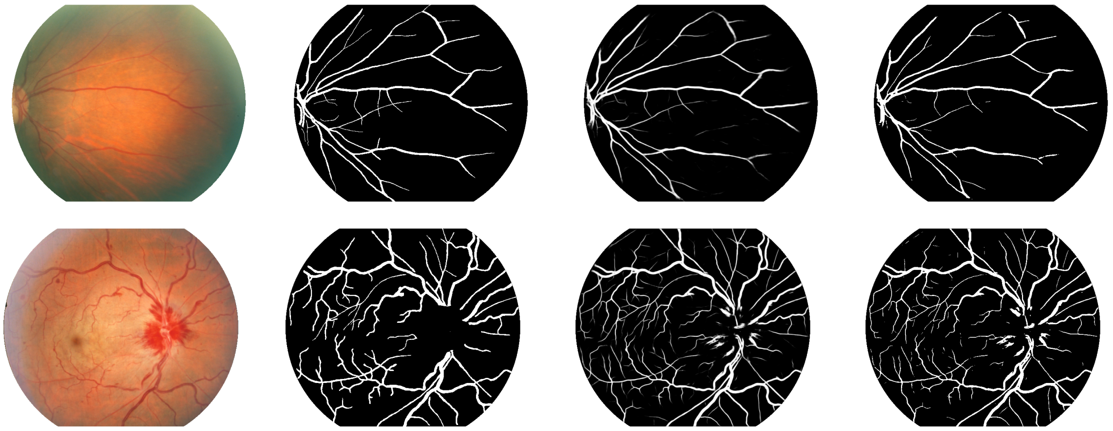
Figure 23: Pieces of segmentation results from DRIVE test images: image 20 (left) and image 17 (right). The presence of FP and FN, in relation to the ground truth, is marked in green and orange, respectively.

5.3.5 Results on STARE and CHASE_DB1

The results for each database are shown in Table 4. It should be noted that not only the final system was tested on CHASE_DB1 and STARE, but also the baseline model, which allowed to validate the improvement.

Analyzing the results of fast-SWA for each database, we notice that the fast-SWA procedure improved the performance of the baseline system, in terms of Acc, Sens and AUC across all databases. In STARE the value of Spec also increased. Additionally, we notice that fast-SWA improved Sens, in DRIVE and CHASE_DB1, respectively, by 0.42% and 1.31%, while the Spec suffered a minor decrease of 0.04% and 0.08%, which indicates that fast-SWA was able to increase the detection of vessel pixels without substantially increasing the number of FP. Fundus images have around 88% non-vessel pixels and only 12% vessel pixels so, this observation is noteworthy because the unbalanced class ratios of this segmentation task hampers the training of a good classifier.

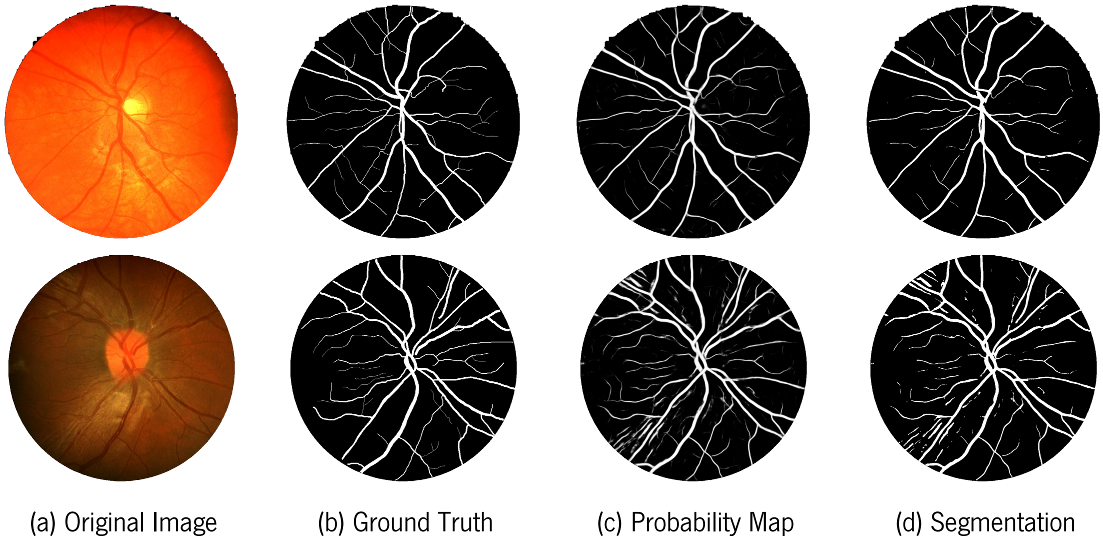
For the STARE database, the Acc, Sens, Spec, and AUC values of the best case were 0.9793, 0.7863, 0.9940 and 0.9870, while those of the worst case matched 0.9470, 0.7249, 0.9788 and 0.9760 (Fig. 24). Examining the bottom row of Figure 24, which shows our worst case for STARE database, it is noticeable that small hemorrhagic blobs in the optic disc area were not rejected, which can impede a more noticeable increase in the method performance. This may be caused by our choice of using 5-fold cross-validation for training, as the number of images with pathological signs was reduced.



(a) Original Image (b) Ground Truth (c) Probability Map (d) Segmentation

Figure 24: Segmentation examples for STARE database for the final system, Baseline + fast-SWA. The first row shows the best case, while the second presents the worst one.

Finally, for the CHASE_DB1 database, the values of the best case were 0.9738, 0.8642, 0.9829 and 0.9896, while those of the worst case matched 0.9520, 0.8465, 0.9633 and 0.9789. Analysing the original image of the worst case (Fig. 25 - second row), it is noted that most of the parts of retina wrongly segmented as vessels have vessel-like characteristics. More specifically, the lighter parts of retina are structured in such a way that darker line segments are formed between them. The segmentation of these parts may be induced by the lack of similar examples in the training set caused by our choice of using 4-fold cross-validation for training.



(a) Original Image (b) Ground Truth (c) Probability Map (d) Segmentation

Figure 25: Segmentation examples for CHASE_DB1 database for the final system, Baseline + fast-SWA. The first row shows the best case, while the second presents the worst one.

5.3.6 Comparison with the state-of-the-art

In Table 4, the proposed method is compared with state-of-art algorithms on the three databases.

Table 4: Segmentation results of different approaches on DRIVE, STARE and CHASE_DB1. Values in bold show the best score among all methods. Ens stands for ensemble methods.

	Methods	Year	DRIVE				STARE				CHASE_DB1			
			Acc	Sens	Spec	AUC	Acc	Sens	Spec	AUC	Acc	Senc	Spec	AUC
Unsupervised	Roychowdhury et al. [127]	2015	0.9494	0.7395	0.9782	0.9672	0.9560	0.7317	0.9842	0.9673	0.9467	0.7615	0.9575	0.9623
	Zhang et al. [128]	2016	0.9476	0.7743	0.9725	0.9636	0.9554	0.7791	0.9758	0.9748	0.9452	0.7626	0.9661	0.9606
	Badawi and Fraz [129]	2018	0.9547	0.7898	0.9709	-	0.9530	**0.8650**	0.9609	-	0.9529	**0.8004**	0.9643	-
	Aguirre-Ramos et al. [130]	2018	0.9503	0.7854	0.9662	-	0.9231	0.7116	0.9454	-	-	-	-	-
Supervised	Li et al. [131]	2016	0.9527	0.7569	0.9816	0.9738	0.9628	0.7726	0.9844	0.9879	0.9581	0.7507	0.9793	0.9716
	Liskowski and Krawiec [132]	2016	0.9515	0.7520	0.9806	0.9710	**0.9696**	0.8145	0.9866	0.9880	-	-	-	-
	Dasgupta and Singh [133]	2017	0.9533	0.7691	0.9801	0.9744	-	-	-	-	-	-	-	-
	Zhang et al. [134]	2017	0.9466	0.7861	0.9712	0.9703	0.9547	0.7882	0.9729	0.9740	0.9502	0.7644	0.9716	0.9706
	Mo and Zhang [135]	2017	0.9521	0.7779	0.9780	0.9782	0.9674	0.8147	0.9844	0.9885	0.9599	0.7661	0.9816	0.9812
	Yan et al. [136]	2018	0.9542	0.7653	0.9818	0.9752	0.9612	0.7581	0.9846	0.9801	0.9610	0.7633	0.9809	0.9781
	Hu et al. [137]	2018	0.9533	0.7772	0.9793	0.9759	0.9632	0.7543	0.9814	0.9751	-	-	-	-
	Oliveira et al. [138]	2018	**0.9576**	**0.8039**	0.9804	**0.9821**	0.9694	0.8315	0.9858	**0.9905**	**0.9653**	0.7779	0.9864	**0.9855**
	Wang et al. [139]	2019	0.9541	0.7648	0.9817	-	0.9640	0.7523	**0.9885**	-	0.9603	0.7730	0.9792	-
	Guo et al. [140]	2019	0.9561	0.7891	0.9804	0.9806	0.9674	0.8212	0.9843	0.9859	0.9627	0.7888	0.9801	0.9840
Ens.	Fraz et al. [141]	2012	0.9480	0.7406	0.9807	0.9747	0.9534	0.7548	0.9763	0.9768	-	-	-	-
	Barkana et al. [142]	2017	0.9502	0.7224	**0.9840**	0.9532	0.9553	0.7014	0.9846	0.9430	-	-	-	-
	Baseline	2019	0.9569	0.7931	0.9811	0.9800	0.9660	0.8145	0.9838	0.9853	0.9635	0.7758	0.9850	0.9825
	Baseline + fast-SWA	2019	0.9571	0.7973	0.9807	0.9803	0.9676	0.8185	0.9850	0.9862	0.9643	0.7897	0.9842	0.9835

Comparing the state-of-the-art methods with our results, it is possible to infere that our baseline network is competitive. In fact, our baseline, a simple encoder-decoder FCN, obtained the highest and the second highest Spec for two of the three databases evaluated, CHASE_DB1 and DRIVE, respectively. Also, it stood in third in terms of Acc, on CHASE_DB1 and DRIVE. This is relevant in the sense that it is hard to improve over a competitive model, which sustains the added value of the SWA approach for retinal vessel semantic segmentation.

In relation to DRIVE, the proposed method achieved the second best score on Acc and Sens, being only surpassed by Oliveira et al. [138]. In their work, Oliveira et al. [138] used Stationary Wavelet Transform to create new FCN input channels and proposed data augmentation for prediction. When excluding the effect of test time data augmentation and the use of wavelets as input, our method obtains better Acc. In relation to AUC, we stood in third, following Oliveira et al. [138] and Guo et al. [140]. The latter employs a FCN with short connections, which pass low level information to high level layers and high level information to low level layers to exploit the semantic and structural information provided by low and high level layers, respectively [140]. In terms of Spec, our method obtained the seventh best result. All works with superior Spec, had a much lower value of Sens, which means that the detection of more background compromised the detection of vessels.

In STARE database, we stood in third for Acc, in fourth for Sens and Spec and in fifth for AUC. By comparing our work with the higher performing methods in STARE, Li et al. [131], Liskowski and Krawiec [132], Wang et al. [139] and Mo and Zhang [135] used leave-one-out cross-validation and Badawi and Fraz [129] and [140] splitted the data in 2 subsets: one for evaluation and one for testing, contrary to Oliveira et al. [138] that used 5-fold cross-validation for training, similarly to us, what can explain the results.

Analyzing the results on CHASE_DB1, it can be seen that the values of Acc and Spec are the second highest, being outperformed by Oliveira et al. [138]. In terms of Sens, we also stood in second being surpassed by Badawi and Fraz [129]. Regarding AUC, our method obtained the third best result, being surpassed by Oliveira et al. [138] and Guo et al. [140]. In Badawi and Fraz [129] and Guo et al. [140], the approach was only tested on a subset of the data, while we tested on the complete database, by applying 4-fold cross-validation.

Focusing in the ensemble methods for retinal vessel segmentation, proposed by Fraz et al. [141] and Barkana et al. [142], we verify that our approach presented a much higher Acc, Sens and AUC across DRIVE and STARE databases. In terms of Spec, we ranked second in DRIVE, with the work of [142] leading, and first in STARE. Nonetheless, knowing that Acc integrates information from Sens and Spec we may conclude that we present better balance between Sens and Spec. Notwithstanding, these ensembles were composed with different strategies from ours. Fraz et al. [141] used decision trees and a boosting algorithm to combine the predictions, and Barkana et al. [142] ensembled three different classification methods combining their predictions by majority voting. Also, notice that none of the other ensemble methods were evaluated on CHASE_DB1.

5.4 Summary

This chapter presents the methods, structure and results of the ensemble algorithm proposed for the retinal vessel segmentation problem. We adapted a recent ensemble approach to retinal vessel segmentation. Although there are several well-established ensemble methods, most of them involve storing and prediction for multiple models before combining them to get the final prediction. Stochastic weight averaging improves model performance with a single network by finding a more central solution in a region of minimum points of the weight space.

Our FCN-based segmentation systems includes pre-processing, data augmentation, and classification steps. The main goal of pre-processing is to standardize the image characteristics and, if possible, to accentuate the vessels so that the segmentation task can be made easier. Regarding data augmentation, one of the greatest challenges in the medical imaging field is how to cope with the small number of data and the limited quantity of labeled data [143]. One way to address this problem is data augmentation in the training set. Effectively, when trained in more data, deep learning models generally generalize better because the leaning for overfitting is reduced [44]. Classification is made through a encoder-decoder FCN, following the line of thought of an algorithm based on supervised classification.

Having tuned the FCN-based ensemble approach, we show that the use of weight averaging techniques was beneficial for retinal vessel segmentation.

Chapter 6

Attention Mechanisms with Adaptive-scale Context for Brain Tumor Segmentation

In this chapter, we present the methodology implemented and the variants studied in the development of an attention mechanism adapted for automatic brain tumor segmentation. The structure of the attention mechanism is presented and described. Then, the steps of the proposed system are briefly explained. Afterwards, we describe the database used to perform the tests, the evaluation metrics used, and the training settings. Finally, the results obtained for our study of adaptive-scale context attention mechanisms are discussed, and the best models are compared with other state-of-the-art methods.

6.1 Methodology

This section presents and describes the structure of the implemented attention mechanism and the variants studied in the development of an attention mechanism adapted for automatic brain tumor segmentation.

6.1.1 Attention Block

Throughout this dissertation, we propose an adaptive attention block that comprises feature map recalibration and adaptive context. Related with our work is the SegSE block proposed by Pereira et al. [73]. The SegSE block learns spatial and channel recalibration jointly, functioning as an intra-channel attention mechanism. Thus, our work directly extends the work of Pereira et al. [73] and adapts it by integrating a strategy in which the scale is variable and the context is adaptively selected.

We propose and attention block that captures multi-scale information by processing n parallel convolutional layers with different receptive fields. It adaptively chooses the optimal scale to focus on in a data-driven learned manner. In particular, when processing different spatial locations, our attention module uses an attention mechanism to establish the importance of each scale. To determine how much focus to put on each branch, a softmax function normalizes the n parallel branches producing n attention weights. Then, the outputs from the parallel layers are fused with a softmax weighted summation. The final output is computed by an element-wise multiplication of the feature maps with the input. Figure 26 summarizes the attention block. Each A_i with $i \in \{1, ..., n\}$ can take the form of one of the branches variants in Fig. 27. The different branches have distinct approaches of capturing contextual information.

Next, we start by describing the different branches used in the attention block.

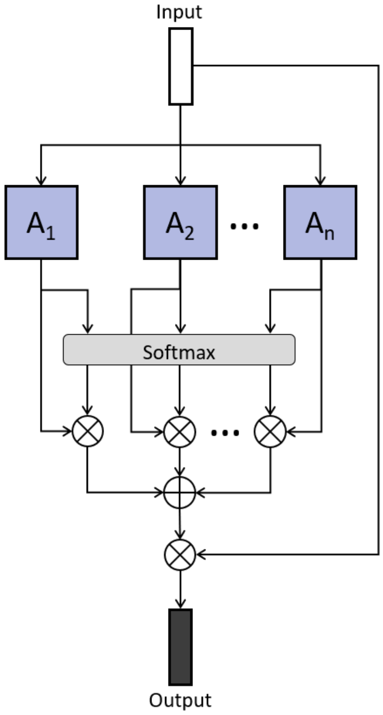


Figure 26: Attention block with n branches.

Motivation Unequivocal classification of different objects in an image is likely to require distinct combinations of local and global information. For example, small structures may require focusing on high-resolution local information, while large objects may be better segmented by processing a large receptive field at the expense of fine details. Consequently, architectures that define multi-scale processing enable more complex features but may be suboptimal.

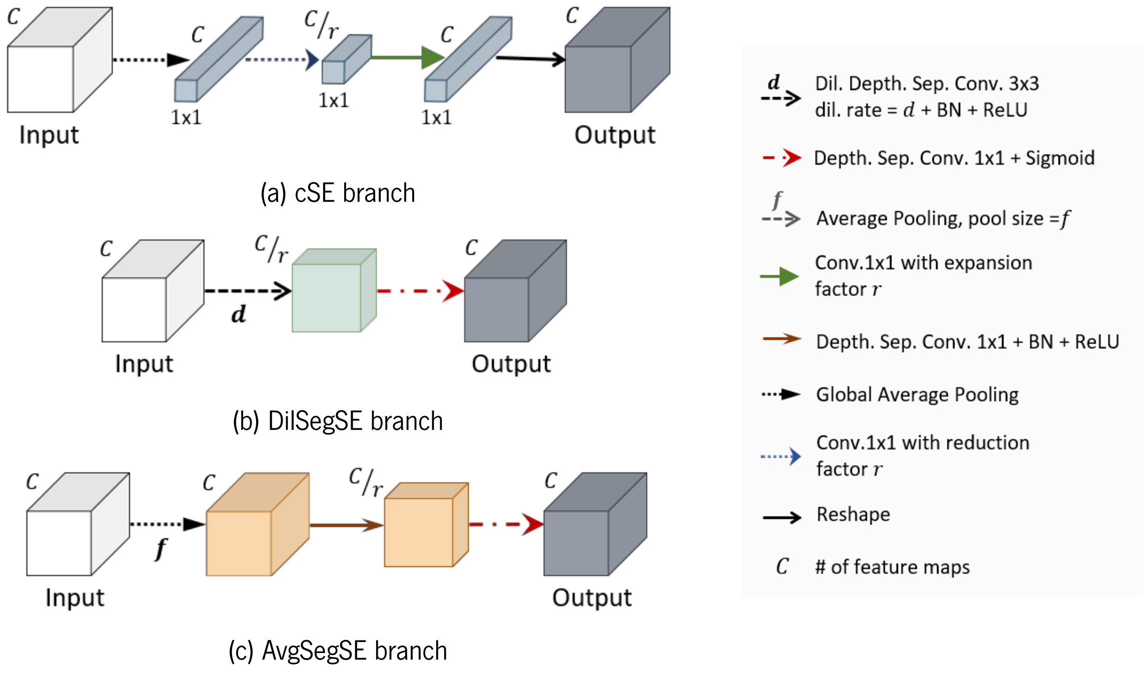
In the context of medical image segmentation, the complex nature of medical images alone makes the task difficult. Moreover, the structures are often non-rigid, vary in location, size and shape, and have indistinct boundaries. For all these reasons, the development of strategies with variable scale and adaptive context is desirable as an adaptive scheme may provide more informative features.

6.1.1.1 Channel Recalibration Branch

Recalibration consists in suppressing the less relevant features of a layer by learning the relationship among the feature maps. To do so, feature maps are linearly reduced to lower dimensions and then restored to the original number of feature maps. Hu et al. [66] first proposed to recalibrate whole feature maps, which we refer to as spatial squeeze and channel excitation block (cSE block). The cSE block performs feature recalibration by explicitly modeling the dependencies across the feature map channels.

It learns to selectively emphasize discriminative features and suppress less useful ones by using global information.

We adopt the cSE block integrated in FCNs for semantic segmentation by Roy et al. [72]. First, global average pooling (GAP) executes the spatial squeeze operation, summarizing each feature map into its average value to embed contextual information, resulting in a channel descriptor. Then, two 1×1 convolutions capture cross-channel relations. The first layer is a compression layer with a reduction factor r, which specifies the bottleneck that encodes the channel-wise dependencies, followed by the ReLU activation function. The succeeding convolutional layer restores the previous dimension and it is followed by the sigmoid activation (Fig. 27 - cSE branch).


C
Input
C
1x1
C/r
1x1
C
1x1
C
Output
C
Input
C/r
d
C
Output
C
Input
f
C
C/r
C
Output
d
Dil. Depth. Sep. Conv. 3x3
dil. rate = d + BN + ReLU
Depth. Sep. Conv. 1x1 + Sigmoid
f
Average Pooling, pool size = f
Conv.1x1 with expansion factor r
Depth. Sep. Conv. 1x1 + BN + ReLU
Global Average Pooling
Conv.1x1 with reduction factor r
Reshape
C
# of feature maps

(a) cSE branch

(b) DilSegSE branch

(c) AvgSegSE branch

Figure 27: Attention branch variants.

6.1.1.2 SegSE Branches

Pereira et al. [73] proposed an alternative approach to the cSE block (described in Section 6.1.1.1), the Segmentation adapted Squeeze-and-Excitation block (SegSE block). The SegSE block emerged from the observation that, in semantic segmentation tasks, contextual information is fundamental to evaluate the spatial importance of a given unit.

The SegSE block performs spatially adaptive recalibration collecting contextual information while maintaining the spatial meaning and still considering the cross-channel relationships. First, the squeeze operation aggregates contextual information through a convolutional layer with dilation, followed by batch normalization and the ReLU activation function. The spatial information is captured by the dilated kernel operating over neighboring voxels. Simultaneously, this layer is responsible for the bottleneck by reducing the number of feature maps. The dilation rate depends of the resolution of the feature maps and is decreased from 3 to 2 and 1 as the depth of the network increases. Then, a 1×1 convolution is applied to the feature maps, followed by the sigmoid activation. Last, the feature maps are element-wise multiplied with the input resulting in recalibrated feature maps. Thus, this block learns spatial and channel recalibration jointly, unlike [72] where they are considered separately.

This approach has shown to improve learning in image semantic segmentation. However, regular convolutions have a greater ability to capture more information while requiring more data for training in order not to overfit. Additionally, in depthwise separable convolutions, the depthwise convolution have the interesting property of having only one kernel per channel, which seems to be more natural when attempting to summarize the feature maps spatial information, rather than merge information from all channels, as happens in regular convolution operations. Thus, the proposed attention block includes depthwise separable convolutions .

We modify the SegSE block to our adaptive attention block and we experiment with different approaches for aggregating context, namely, dilated convolution and average pooling.

SegSE Branch with Dilated Convolutions The proposed branch can be observed in Fig. 27 - DilSegSE branch. We employ depthwise separable convolutional layers with dilated kernels to capture context. The branches with dilations follow the structure of the SegSE block. First, a depthwise separable convolutional layer with dilation aggregates context, right after batch normalization and ReLU. This layer is simultaneously responsible for the squeeze operation reducing the number of channels by a factor of r. Subsequently, a 1×1 depthwise separable convolution is applied to the feature maps, restoring the number of feature maps, followed by the sigmoid activation. The kernel size and dilation rate of the first convolutional layer was varied across the experiments.

SegSE Branch with Average Pooling The design of this branch follows the reasoning of the previous one. Contextual information may be also obtained by average pooling operations. These act over a kernel-based region instead of the entire channel of the feature maps, contrasting with global pooling. So, firstly, an average pooling layer with filter size f is employed. Then, a 1×1 depthwise separable convolutional layer combines the feature maps and learns their relationship, followed by batch normalization and dropout. Once again, this convolution reduces the number of channels by a factor of r, being accountable for the compression stage. The ensuing convolutional layer restores the previous number of feature maps. It consists of a 1×1 depthwise separable convolutional layer followed by a sigmoid function. This branch is depicted in Fig. 27 - AvgSegSE branch. The average pooling kernel size of the first convolutional layer was varied across the experiments.

6.2 Experimental Details

In this section, we describe the steps of our automatic segmentation strategy, the databases, evaluation metrics and training setting used to evaluate the attention mechanisms deceloped for brain tumor segmentation. Finally, we describe the software and hardware used.

6.2.1 Brain Tumor Segmentation Setup

The various steps of the proposed automatic brain tumor segmentation system are illustrated in Figure 28. These steps can be encompassed in 3 phases: pre-processing, classification via FCN and post-processing. Next, the implementation of each step will be described.

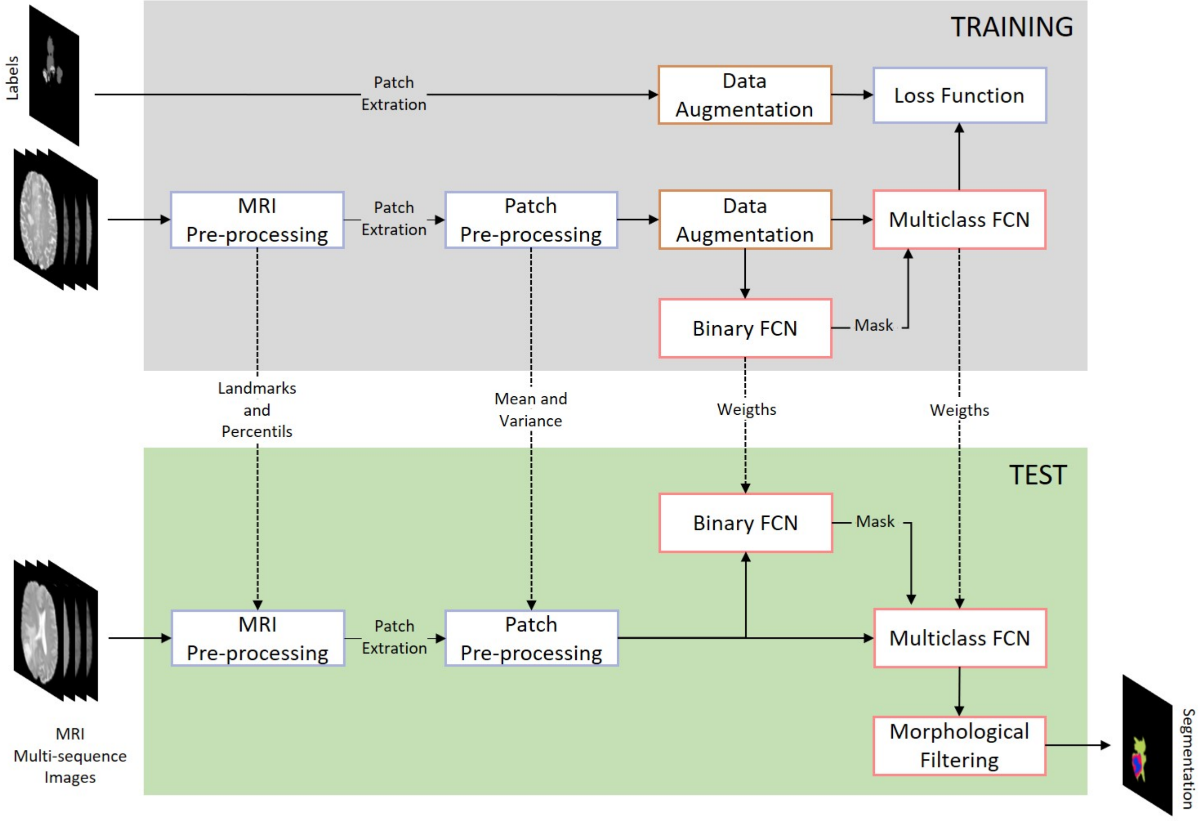


Figure 28: Overview of the proposed automatic brain tumor segmentation system.

6.2.1.1 Pre-processing

Image pre-processing included bias field correction, and standardization of the intensity histograms of each MRI sequence, as in Pereira et al. [144]. Bias field distortion can affect MRI images causing the intensity of the same tissues to change across the image. To deal with this, the correction of the inhomogeneity of intensities is based on the N4IT method [145]. However, this does not ensure similar intensity values for the same MRI sequence but at different moments or for different patients, which is an assumption in most segmentation systems [41, 42]. The intensity normalization method proposed by Nyúl et al. [41] seek, using nonlinear transformations, to manipulate the histogram of a given image into a standard histogram while maintaining the existing relationship between the intensities of the different tissues. To make the contrast and intensity ranges more similar across patients and acquisitions, in this method a set of intensity landmarks $I_L = \{pc_1, i_{p10}, i_{p20}, ..., i_{p90}, pc_2\}$ are learned for each sequence from the training set. i_{pl} represents the intensity at the l^{th} percentile and, pc_1 and pc_2 are chosen as described in [41], for each MRI sequence. Subsequent to the images normalization, each patch was normalized to have zero mean and unit standard deviation by computing the mean intensity value and standard deviation across all training patches extracted for each sequence. These statistics are then used to normalize the testing patches.

6.2.1.2 Data Augmentation

Data augmentation was performed through two processes. One consists on introducing random rotations of multiples of $90°$ ($90°$, $180°$, $270°$). The other comprised of sagittal flipping.

6.2.1.3 Post-Processing

The presence of a brain tumor, whether it is HGG or LGG, consists of a substantial volume structure, that is, a large connected component. Some small clusters may be incorrectly classified as tumor. Additionally, although rare, patients may have more than one region of brain tumor. To address these problems, we impose volumetric constraints by applying morphological filtering to remove clusters that are smaller than a predefined threshold, τ_{VOL}, in the segmentation obtained by the FCN.

6.2.1.4 Hierarchical Segmentation

In brain tumor segmentation most of the pixels in the brain structure consist of healthy tissue. To handle the data imbalance problem we follow a hierarchical FCN-based brain tumor segmentation approach as reported by Pereira et al. [146]. We start by segmenting the whole tumor as a binary segmentation problem with a binary FCN – the WT-FCN. Afterward, the resulting binary segmentation is used to define a cubic region of interest (ROI) around the tumor with a margin of 10 extra voxels on each side. Then, a multi-class FCN segments the multiple tumor structures inside the ROI.

6.2.1.5 Baseline Architectures

The proposed FCNs are inspired by an encoder-decoder architecture with long skip connections [71]. The input for the FCNs are image patches extracted from all the available MRI sequences. The architecture of WT-FCN is depicted in Fig. 29. It is an 3D FCN with a large field of view, achieved by three pooling layers, which is important to reduce the false positive detections. The WT-FCN is kept constant across all experiments to isolate and enable to evaluate and compare the benefits introduced by the attention block in the multi-class FCN.

The baseline multi-class FCN architecture can be discerned from Fig. 30, by disregarding the attention blocks. We design the multi-class FCN as a 2D network, as a proof of concept to evaluate the projected components, following the work of Pereira et al. [146]. Thus, 2D image patches are extracted in the axial plane, which reduces the computational cost compared to the WT-FCN.

The multi-class network consists of 6 groups of 2 convolutional layers with 3×3 kernels, followed by ReLU as activation, batch normalization and dropout (Fig. 30 - B1 block). After each group, in the contracting path, 2×2 max pooling is applied to generate higher-level feature maps and the number of feature maps is doubled, whereas, in the expanding path, the feature maps are upsampled and their number is halved. Then, the lower level feature maps are added to higher-level ones, through long skip

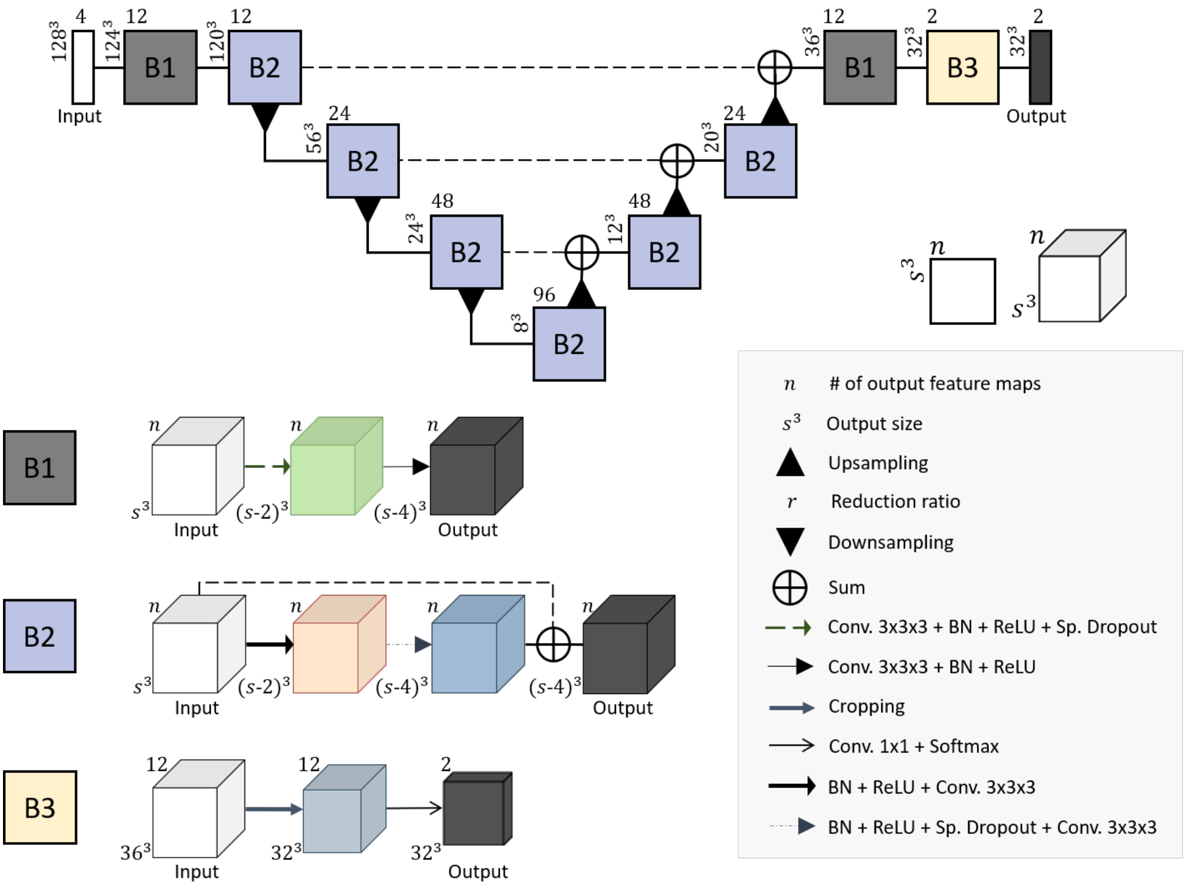


Figure 29: Architecture of the WT-FCN. We use $1 \times 1 \times 1$ convolutional layers to adjust the number of feature maps, before addition. BN stands for batch normalization, and Sp. Dropout for spatial dropout.

connections. Since we used convolution without padding, during the sum of feature maps with different sizes, we cropped the center part of the biggest one to fit the smaller. This encoder-decoder architecture enables the network to capture both context and precise location. The pooling operations in the encoder increase the receptive field while reducing the spatial information of the features, creating low-resolution representations highly suitable for recognizing objects. In the decoder the features are gradually upsampled back to the input patch resolution and combined with higher-resolution features of the encoder path. The further convolutional layers fuse the lower and higher-level features. At the final layer, a 1×1 convolution and softmax is used to map each feature vector to the number of classes and compute prediction (Fig. 30 - B2 block).

6.2.2 Database

During the course of this project, the proposed method was validated on brain tumor segmentation. To do so, the method was tested with the publicly available BRATS 2017 [8, 147] database that utilizes MRI scans and focuses on the segmentation of gliomas.

The MRI images of brain tumors were made available as a part of the Multimodal Brain Tumor Segmentation Challenge 2017: MICCAI BraTS 2017 [8, 147], which provides a large dataset of annotated LGG and HGG. The BraTS 2017 dataset comprises two sets: Training with 210 HGG and 75 LGG cases

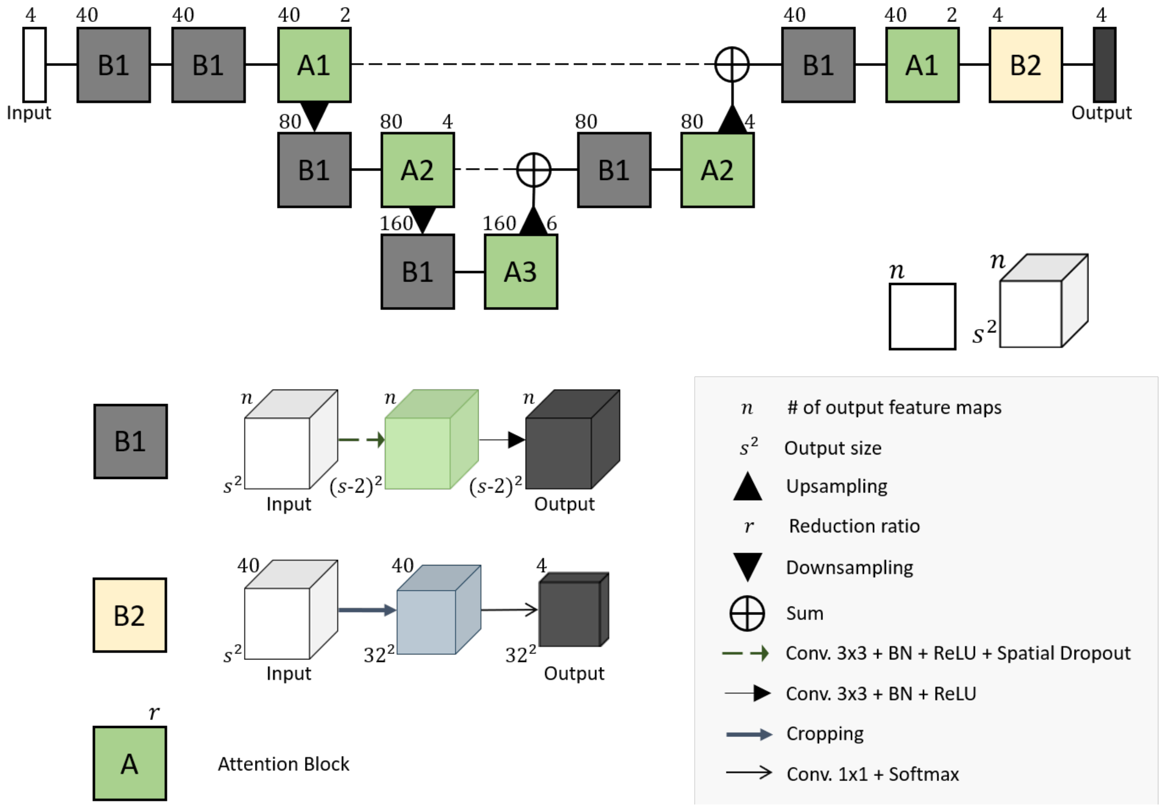


Figure 30: Architecture of the FCN with the attention block. BN stands for batch normalization.

totaling 285 subjects, and Leaderboard/Validation with 46 subjects from both HGG and LGG, but the grade is not revealed.

For each multimodal scan, there are four MRI sequences available: T1-weighted (T1), post-contrast T1-weighted (T1c), T2-weighted (T2), and Fluid-Attenuated Inversion Recovery (FLAIR). The provided images were formerly co-registered to the same anatomical template, interpolated to 1 mm isotropic resolution and skull stripped. Dimensions of each volume are $240 \times 240 \times 155$.

Each tumor was manually segmented into edema (class 2), necrosis/non-enhancing tumor (class 1) and enhancing tumor (class 4), by one to four raters, following the same annotation protocol, and their annotations were approved by experts. Evaluation is performed for the whole tumor (all classes), tumor core (all, excluding edema), and enhancing tumor. Only the manual annotations from the training set are publicy available. Thus, the evaluation is computed by the CBICA IPP online platform[1]. The development of the attention block was conducted in the training set, which was divided into 3 random non-overlapping sets, following a $60\%/20\%/20\%$ split across training, validation and test sets, respectively[2].

6.2.3 Training Settings

During training of the WT-FCN, crossentropy loss, the Adam optimizer with learning rate of 5×10^{-5}, weight decay of 1×10^{-6} , and spatial dropout probability of 0.05 were used. For training this binary whole tumor FCN, all tumor regions in manual segmentations were fused into a single label. For the

[1] https://ipp.cbica.upenn.edu/

[2] Subjects id in each set are available: https://github.com/sergiormpereira/rr_segse/

multi-class FCN setup, we use the one-cycle training method [109] together with AdamW optimizer [104] with the maximum LR set to 1×10^{-3} and the minimum to 5×10^{-5}, weight decay 5×10^{-7}, and 0.05 spatial dropout probability. The networks were trained for 143 epochs. He normal [95] was used to initialize the weights of the convolutional layers, and L_2 weight regularization penalty as regularization. All the hyperparameters were tuned using the validation set, before evaluation in the test set.

6.2.4 Evaluation Metrics

In the case of glioma segmentation, they can be defined as edema, necrosis/non-enhancing and enhancing tumor. For BRATS 2017 quantitative evaluation we follow the metrics used in the challenge associated with the database. Therefore, we use the Dice Coefficient (Dice), the sensitivity (Sens) and the 95^{th} percentile of the Hausdorff Distance (HD$_{95}$). The Dice is a measure of overlap, but it is sensitive to the size of the lesions, and does not provide information regarding under- and over-segmentation. Notwithstanding, such behavior can be inferred from Sens. The distance metrics provide insights about the correctness of the segmentations contour [8].

6.2.5 Software and Hardware

The functions used throughout the various steps of this work were developed in Python [126]. More specifically, the proposed methods were implemented using Keras[3] with TensorFlow[4] backend.

The tests were performed on a computer with the Linux Mint 18 operating system, equipped with an NVIDIA GeForce GTX 1070 GPU, an Intel® CoreTM i7-6850k 3.6 GHz processor and 128 GB of RAM.

The models employing attention blocks with a higher number of branches or with higher filter sizes in the context aggregation operation require more training time. Also, the models employing average pooling in the attention blocks normally have higher time requirements. The fastest models can take about 20 hours to train, while heavier models can reach up to 27 hours.

6.3 Results and Discussion

In this section, we investigate brain tumor segmentation in MRI images using FCN-based approaches coupled with an adaptive-scale context attention module. We start by the validation of the various components of the attention model and the identification of the best model. Finally, the best model is compared with other state-of-the-art methods.

6.3.1 Context Aggregation Operation

We tested two types of blocks that differ in the context aggregation operation. Some use average pooling operations to aggregate context, referred to as AVG models, and others use dilated convolutions,

[3] Keras: The python deep learning library in https://keras.io/

[4] https://www.tensorflow.org/.

which we call DIL models. The dilation rates d of our attention block branches were established according to the scale of feature maps as the receptive field increases with the number of convolutional and pooling layers. So, after each layer, each unit represents a larger area of the input space and may require less dilation or smaller kernel sizes. Similarly, the kernel sizes and strides of the branches with average pooling layers were defined according to the scale of the layer it is operating in. Thus, in a network, the attention blocks configuration changes across the network. Likewise, we set the reduction factor r in $\{A1, A2, A3\}$ to $\{2, 4, 6\}$.

The dilated convolutions kernel sizes and dilation rates, and average pooling filter sizes of the attention blocks for each tested system are shown in Table 5. In each model, the attention block is composed of a set of branches with dilated convolutions or of a set of branches with average pooling to aggregate the context. Additionally, to allow the attention block to relate different contexts, we also added the case in which the global average of the features maps is considered, since this is the largest possible context. Thus, the attention blocks contain a cSE branch (Section 6.1.1.1) whose general approach is presented in [66].

To easy the identification of the different attention modules, the name of each model is defined by its configuration, i.e., by the operation and the larger filter size used to aggregate context in the attention blocks.

Table 5: Configuration of the different attention block variants. A1, A2, and A3 refer to blocks A1, A2, and A3 of Figure 30

Context Aggregation		**A1**		**A2**		**A3**		**cSE branch**
Operation	**Name**	**kernel size**	**dil. rate**	**kernel size**	**dil. rate**	**kernel size**	**dil. rate**	
Dilated Convolutions	DIL-K3D4	3	$\{4, 3, 2, 1\}$	3	$\{3, 2, 1\}$	3	$\{3, 2, 1\}$	✓
	DIL-K5D3	$\{5, 3, 3, 3\}$	$\{3, 3, 2, 1\}$	$\{5, 3, 3\}$	$\{2, 2, 1\}$	$\{5, 3, 3\}$	$\{2, 2, 1\}$	✓
		kernel size		**kernel size**		**kernel size**		
Average Pooling	AVG-11-11	$\{11, 7, 3\}$		$\{11, 7, 3\}$		$\{11, 7, 3\}$		✓
	AVG-15-11	$\{15, 11, 7, 3\}$		$\{11, 7, 3\}$		$\{11, 7, 3\}$		✓
	AVG-9-7	$\{9, 7, 5, 3\}$		$\{7, 5, 3\}$		$\{7, 5, 3\}$		✓
	AVG-11-7	$\{11, 9, 7, 5, 3\}$		$\{7, 5, 3\}$		$\{7, 5, 3\}$		✓
	AVG-13-7	$\{13, 11, 9, 7, 5, 3\}$		$\{7, 5, 3\}$		$\{7, 5, 3\}$		✓

The quantitative results obtained for the baseline networks and for the different configurations of the attention block for the proposed segmentation approach are depicted in Table 6. Figure 31 shows the qualitative results as segmentation examples.

We started from a baseline consisting of a FCN identical to the one shown in Fig. 30, but where the attention block is absent. Then, we evaluate the proposed attention block variants. We can observe that the addition of the attention block leads to better Dice in all the tumor regions and better Sens in enhancing and tumor core, regardless of its configuration. However, in one of the AVG models, the AVG-11-7 variant, the Sens for the whole tumor is below the baseline value. Also, in some variants, the HD$_{95}$ increased. Namely, all the AVG models exhibit higher HD$_{95}$ in the enhancing region. This may be due to over-segmentation, considering that the Sens in these models is higher than the one obtained by the baseline.

Globally comparing DIL with AVG attention blocks, it is possible to observe that DIL blocks outperform the AVG blocks in terms of HD$_{95}$ considering the overall performance in all the tumor regions, which suggests that the contours of the DIL blocks segmentations are closer to the annotated ones and more detailed. This can be a result of the feature map smoothing effect of the average pooling filters that may cause loss of details in the contours.

Table 6: Results obtained in BRATS 2017 Leaderboard set. Bold results are the best among all attention block variants

	Dice			**Sens**			**HD$_{95}$**		
Method	**Whole**	**Core**	**Enh.**	**Whole**	**Core**	**Enh.**	**Whole**	**Core**	**Enh.**
Baseline	0.888	0.767	0.724	0.890	0.775	0.784	7.21	9.54	5.60
DIL-K3D4	**0.897**	0.808	**0.739**	**0.896**	0.803	0.798	6.17	8.45	5.29
DIL-K5D3	0.893	0.815	0.738	0.892	**0.814**	0.798	6.76	**7.98**	**5.23**
AVG-11-11	0.896	**0.817**	0.729	0.890	0.812	0.797	6.48	8.55	5.89
AVG-15-11	0.895	0.802	0.735	0.890	0.800	0.801	6.63	8.75	6.06
AVG-9-7	0.896	0.801	0.732	0.890	0.794	0.806	6.16	9.37	6.16
AVG-11-7	0.894	**0.817**	0.736	0.888	0.809	**0.809**	6.35	8.76	6.67
AVG-13-7	0.894	0.816	**0.739**	0.892	0.808	0.802	6.42	8.26	6.16

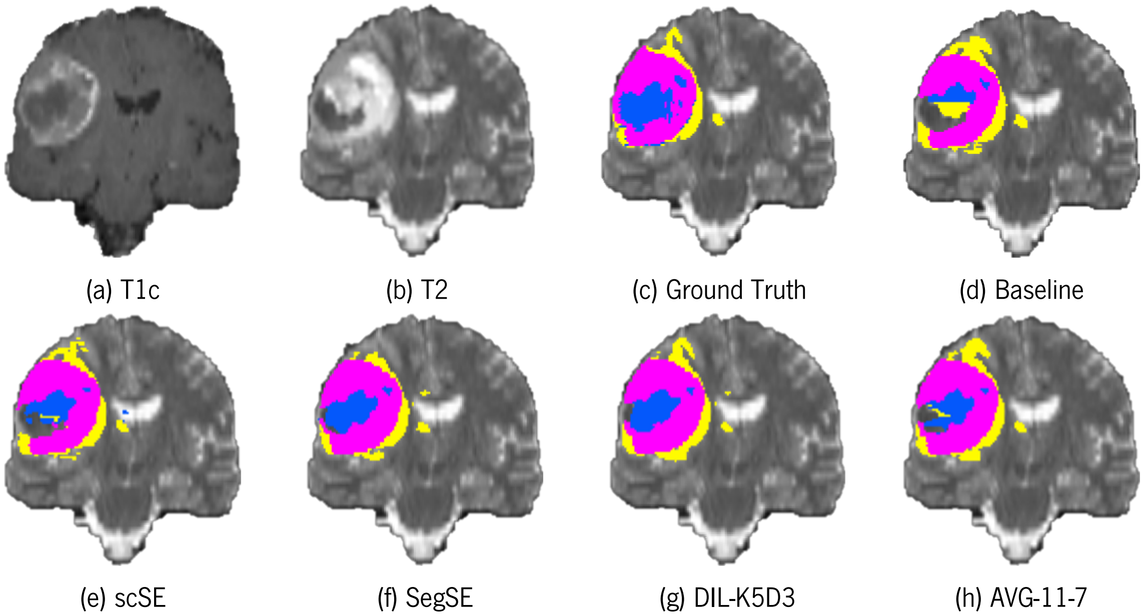
Considering DIL-K3D4 and AVG-9-7, in which the coverage area of the operations responsible for aggregating context (dilated convolutions and average pooling, respectively) is equivalent, we can notice that DIL-K3D4 yields better Dice scores in all the tumor regions, and it shows better or equal performance in all the remaining metrics but in the Sens of the enhancing tumor. The superiority of DIL-K3D4 over AVG-9-7 is especially expressive in the whole tumor region, where it achieves the highest Dice and Sens within all systems.

Analyzing the effect of changing the dilation rate and kernel size in the networks with DIL blocks, it is possible to verify constant behavior across the results for all tumor classes. In particular, DIL-K3D4 performs better in whole tumor and DIL-K5D3 in tumor core for all metrics, and they present similar performance in the Dice and Sens of the enhancing tumor. These results support the idea that, in networks with DIL blocks, increasing the context allows better discrimination between core and edema. Also, increasing the context brought benefits in the segmentation of brain tumors as, globally, DIL-K5D3 performs better than DIL-K3D4, presenting higher improvements over the baseline.

Among the models with attention blocks with average pooling operations, the blocks AVG-9-7, AVG-11-7 and AVG-13-7 vary only in the configuration of the first level of the network, specifically, there is a progressive increase in the number of filters, and in the range of the scales of these filters. Between models AVG-9-7 and AVG-11-7 it is possible to verify that the addition of a branch in the first U-Net level (A1 blocks of Figure 30) was mainly beneficial in distinguishing between core and edema, improving all tumor core metrics. The Dice of the whole tumor and enhancing region and the Sens oh the latter slightly increases. In terms of HD$_{95}$, the AVG-11-7 performs worse in the whole tumor and enhancing region. Considering the AVG-13-7 system, the inclusion of an additional scale had little effect on Dice of all tumor regions and in the Sens of the whole tumor and tumor core. But, this model exhibits lower HD$_{95}$ in the tumor core and enhancing region than the AVG-11-7 model. However, the Sens score of the enhancing tumor decreased, and the HD_{95} has increased for the whole tumor region.

The blocks AVG-11-11, and AVG-15-11 also vary only in the configuration of the first level, with an increase in the number of filters and in the range of the scales of these filters, respectively. But, the configuration of the second and third U-Net levels is different from the previously mentioned models (AVG-9-7, AVG-11-7 and AVG-13-7). In this case, the introduction of an additional scale led to an improvement in the Dice and Sens of the enhancing tumor. Hence, using an average pooling layer with a larger area increased the discrimination between core and enhancing tumor. In contrast, the additional scale was detrimental in the core region, where AVG-15-11 achieves lower Dice and Sens, and HD_{95} has increased.

Globally we can identify models DIL-K3D4 and AVG-11-7 as the best of each context aggregation operation. The system DIL-K3D4 has the best Dice, Sens and HD_{95} in the whole tumor and the best Dice score in the enchancing region among the DIL models. Also, it achieves the same performance in terms of Sens and similar HD_{95} result in the the enchancing tumor, falling short of the DIL-K5D3 model only in the tumor core. The AVG-11-7 system achieves the best metrics in the Dice and Sens set among all AVG models. Qualitatively examining Fig. 31, both DIL-K3D4 (Fig. 31g) and AVG-11-7 (Fig. 31h) models result in better segmentations than the baseline model (Fig. 31d). Particularly, in this suject, the DIL-K3D4 model segments the whole tumor and tumor core more effectively, but AVG-11-7 seems to be superior in the enhancing tumor. Also, the AVG-11-7 model seems to roughly segment more intricate regions of the tumor, such as the upper part of the edema region, which may be due to the smoothing effect caused by the average pooling operation.



(a) T1c

(b) T2

(c) Ground Truth

(d) Baseline

(e) scSE

(f) SegSE

(g) DIL-K5D3

(h) AVG-11-7

Figure 31: MRI acquisitions and segmentation examples obtained by the baseline network, AVG-11-7 and DIL-K5D3 variants, seSE block and SegSE block. Colors identify different tumor regions: blue - necrosis and non-enhancing tumor, pink - enhancing tumor, and yellow - edema. The subject can be found in BRATS 2017 with ID Brats17_TCIA_430_1.

6.3.2 Comparison with the state-of-the-art

Table 7 presents the results obtained in BRATS 2017 Leaderboard with the baseline FCN, the best system using dilated convolutions in the attention block, and the best system using average pooling layers in the attention block. In this table, these methods are compared with the state of the art in BRATS 2017 Leaderboard. We separate single prediction approaches[5] from ensembles because ensembles have a competitive advantage. The reason for this is that ensembling is a way of improving the performance by reducing the effect of the possible high variance of each individual model by combining a variety of models, as different models make different mistakes [44, 148]. In the addressed methods, ensembles resulted from training a FCN in different MRI planes [149–151], from training multiple FCNs with different settings [148], or from using the models previously trained for k-fold cross-validation in the training set [152].

In addition, we evaluated the block proposed by Roy et al. [72]. In their work, the authors implemented a joint spatial and channel squeeze and excitation attention mechanism where a single attention map is inferred for all channels. Pereira et al. [73] reproduced this strategy using the parameters proposed by Roy, but with the same settings as their SegSE block, in a hierarchical segmentation approach. We present the results of the spatial and channel squeeze and excitation block before fine-tuning.

Table 7: Results obtained in BRATS 2017 Leaderboard set. Underlined scores show the best resuls for each tumor region among single prediction approaches. Bold results are the best among all methods

			Dice			**Sens**			**HD$_{95}$**		
Approach	**Method**		**Whole**	**Core**	**Enh.**	**Whole**	**Core**	**Enh.**	**Whole**	**Core**	**Enh.**
Ensemble	Kamnitsas et al. [148]	3D	0.901	0.797	0.738	0.895	0.762	0.7829	4.23	6.56	4.50
	Zhao et al. [151]	3D	0.887	0.794	0.754	-	-	-	-	-	-
	Isensee et al. [152]	3D	0.896	0.797	0.732	0.896	0.781	0.790	6.97	9.48	4.55
	Jungo et al. [150]	3D	0.901	0.790	0.749	0.900	0.760	0.800	5.41	7.49	5.38
	Wang et al. [149]	2.5D[6]	**0.905**	**0.838**	**0.786**	**0.915**	**0.822**	0.771	**3.89**	**6.48**	**3.28**
Single Prediction	Islam and Ren [153]	3D	0.876	0.761	0.689	0.837	0.711	0.706	9.82	12.36	12.94
	Jesson and Arbel [154]	3D	0.899	0.751	0.713	0.904	0.720	0.732	4.16	8.65	6.98
	Roy et al. [72]	2D	0.892	0.792	0.716	0.894	0.814	0.778	6.74	9.81	6.61
	Pereira et al. [73]	2D	0.895	0.798	0.733	0.883	0.786	0.776	5.92	8.95	5.07
	Baseline	2D	0.888	0.767	0.724	0.890	0.775	0.784	7.21	9.54	5.60
	DIL-K3D4	2D	0.897	0.808	0.739	0.896	0.803	0.798	6.17	8.45	5.29
	AVG-11-7	2D	0.894	0.817	0.736	0.888	0.809	**0.809**	6.35	8.76	6.67

In brain tumor segmentation, most of the works have been adopting 3D networks, as can be seen from Table 7, which, theoretically, allows the systems to consider the 3D nature of the MRI scans. At the same time, 3D networks come with additional concerns in terms of memory and computational requirements. Also, the number of parameters and the data imbalance problem are increased. In this work, we wanted to emphasize the potentiality of the proposed attention mechanism. Therefore, we started from a simple 2D baseline FCN without any sophisticated techniques for training. The successful state-of-the-art methods that use 3D FCNs employ more complex training or prediction methods. For instance, Jesson and Arbel

[5] Approaches where the predictions of a single model are used to produce the final segmentation.

[6] The autors divide the problem of brain tumor segmentation into three 3D binary networks to hierarchically segment each tumor region.

[154] train a single 3D FCN with multiple prediction layers, and employed a learning curriculum to deal with class imbalance. Kamnitsas et al. [148] used an ensemble of several 3D network architectures with different settings. Wang et al. [149] divide the segmentation task into three hierarchical binary problems, in which three FCNs are trained (one for each MRI plane), in each stage, and their predictions are ensembled.

Analyzing the single prediction methods, it is possible to state that our baseline is competitive, which is relevant in the sense that it is difficult to improve performance over a competitive approach. This supports the added benefit of the adaptive context attention modules for brain tumor semantic segmentation. As a matter of fact, our FCN with the AVG-11-7 attention block achieves the highest Dice score of the tumor core, as well as the highest Sens of the enhancing tumor region, over all the single prediction approaches, being very close to the DIL-K3D4 model in Dice of the enhancing tumor. Moreover, the DIL-K3D4 attention block outperforms the other single prediction approaches in the Dice of the enhancing tumor, and in the HD$_{95}$ of the tumor core, being very close to Jesson and Arbel [154] in the Dice score of the whole tumor. Additionally, it achieved the second best Sens and the third best HD$_{95}$ score for the whole tumor.

Qualitatively comparing the segmentation examples from the works of Roy et al. [72] and Pereira et al. [73] with our segmentation results, we observe that the scSE block (Fig. 31e), proposed in [72], appears to over-segment the edema on the one hand, and to under-segment necrosis and enhancing tumor on the other. The SegSE block (Fig. 31f), proposed in [73], results in better segmentations than the scSE block, improving the detection of the enhancing tumor and necrosis. However, it is not able to correctly delimit the edema. Both the proposed attention blocks, DIL-K3D4 (Fig. 31g) and AVG-11-7 (Fig. 31h) models, were able to improve the overall segmentation of all tumor tissues compared to the scSE block. Also, they were able to improve the edema segmentation, compared to the SegSE block. Additionaly, we notice that the SegSE block is able to locate the necrotic tissue better that the AVG-11-7 model, but in the edema the opposite happens. Finally, the DIL-K3D4 model appears to result in better segmentations compared to all the others.

Comparing with the ensemble methods, it is possible to observe that both our attention blocks, with dilated convolutions and average pooling, permitted our method to reach competitive results, especially in Dice and Sens. Regarding the HD$_{95}$ metric, it is negatively affected by the presence of false positives, especially if they are distant from the corresponding tumor region. Thus, ensembles may effectively handle this problem as different models make different predictive mistakes. Comparing with Kamnitsas et al. [148], whose work won BRATS 2017 Challenge edition by generalizing well to the Challenge set, our performance in Dice and Sens is similar or superior. Yet, Wang et al. [149] retain the leading Leaderboard set results. In fact, Wang et al. [149] were able to achieve the best overall values in all metrics but Sens of the enhancing tumor, where both our attention models surpass its classification and the one with context aggregation by average polling (AVG-11-7) achieved the highest score. However, Wang et al. [149] used a 3-stage binary segmentation strategy, in which three FCNs are trained (one for each MRI plane), in each stage, and their predictions are ensembled.

From the analysis of Table 7, it can also be seen that no method can achieve the best performance in all metrics simultaneously, reinforcing the existing difficulty in the area of brain tumor segmentation.Notwithstanding, the results obtained with the adaptive context attention module are competitive with the state of the art. Therefore, we can consider that our approach is as capable of segmenting tumors as other state-of-the-art works, at least in BRATS 2017 Leaderboard dataset.

6.4 Summary

Fully Convolutional Networks were developed for semantic segmentation enabling taking larger contexts into account. In this chapter, we explored the capabilities of FCNs coupled with adaptive attention blocks for brain image semantic segmentation.

Our segmentation scheme includes pre-processing, data augmentation, classification and post-processing steps. The main goal of pre-processing is to standardize the image characteristics and, if possible, to highlight the most interesting properties so that the segmentation task can be made easier. Data augmentation is employed in the training set to cope the limited quantity of data. Classification is made through encoder-decoder-based FCNs, following a hierarchical segmentation strategy. Finally, post-processing is used to refine results based on some prior knowledge.

We proposed a hierarchical brain tumor segmentation approach with an adaptive-scale context attention block for brain tumor segmentation. Even though FCN architectures define multi-scale processing and optimize the representations for the given task, some features are more important for detecting some classes than others. Additionally, the complex nature of medical images makes the task more challenging. Our attention mechanism comprises feature map recalibration and an adaptive context scheme to enable the selection of more important features and the generation of more complex features.

We show the ability of the proposed methodology for glioma segmentation. Indeed, adaptively selecting the context was shown to be better-suited than other attention approaches. The reason for this observation may be because our attention blocks enable the mixture and generation of complex features.

Chapter 7

Conclusions and Future Perspectives

Medical imaging techniques provide the ability to non-invasively assess the human body. Manual segmentation of body structures or tissues is a difficult, time–consuming task and prone to inter– and intra–rater variability, even for the most experienced specialists. Therefore, the main objective of this dissertation was the development of two automated computerized solutions for medical image segmentation using Deep Learning techniques. First, our work focused on applying an ensemble approach using stochastic weight averaging for retinal vessel segmentation. Then, in the development of an adaptive-scale context attention mechanism for the segmentation of brain tumors.

We have presented the developed approaches and the obtained results, as well as discussion wrapping up our observations, for each topic. In this final chapter, we sum up the main conclusions and future perspectives on opened lines of research.

7.1 Stochastic Weight Averaging for Retinal Vessel Semantic Segmentation

The morphological characteristics of the retinal vascular tree provide extremely important information for early diagnosis and monitoring of various diseases such as glaucoma diabetes or hypertension. These diseases have a huge impact on the quality of life of millions of people worldwide. Although not all pathological signs observed in fundus images are directly related to the retinal vascular tree, branching geometry and the condition of the vessels are valuable indicators of the health status of a patient. To assess the morphological characteristics of retinal blood vessels, segmentation is essential. Also, it is an intermediate step in the calculation of several indicators that are particularly useful in screening programs.

Over the years, several proposals for retinal vessel segmentation have been presented, with the ultimate goal of replacing manual segmentation and bypassing the associated disadvantages such as the need for human resources and intra- and inter-rater variability. Throughout this work, we sought to study an ensemble approach to improve the performance of retinal vessel segmentation methods.

In this study, we started from a pre-trained FCN with dilated convolutions and we reproduce two ensemble methods, SWA and fast-SWA. In the design of SWA, there are some hyperparameters that were studied and validated, so that the SWA approach is optimized for our task at hand. These approaches were able to improve our already competitive baseline, showing that weight averaging techniques for ensembling successfully explore the local minima region of the weight space, achieving a solution with better generalization. The results obtained by the proposed method show that SWA ensured a segmentation of great global efficacy, allowing the detection of more vessels. When compared to other state-of-the-art methods, the proposed strategy achieved a very promising performance with a simple methodology.

Concerning this work, one of the factors that can be developed is the study of strategies capable of simultaneously targeting thick and thin vessels. The majority of state-of-the-art methods are already extremely effective in segmenting the large caliber vessels. The remaining differences between current methods generally lie in the greater or lesser ability to avoid false detections or to detect thinner vessels. Thus, it is essential to develop multi-level and multi-scale methods capable of handling the complex nature of fundus images. A possible approach is the study of attention mechanisms for vessel segmentation. Attention mechanisms are designed to allow the systems to focus on the most relevant information and disregard less discriminative features [44]. In recent past, different attention strategies have been successfully proposed in the context of medical image segmentation [69, 70, 73, 146]. Therefore, we envision that in the near future the most successful systems for retinal vessel segmentation will include attention mechanisms to enhance the representational power of multi-scale features.

Another direction is the inclusion of more input channels with handcrafted features. The main motivation for this approach lies in a recent trend that, despite its great ability to learn good representations of data, Deep Learning methods can benefit from the inclusion of domain knowledge. Recently, Oliveira et al. [138] demonstrated that a FCN for vessel segmentation may benefit from the addition of channels from the wavelet transform. Therefore, models that join both domain knowledge and learned features are a potential line of research.

7.2 Attention Mechanisms with Adaptive-scale Context for Brain Tumor Segmentation

Gliomas, the most common type of brain tumors, have high mortality rates. In clinical practice, MRI has become the standard imaging technology for assessing these tumors. Multi-sequence MRI provides volumetric data and sequences with different contrasts, which are essential for glioma inspection. In clinical practice, the analysis methods physicians employ do not exploit the full potential of MRI images, and are very time consuming and prone to variability. Moreover, gliomas are very heterogeneous, and MRI acquisitions vary a lot with the equipment and acquisition site. This makes it very challenging to develop accurate image segmentation algorithms. In this dissertation, we focused our research on improving brain tumor segmentation using Deep Learning techniques.

In this work, we propose an adaptive-scale context attention block coupled with feature map recalibration for semantic segmentation. We show that the attention module is capable of selecting the most discriminative features and suppress the more irrelevant ones for the predicted class, improving performance over a competitive baseline. Further, to mitigate the class imbalance inherent in the problem, we employed a hierarchical FCN-based scheme. First, we do a coarse segmentation of the whole tumor, then we define a ROI, and segment all tissues inside the ROI. Also, our simple U-Net inspired network with the adapttive attention block achieves competitive results in BRATS 2017 Leaderboard set. The winner of BRATS 2017 Challenge [148] employed a large ensemble of CNNs with different architectures and trained with different settings, showing the difficulty in bringing forth an architecture that outperforms in every tumor class and metric. Our proposal is modular and independent of architecture. Thus, it can be easily incorporated into existing network architectures to increase their representational power.

Deep Learning-based systems often rely on increasing the depth of architectures to improve performance [155]. However, very deep networks may not be viable when working with medical images due to lack of labeled data. Therefore, we may also conclude that improving the performance of a CNN can be done through design strategies, and not only by increasing depth. In the context of brain tumor segmentation, deeper architectures have been proposed over time [144, 149, 156]. Nevertheless, there is no evidence that it is beneficial to use such deep networks. Our approach addresses automatic adaptive choice of context to extract more discriminative features. Therefore, we may conclude that improving the performance of a network can be done not only by increasing depth, but through design strategies that use adaptive feature learning.

In this work, the brain tumor segmentation scheme was designed for a multi-class 2D FCN as a proof of concept to evaluate the benefits introduced by the attention block. However, most of the works have been adopting 3D networks [148–154]. The whole volume 3D assessment is important as it theoretically allows the systems to consider the 3D nature of the MRI scans. Despite increasing computational and memory load, extending our proposal to a 3D setting could leverage system performance.

Additionally, we used conventional structural MRI images as input for our segmentation system. These MRI sequences are part of the consensus for standardized brain tumor imaging protocol [31]. Thus, methods based on these sequences have more potentiality of having practical value as they are systematically acquired in clinical practice. However, some other MRI sequences, such as, perfusion and diffusion MRI, may provide additional information. Meanwhile, research is needed regarding if they can be helpful in brain tumor assessment, and how to exploit them.

Finally, in the current system, a weighted summation of the attention branches is made according to the importance assigned by the softmax function. Therefore, the output features of the attention blocks correspond to a composition of scales in which all branches are taken into account, which may not be the best approach. It may be more desirable to select only the branch with the most significant features or a set of the most significant scales. This would entail, for example, adding a selection mechanism to the attention block. Therefore, we believe it might be useful to verify if these approaches would be benefical in brain tumor segmentation.

References

[1] Ahmed Elnakib, Georgy Gimel'farb, Jasjit S Suri, and Ayman El-Baz. Medical image segmentation: a brief survey. In *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, pages 1–39. Springer, 2011.

[2] Maedeh Sadat Fasihi and Wasfy B Mikhael. Overview of current biomedical image segmentation methods. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 803–808. IEEE, 2016.

[3] Nelly Gordillo, Eduard Montseny, and Pilar Sobrevilla. State of the art survey on mri brain tumor segmentation. *Magnetic resonance imaging*, 31(8):1426–1438, 2013.

[4] Duncan RR White, Alexander S Houston, William FD Sampson, and Graham P Wilkins. Intra- and interoperator variations in region-of-interest drawing and their effect on the measurement of glomerular filtration rates. *Clinical nuclear medicine*, 24(3):177–181, 1999.

[5] Clyde W Oyster. *The human eye: structure and function*. Sinauer Associates, 1999.

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

[7] H Royden Jones Jr, Jayashri Srinivasan, Gregory J Allam, and Richard A Baker. *Netter's Neurology E-Book*. Elsevier Health Sciences, 2011.

[8] 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.

[9] Helga Kolb, Eduardo Fernandez, Ralph Nelson, and Bryan William Jones. Webvision: Organization of the retina and visual system. 2005. *John Moran Eye Center, University of Utah, USA*.

[10] David H Hubel. *Eye, brain, and vision.* Scientific American Library/Scientific American Books, 1995.

[11] 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.

[12] Klara Landau and Malaika Kurz-levin. Retinal disorders. In *Handbook of clinical neurology*, volume 102, pages 97–116. Elsevier, 2011.

[13] Rama D Jager, William F Mieler, and Joan W Miller. Age-related macular degeneration. *New England Journal of Medicine*, 358(24):2606–2617, 2008.

[14] Robert N Weinreb, Tin Aung, and Felipe A Medeiros. The pathophysiology and treatment of glaucoma: a review. *Jama*, 311(18):1901–1911, 2014.

[15] Alan W Stitt, Timothy M Curtis, Mei Chen, Reinhold J Medina, Gareth J McKay, Alicia Jenkins, Thomas A Gardiner, Timothy J Lyons, Hans-Peter Hammes, Rafael Simo, et al. The progress in understanding and treatment of diabetic retinopathy. *Progress in retinal and eye research*, 51: 156–186, 2016.

[16] Tien Wong and Paul Mitchell. The eye in hypertension. *The Lancet*, 369(9559):425–435, 2007.

[17] M Bhargava, MK Ikram, and TY Wong. How does hypertension affect your eyes? *Journal of human hypertension*, 26(2):71, 2012.

[18] M Mirzaei, VB Gupta, VK Gupta, et al. Retinal changes in alzheimer's disease: Disease mechanisms to evaluation perspectives. 2018.

[19] 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.

[20] World Health Organization. Cancer fact sheet n 297, 2018.

[21] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 68(6):394–424, 2018.

[22] Vladimir Parpura, Michael T Heneka, Vedrana Montana, Stéphane HR Oliet, Arne Schousboe, Philip G Haydon, Randy F Stout Jr, David C Spray, Andreas Reichenbach, Thomas Pannicke, et al. Glial cells in (patho) physiology. *Journal of neurochemistry*, 121(1):4–27, 2012.

[23] Stuart Ira Fox. *Human Physiology 9th Editon*. McGraw-Hill press, New York, USA, 2006.

[24] 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.

[25] Freddie Bray, Jian-Song Ren, Eric Masuyer, and Jacques Ferlay. Global estimates of cancer prevalence for 27 sites in the adult population in 2008. *International journal of cancer*, 132(5):1133–1145, 2013.

[26] F Lafitte, S Morel-Precetti, N Martin-Duverneuil, A Guermazi, E Brunet, F Heran, and J Chiras. Multiple glioblastomas: Ct and mr features. *European radiology*, 11(1):131–136, 2001.

[27] 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.

[28] 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.

[29] William R Hendee and E Russell Ritenour. *Medical imaging physics*. John Wiley & Sons, 2003.

[30] Marc C Mabray, Ramon F Barajas, and Soonmee Cha. Modern brain tumor imaging. *Brain tumor research and treatment*, 3(1):8–23, 2015.

[31] 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.

[32] Tumor Segmentation Challenge and András Jakab. Segmenting brain tumors with the slicer 3d software.

[33] Patrick Y Wen, David R Macdonald, David A Reardon, Timothy F Cloughesy, A Gregory Sorensen, Evanthia Galanis, John DeGroot, Wolfgang Wick, Mark R Gilbert, Andrew B Lassman, et al. Updated response assessment criteria for high-grade gliomas: response assessment in neuro-oncology working group. *J Clin Oncol*, 28(11):1963–1972, 2010.

[34] 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.

[35] Yuanjie Zheng, Ebenezer Daniel, Allan A Hunter III, Rui Xiao, Jianbin Gao, Hongsheng Li, Maureen G Maguire, David H Brainard, and James C Gee. Landmark matching based retinal image alignment by enforcing sparsity in correspondence matrix. *Medical image analysis*, 18(6):903–913, 2014.

[36] Ana Maria Mendonça, António Sousa, Luís Mendonça, and Aurélio Campilho. Automatic localization of the optic disc by combining vascular and intensity information. *Computerized medical imaging and graphics*, 37(5-6):409–417, 2013.

[37] Chetan L Srinidhi, P Aparna, and Jeny Rajan. Recent advancements in retinal vessel segmentation. *Journal of medical systems*, 41(4):70, 2017.

[38] Sara Moccia, Elena De Momi, Sara El Hadji, and Leonardo S Mattos. Blood vessel segmentation algorithms—review of methods, datasets and evaluation metrics. *Computer methods and programs in biomedicine*, 158:71–91, 2018.

[39] Uyen TV Nguyen, Alauddin Bhuiyan, Laurence AF Park, and Kotagiri Ramamohanarao. An effective retinal blood vessel segmentation method using multi-scale line detection. *Pattern recognition*, 46 (3):703–715, 2013.

[40] Nicole Porz, Simon Habegger, Raphael Meier, Rajeev Verma, Astrid Jilch, Jens Fichtner, Urspeter Knecht, Christian Radina, Philippe Schucht, Jürgen Beck, et al. Fully automated enhanced tumor compartmentalization: man vs. machine reloaded. *PloS one*, 11(11):e0165302, 2016.

[41] 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.

[42] Mohak Shah, Yiming Xiao, Nagesh Subbanna, Simon Francis, Douglas L Arnold, D Louis Collins, and Tal Arbel. Evaluating intensity normalization on mris of human brain with multiple sclerosis. *Medical image analysis*, 15(2):267–282, 2011.

[43] Uro Vovk, Franjo Pernus, and Botjan Likar. A review of methods for correction of intensity inhomogeneity in mri. *IEEE transactions on medical imaging*, 26(3):405–421, 2007.

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

[45] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

[46] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pages 231–238, 1995.

[47] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[48] Michael P Perrone and Leon N Cooper. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS, 1992.

[49] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

[50] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[51] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, pages 8789–8798, 2018.

[52] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):993–1001, 1990.

[53] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[54] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[55] Holger Schwenk and Yoshua Bengio. Training methods for adaptive boosting of neural networks. In *Advances in neural information processing systems*, pages 647–653, 1998.

[56] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[57] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[58] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[59] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[60] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017.

[61] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[62] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[63] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[64] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[65] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[66] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[67] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[68] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[69] Yao Qin, Konstantinos Kamnitsas, Siddharth Ancha, Jay Nanavati, Garrison Cottrell, Antonio Criminisi, and Aditya Nori. Autofocus layer for semantic segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 603–611. Springer, 2018.

[70] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[71] 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.

[72] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *IEEE transactions on medical imaging*, 38(2):540–549, 2018.

[73] 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.

[74] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[75] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7289–7298, 2019.

[76] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[77] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[78] Koon-Pong Wong. Medical image segmentation: methods and applications in functional imaging. In *Handbook of biomedical image analysis*, pages 111–182. Springer, 2005.

[79] Suzana Herculano-Houzel and Roberto Lent. Isotropic fractionator: a simple, rapid method for the quantification of total cell and neuron numbers in the brain. *Journal of Neuroscience*, 25(10): 2518–2521, 2005.

[80] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[81] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[82] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE, 2009.

[83] 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.

[84] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[85] 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.

[86] 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.

[87] 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.

[88] Laurent Sifre and Stéphane Mallat. Rigid-motion scattering for image classification. *Ph. D. dissertation*, 2014.

[89] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[90] 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.

[91] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[92] F Chollet. Deep learning with python, vol. 1. *Greenwich, CT: Manning Publications CO*, 2017.

[93] 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.

[94] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[95] 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.

[96] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[97] 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.

[98] 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.

[99] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

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

[101] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[103] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[104] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

[105] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[106] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.

[107] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

[108] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[109] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[110] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[111] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[112] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[113] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194, 2001.

[114] Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

[115] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006.

[116] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[117] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[118] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[119] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Improving consistency-based semi-supervised learning with weight averaging. *arXiv preprint arXiv:1806.05594*, 2, 2018.

[120] 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.

[121] 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(4):501–509, 2004.

[122] Christopher G Owen, Alicja R Rudnicka, Robert Mullen, Sarah A Barman, Dorothy Monekosso, Peter H Whincup, Jeffrey Ng, and Carl Paterson. Measuring retinal vessel tortuosity in 10-year-old children: validation of the computer-assisted image analysis of the retina (caiar) program. *Investigative ophthalmology & visual science*, 50(5):2004–2010, 2009.

[123] Aleksandra Popovic, Matías De la Fuente, Martin Engelhardt, and Klaus Radermacher. Statistical validation metric for accuracy assessment in medical image segmentation. *International Journal of Computer Assisted Radiology and Surgery*, 2(3-4):169–181, 2007.

[124] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[125] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[126] Guido Van Rossum et al. Python programming language. In *USENIX annual technical conference*, volume 41, page 36, 2007.

[127] Sohini Roychowdhury, Dara D Koozekanani, and Keshab K Parhi. Iterative vessel segmentation of fundus images. *IEEE Transactions on Biomedical Engineering*, 62(7):1738–1749, 2015.

[128] Jiong Zhang, Behdad Dashtbozorg, Erik Bekkers, Josien PW Pluim, Remco Duits, and Bart M ter Haar Romeny. Robust retinal vessel segmentation via locally adaptive derivative frames in orientation scores. *IEEE transactions on medical imaging*, 35(12):2631–2644, 2016.

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

[130] 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.

[131] Qiaoliang Li, Bowei Feng, LinPei Xie, Ping Liang, Huisheng Zhang, and Tianfu Wang. A cross-modality learning approach for vessel segmentation in retinal images. *IEEE transactions on medical imaging*, 35(1):109–118, 2016.

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

[133] 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.

[134] 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.

[135] 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.

[136] 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.

[137] 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.

[138] 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.

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

[140] 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.

[141] Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Transactions on Biomedical Engineering*, 59(9): 2538–2548, 2012.

[142] Buket D Barkana, Inci Saricicek, and Burak Yildirim. Performance analysis of descriptive statistical features in retinal vessel segmentation via fuzzy logic, ann, svm, and classifier fusion. *Knowledge-Based Systems*, 118:165–176, 2017.

[143] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[144] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.

[145] 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.

[146] 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.

[147] 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.

[148] 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.

[149] 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.

[150] Alain Jungo, Richard McKinley, Raphael Meier, Urspeter Knecht, Luis Vera, Julián Pérez-Beteta, David Molina-García, Víctor M Pérez-García, Roland Wiest, and Mauricio Reyes. Towards uncertainty-assisted brain tumor segmentation and survival prediction. In *International MICCAI Brainlesion Workshop*, pages 474–485. Springer, 2017.

[151] Xiaomei Zhao, Yihong Wu, Guidong Song, Zhenye Li, Yazhuo Zhang, and Yong Fan. 3d brain tumor segmentation through integrating multiple 2d fcnns. In *International MICCAI Brainlesion Workshop*, pages 191–203. Springer, 2017.

[152] 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.

[153] Mobarakol Islam and Hongliang Ren. Multi-modal pixelnet for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 298–308. Springer, 2017.

[154] 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.

[155] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[156] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.