UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica e de Computação

Diedre Santos do Carmo

Deep Learning for Hippocampus Segmentation

Aprendizado Profundo para Segmentação do Hipocampo

Campinas

2020

Diedre Santos do Carmo

Deep Learning for Hippocampus Segmentation

Aprendizado Profundo para Segmentação do Hipocampo

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Computer Engineering.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Engenharia de Computação.

Supervisor: Prof. Dr. Roberto de Alencar Lotufo

Co-supervisor: Profa. Dra. Letícia Rittner

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Diedre Santos do Carmo, orientada pelo Prof. Dr. Roberto de Alencar Lotufo e co-orientada pela Profa. Dra. Letícia Rittner.

Campinas

2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

C213d

Carmo, Diedre Santos do, 1993-
Deep learning for hippocampus segmentation / Diedre Santos do Carmo. – Campinas, SP : [s.n.], 2020.

Orientador: Roberto de Alencar Lotufo.
Coorientador: Letícia Rittner.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Aprendizado profundo. 2. Hipocampo (Cérebro). 3. Alzheimer, Doença de. 4. Epilepsia. I. Lotufo, Roberto de Alencar, 1955-. II. Rittner, Letícia, 1972-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Aprendizado profundo para segmentação do hipocampo
Palavras-chave em inglês:
Deep learning
Hippocampus (Brain)
Alzheimer's disease
Epilepsy
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Roberto de Alencar Lotufo [Orientador]
Fernando José Von Zuben
Nina Sumiko Tomita Hirata
Data de defesa: 29-06-2020
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-5922-9120
- Currículo Lattes do autor: http://lattes.cnpq.br/6252748421778841

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Diedre Santos do Carmo **RA:** 211492
Data da defesa: 29/06/2020

Dissertation Title: "Deep Learning for Hippocampus Segmentation".
Titulo da Dissertação: "Aprendizado Profundo para Segmentação do Hipocampo".

Prof. Dr. Roberto de Alencar Lotufo (Presidente, FEEC/UNICAMP)
Prof. Dr. Fernado José Von Zuben (FEEC/UNICAMP)
Profa. Dra. Nina Sumiko Tomita Hirata (IME/USP)

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora,encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Acknowledgements

Primeiramente, gostaria de agradecer minha família: meus pais, Carlos e Denise, meus irmãos e sobrinha, meus avós, e toda minha família em Salvador, por todo amor e suporte à minha jornada. À Laís, agradeço por todo o amor e companheirismo, palavras de encorajamento e suporte, durante todo esse período, tanto nos momentos difíceis quanto nos momentos felizes.

Me sinto honrado pela oportunidade de trabalhar com o meu orientador, Professor Roberto Lotufo, e minha co-orientadora Professora Letícia Rittner. Agradeço todas as discussões, orientações e ideias geniais, que levaram este trabalho a patamares além do que eu imaginava. Obrigado pela oportunidade de trabalhar nesta instituição de excelência. Aos meus colegas de laboratório do MICLab, LCA e UNICAMP em geral, vocês também me ajudaram imensamente, tanto em relação a discusssões acadêmicas, quanto pela recepção calorosa, tornando essa jornada mais agradável e divertida.

O presente trabalho foi realizado com o apoio do processo nº 2018/00186-0, convênio Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). As opniões, hipóteses e conclusões ou recomendações expressas neste material são de responsabilidade dos autores e não necessariamente refletem a visão da FAPESP e da CAPES. Agradeço também aos nossos parceiros do Instituto Brasileiro de Neurociência e Neurotecnologia (BRAINN) por fornecerem parte dos dados e o conhecimento médico necessário para este trabalho.

A todos, Obrigado!

"Faça e faça bem feito tudo que tiver de ser feito."
(Carlos Carmo)

Abstract

Segmentation of the hippocampus in magnetic resonance imaging (MRI) is of fundamental importance for the diagnosis, treatment and investigation of neuropsychiatric disorders. Automatic segmentation is an active research field, with many traditional atlas-based works and, recently, deep learning based methods. This thesis examines the state-of-the-art in automated hippocampal segmentation with Deep Learning and proposes a method for automatic segmentation that contains recent advances in this field of research. A public study on Alzheimer's disease called HarP with manual hippocampal annotations is used for validation during the development of the methodology. Additionally, a hypothesis is raised that current state-of-the-art methods are not ready to deal with cases of hippocampus resection due to the treatment of epilepsy.

The methodology, called Extended 2D Consensus Hippocampus Segmentation, is justified in theory and later validated in experiments. A consensus of 2D CNNs based on a modification of the UNet architecture assesses all three orthogonal orientations of the input volume. This consensus is followed by post-processing using 3D labeling. Traditional design choices inspired by literature are used, such as data augmentation and transfer learning with encoder weight initialization. The now traditional UNet CNN architecture is modified, resulting in better performance. In addition, the RADAM optimizer and Boundary Loss, recently proposed in the literature, are implemented and show superior performance when compared to other traditional options.

The performance of E2DHipseg is analyzed together with other recent methods of Deep Learning in two domains: the HarP benchmark and an internal epilepsy data set, called HCUnicamp. HCUnicamp differs significantly from examinations of healthy individuals with Alzheimer's, due to the presence of patients that undergone hippocampal resection surgery. E2DHipseg outperforms other methods in the literature in the Alzheimer's and Epilepsy test data sets, with the code and binary executable available online. However, no method achieves good performance in cases of hippocampus resection. As a final experiment, E2DHipseg is trained on the epilepsy data, resulting in improved results.

Keywords: deep learning; hippocampus (brain); Alzheimer's disease; epilepsy

Resumo

A segmentação do hipocampo em ressonância magnética (RM) é de fundamental importância para o diagnóstico, tratamento e investigação de distúrbios neuropsiquiátricos. A segmentação automática é um campo de pesquisa ativo, com muitos métodos tradicionais baseados em atlas e modelos utilizando-se de aprendizado profundo sendo recentemente propostos. Esta tese examina o estado da arte na segmentação automatizada de hipocampo com Aprendizado Profundo e propõe um método para segmentação automática que contém recentes avanços deste campo de pesquisa. Um estudo público sobre a doença de Alzheimer chamado HarP com anotações do hipocampo é usado para validação durante o desenvolvimento da metodologia. Paralelamente, é levantada uma hipótese de que os métodos atuais não são ideais para casos de ressecção do hipocampo devido ao tratamento da epilepsia.

A metodologia, denominada Extended 2D Consensus Hippocampus Segmentation, é justificada em teoria e posteriormente validada em experimentos. Um consenso de CNNs 2D baseadas numa modificação da arquitetura U-Net avalia todas as três orientações ortogonais do volume de entrada. Esse consenso é seguido por um pós-processamento usando rotulagem 3D. Escolhas tradicionais de design inspiradas na literatura são usadas em seu desenvolvimento, como aumento de dados e transferência de aprendizado com pré-inicialização dos pesos do codificador. A tradicional arquitetura de CNNs UNet é modificada, resultando em melhor desempenho. Além disso, o otimizador RADAM e a Boundary Loss, recentemente propostos na literatura, são implementados e mostram desempenho superior quando comparados a outras opções tradicionais.

O desempenho do E2DHipseg é analisado juntamente com outros métodos recentes de Aprendizado Profundo em dois domínios: o benchmark HarP e um conjunto de dados interno de epilepsia, chamado HCUnicamp. O HCUnicamp difere significativamente dos exames de indivíduos saudáveis e com Alzheimer, devido à presença de pacientes submetidos à cirurgia de ressecção do hipocampo. O E2DHipseg supera outros métodos da literatura nos conjuntos de dados de teste de Alzheimer e Epilepsia, com o código e o executável binário disponíveis on-line. No entanto, nenhum método alcança bom desempenho nos casos de ressecção do hipocampo. Como um experimento final, E2DHipseg é treinado nos dados de epilepsia, o que resulta em melhora dos resultados.

Palavras-chaves: aprendizado profundo; hipocampo (cérebro); Alzheimer, doença; epilepsia

List of Figures

List of Tables

List of Acronyms

TLE	Temporal Lobe Epilepsy
DL	Deep Learning
NN	Neural Network
CNN	Convolutional Neural Network
FCNN	Fully Convolutional Neural Network
MRI	Magnetic Resonance Imaging
CPU	Central Processing Unit
GPU	Graphics Processing Unit
GUI	Graphical User Interface
CN	Control Normal
MCI	Mild Cognitive Impairment
AD	Alzheimer's Disease
SGD	Stochastic Gradient Descent
ADAM	Adaptative Moment Estimation
RADAM	Rectified Adaptative Moment Estimation
LR	Learning Rate
MSE	Mean Square Error
BCE	Binary Cross Entropy
E2D	Extended 2D
GDL	Generalized Dice Loss
E2DHipseg	Extended 2D Consensus Hippocampus Segmentation

Contents

1 Introduction

The hippocampus is a small, medial, subcortical brain structure related to long and short term memory (ANDERSEN, 2007). Hippocampal segmentation (Figure 1) from magnetic resonance imaging (MRI) is of great importance for research of neuropsychiatric disorders and can also be used in the preoperatory investigation of pharmacoresistant temporal lobe epilpesy (GHIZONI *et al.*, 2015). The hippocampus can be affected in shape and volume by different pathologies, such as the neurodegeneration associated to Alzheimer's disease (PETERSEN *et al.*, 2010), or surgical intervention to treat temporal lobe epilepsy (GHIZONI *et al.*, 2017). The medical research of these diseases usually involves manual segmentation of the hippocampus, requiring time and expertise in the field. The high-cost associated to manual segmentation has stimulated the search for effective automatic segmentation methods. Some of those methods, such as FreeSurfer (FISCHL, 2012), are already used as a starting point for a subsequent manual finer segmentation later (MCCARTHY *et al.*, 2015).

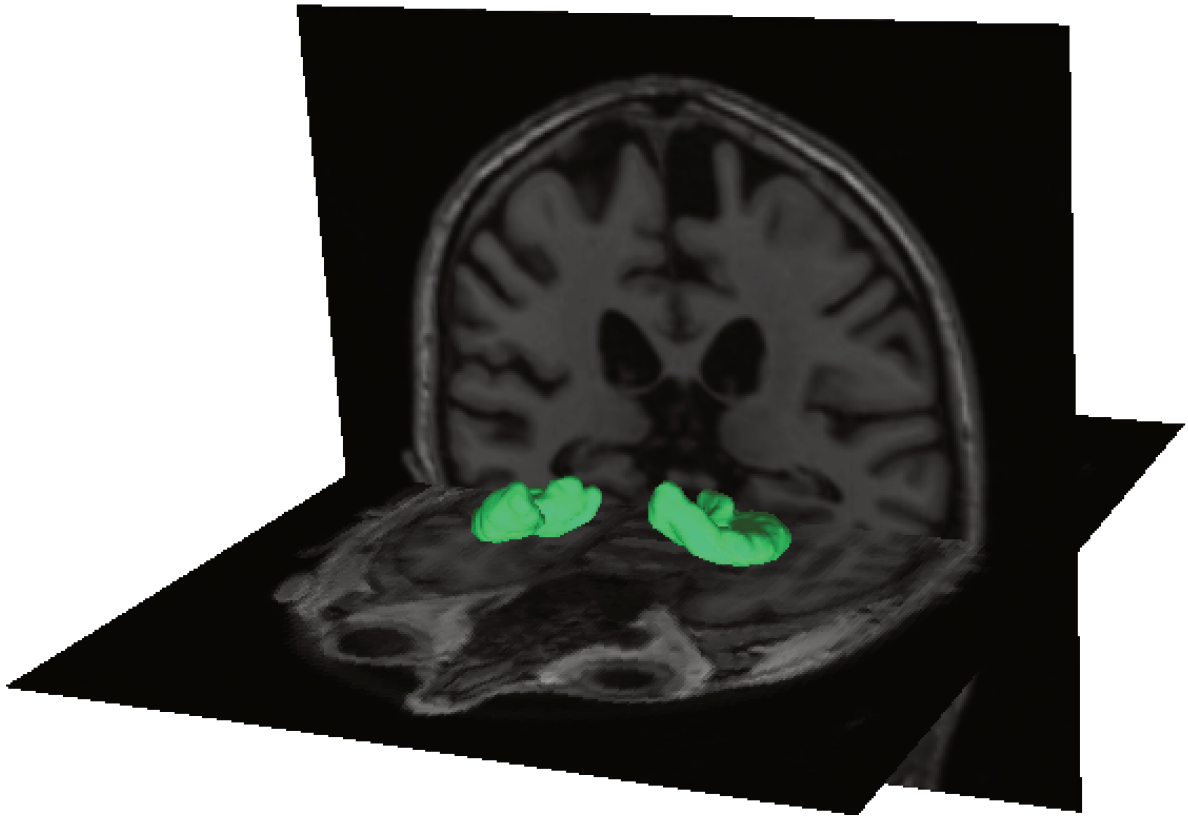
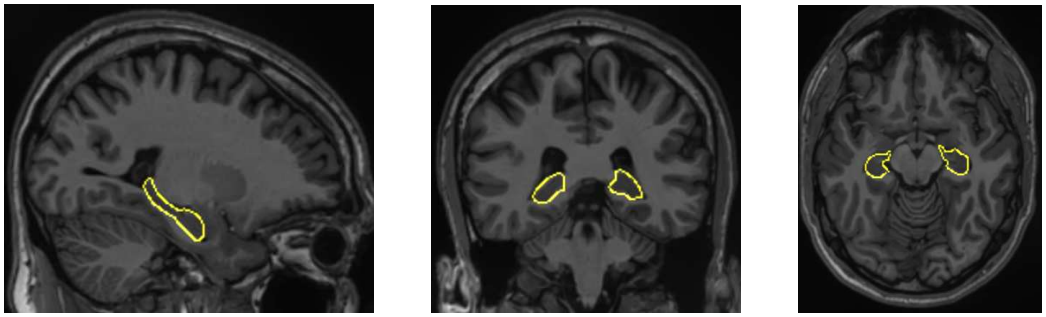


Figure 1 – 3D rendering of the manual annotation of one of the HarP dataset volumes.



(a) sagittal (b) coronal (c) axial

Figure 2 – Sample slices of an MRI volume, in every orthogonal orientation. In yellow, manual hippocampus annotation's borders.

1.1 The Hippocampus

Humans and other mammals have two hippocampi, one in each side of the brain (Figure 2). As a part of the limbic system, it has an important role in the consolidation of information from short-term to long-term memory and spatial memory. Anatomically, it consists of gray matter tissue elevating from the floor of each lateral ventricle to the temporal horn, only being visible through dissection (ANDERSEN, 2007). There is not a full consensus of which neighbouring tissues are part of its definition, with variations among publications in this topic (MARTIN, 2003; AMARAL; LAVENEX, 2007).

The hippocampus is one of the first brain regions to suffer damage in Alzheimer's disease and other forms of dementia. The atrophies in the Hippocampus due to Alzheimer can be visualized with MRI scans. These atrophies can vary in severity according to which stage of Alzheimer's the patient is in (PETERSEN *et al.*, 2010).

The hippocampus can also be affected by medial temporal lobe epilepsy (TLE). TLE is a disorder of the nervous system that can cause unprovoked seizures in the temporal lobe, lasting one to two minutes (DISORDERS *et al.*, 2015). While treatment can be done with anticonvulsants, in some cases resection of one of the hippocampi may be the only effective option to avoid complications (GHIZONI *et al.*, 2015).

1.2 Motivation

Manual segmentation of the Hippocampus can take hours. Existing automated commercial methods such as FreeSurfer (FISCHL, 2012) can take a whole day to segment a volume. Our work is inspired by the need to reduce the computational time of automatic hippocampus segmentation, while at the same time achieving better performance than traditional methods. This thesis uses a rising approach in the literature, namely, Fully Convolutional Neural Networks (FCNNs). Recently, some works have used

CNNs with promising runtime, in the order of seconds, and accuracy in the high 80s Dice (WACHINGER *et al.*, 2018; THYREAU *et al.*, 2018; XIE; GILLIES, 2018; CHEN *et al.*, 2017). Current literature supports that there is space for development of continuously better Deep Learning based methods. With that in mind, this thesis aims to deliver a deep learning based method focused on high Dice and low runtimes.

Additionally, while conducting research on Epilepsy and methods for hippocampus segmentation, another subject got the author's attention. Many of those methods, including the proposed method, focus on training and evaluating on healthy scans, or patients of Alzheimer's disease, due to that being what is publicly available. With that in mind, this thesis also includes a dataset from a different domain in testing. This in-house dataset, named HCUnicamp, contains scans from epilepsy patients pre and post hippocampus removal surgery, with very different atrophies to that found in public Alzheimer's data or healthy subjects.

Another motivation of this work is to provide an easy to use hippocampus segmentation method, given many other works publish results but do not provide easy to use tools for physicians or researchers. Open science has been very important to the fast development of Deep Learning methods, and making the results of this thesis public and reusable is important to encourage extensions and improvements in future work (NOSEK *et al.*, 2015).

1.3 Objectives

The thesis has the following main objectives:

- Combine CNN architecture ideas from the literature and novel ideas into a state-of-the-art hippocampus segmentation method. The method should be fast, with runtime in the order of seconds.
- Test the method's performance alongside others in the literature in public benchmarks and in a challenging epilepsy dataset.
- Make the method available for external use, without dependencies on multiple libraries and with a simple user interface.

1.4 Contributions

This thesis proposes a hippocampus segmentation method consisting of evaluating the consensus of volumes generated by three separate U-Net like (RONNEBERGER *et al.*, 2015) 2D CNNs. Modifications on the base U-Net architeture are performed and studied, such as encoders weight initialization with ImageNet VGG11 weights (SIMONYAN;

ZISSERMAN, 2014), applying residual connections from ResNet (HE *et al.*, 2016) to convolutional block and others. Each 2D network is trained on each brain orientation; sagital, coronal and axial. Traditional 3D labeling post-processing for false positive removal is implemented. Several studies in hyperparameter's definitions such as soft targets, custom losses and training were performed, with some success and failure cases.

An interesting characteristic of this project is the use of an inhouse 3T epilepsy dataset, containing manually annotated MRI scans with surgically removed hippocampus, a big difference to the data used in most similar research. Public data from Alzheimer's disease studies in the form of the HarP dataset is also used in this thesis, allowing for comparisons with other methods.

In addition, the method runs with a low memory footprint and seconds of runtime in a common computer, with ease of use features for doctors and researchers. Finally, the method is validated in HCUnicamp and HarP, including comparisons with other methods. The method presented on this thesis reachs state-of-the-art performance on HarP and beats other recent Deep Learning based hippocampus segmentation methods in HCUnicamp. Code is open source and weights are available for the community in <github.com/MICLab-Unicamp/e2dhipseg>, alongside a binary release for ease of use.

1.5 Outline of the Thesis

This thesis is organized as follows: Chapter 2 presents a literature review of the Deep Learning in Medical Imaging field and recent Deep Learning based hippocampus segmentation methods. More details to the involved data and implementation means are in Chapter 3. A detailed description of our hippocampus segmentation methodology is in Chapter 4. Chapter 5 has experimental results from our methodology development and qualitative and quantitative comparisons with other methods in HarP and HCUnicamp, while Chapter 6 has extended discussion of those results and conclusion.

2 Literature Review

This chapter presents a brief introduction to Deep Learning, its application to medical imaging segmentation and, more specifically, hippocampus segmentation.

2.1 Deep Learning

Nowadays, deep learning is taking the computer vision world by storm, to the point of being called a "revolution" (BENGIO *et al.*, 2015). Deep learning is a form of representational learning using very large neural networks, being used in various learning environments, from face identification (SUN *et al.*, 2014) to social media big data (LEVI; HASSNER, 2015). Usually, deep learning algorithms have a cascade of nonlinear processing units starting from raw data, generating many layers of representation from low to high levels of abstraction, and use some form of optimization for training (LECUN *et al.*, 2015).

One example of Deep Learning implementation is Convolutional Neural Networks (CNN). Starting from raw data (such as pixels from an image), CNNs apply various non-linear convolutions and pooling stages connected by weights to extract the most relevant features of the image for the intended application, through optimization of a target function. This eliminates the need for manual feature extraction (KRIZHEVSKY *et al.*, 2012). In a classification case, those convolutional features are then fed to densely connected layers that compute the final answer. The concept of CNNs has existed for a long time, but only recently the proliferation of large datasets and GPU computational power has allowed for better than traditional machine learning performance when using CNNs. CNN based approaches have been achieving state-of-the-art in most computer vision applications.

In segmentation applications, instead of fully connected classification outputs, upsampling or transposed convolutions can bring the abstract representation back to an output layer of similar spatial resolution to the input (RONNEBERGER *et al.*, 2015). This output is usually in the form of sigmoid activations or softmax outputs, representing segmentation masks or more abstract concepts such as attention (OKTAY *et al.*, 2018). Due to not having a fully connected layer and being composed mainly of convolutions, these are sometimes called Fully Convolutional Neural Networks (FCNNs).

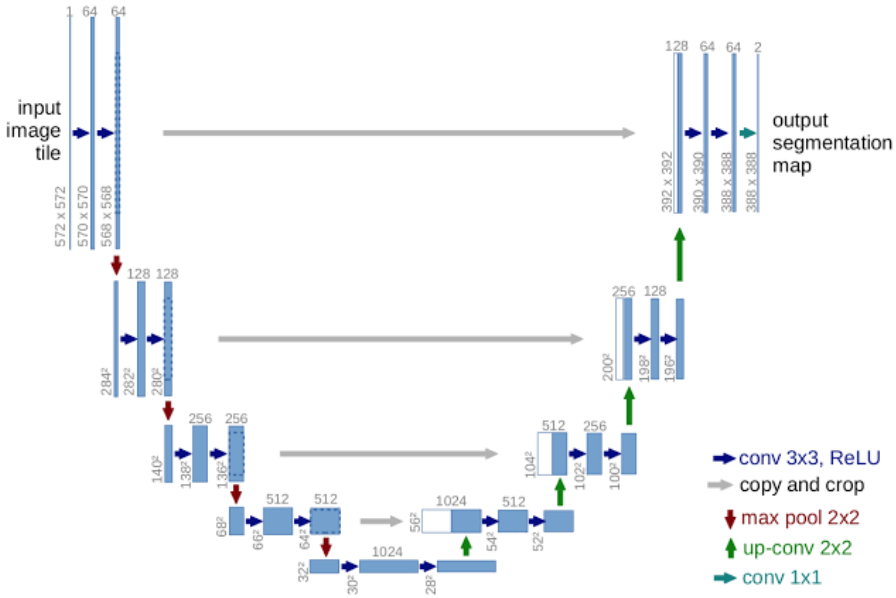


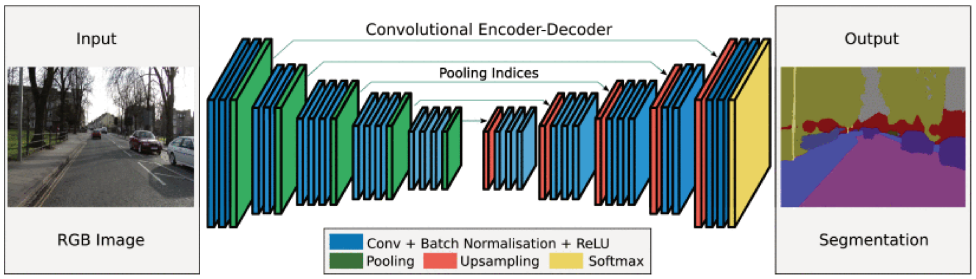
Figure 3 – U-Net, a Fully Convolutional Neural Network architecture, originally developed for biomedical imaging segmentation. Reproduced from (RONNEBERGER *et al.*, 2015).

2.2 Medical Imaging Segmentation with Deep Learning

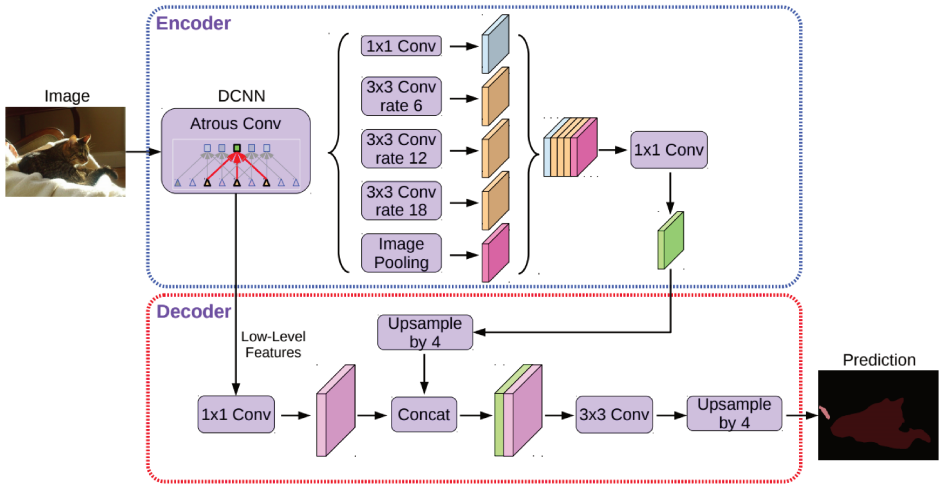
Automatic image segmentation is widely researched in the medical imaging field. Volumetry and shape of organs is of interest to medical research and automated methods can help when analysing a large amount of data, where manual labelling would be too time consuming. In many recent publications, general semantic segmentation FCNN architectures are modified and applied to the medical imaging segmentation field, with great success (KAMNITSAS *et al.*, 2017).

One of the most common approaches to medical imaging segmentation is the adaptation of the famous U-Net CNN architecture (Figure 3). The U-Net seems to facilitate learning for relatively smaller datasets (RONNEBERGER *et al.*, 2015), which is commonly the case in the medical imaging segmentation. It has applications from cell counting (FALK *et al.*, 2019) to brain structure segmentation (MEHTA; SIVASWAMY, 2017), pancreas segmentation (OKTAY *et al.*, 2018), brain tumor segmentation and classification (ISENSEE *et al.*, 2017) and hippocampus segmentation (CARMO *et al.*, 2019a). Most of these works add something to the original architecture in attempts to improve it, but the basic concept of encoder-decoder with concatenation of features remains the same.

Other successful FCNN architectures exist and are also applied to the field, such as DeepLab and SegNet (Figure 4). Although they are different in comparison to the U-Net, the encoder-decoder concept is still present. An example is DeepLab being applied alongside a Long-Short Term Memory Recurrent Neural Network (SUNDERMEYER *et*



(a)



(a)

Figure 4 – (a) SegNet and (b) DeepLab_v3+ encoder-decoder architectures for semantic segmentation. Reproduced from (BADRINARAYANAN *et al.*, 2017) and (CHEN *et al.*, 2018), respectively.

al., 2012) to perform segmentation of colorectal polyps (XIAO *et al.*, 2018). Another example is SegNet being used for gland segmentation in colon cancer images (TANG *et al.*, 2018). This suggests that advancements are still possible in the medical imaging segmentation field, concurrently with new findings on semantic segmentation as a whole. Although currently powerful, the U-Net may be surpassed by other completely different architecture in the future.

2.3 Hippocampus Segmentation with Deep Learning

Before the rise of Deep Learning methods in medical imaging segmentation, most hippocampus segmentation methods used some form of optimization of registration and deformation to reference volumes, called atlas(es) (WANG *et al.*, 2013; IGLESIAS; SABUNCU, 2015; PIPITONE *et al.*, 2014; FISCHL, 2012; CHINCARINI *et al.*, 2016; PLATERO; TOBAR, 2017). Even today, medical research uses results from FreeSurfer (FISCHL, 2012), a high impact multiple brain structures segmentation work, available as a software suite. Those atlas-based methods can produce high quality segmentations, how-

ever they can take more than 8 hours in a single volume. Lately, a more time efficient approach appeared in the literature, namely the use of such atlases as training volumes for CNNs. Deep Learning methods can achieve similar or better overlap metrics while predicting results in a matter of seconds per volume (CHEN *et al.*, 2017; XIE; GILLIES, 2018; WACHINGER *et al.*, 2018; THYREAU *et al.*, 2018; ROY *et al.*, 2019; ATALOGLOU *et al.*, 2019; DINSDALE *et al.*, 2019).

Recent literature in hippocampus segmentation with Deep Learning is exploring different architectures, loss functions and overall methodologies for the task. One approach that seems to be common to most works is using a combination of 2D or 3D CNNs, and patches as inputs in the training phase. A diagram of the works discussed here and their relationships is in Figure 5. Note that some works focus on hippocampus segmentation, while others are devoted to segmentation of multiple neuroanatomy. Segmentation performance is often measured with Dice, an overlap metric (SUDRE *et al.*, 2017). Following, a brief summary of each of those works, in chronological order.

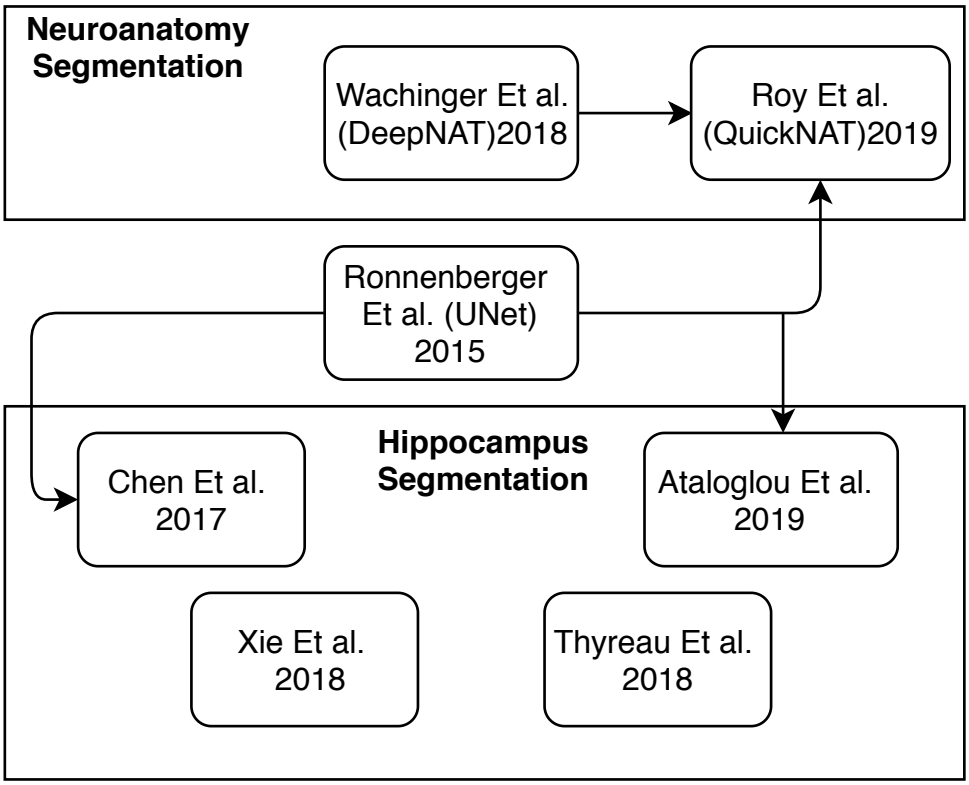


Figure 5 – Papers discussed in this brief literature review. Arrows indicate closely related works.

Chen et al. (CHEN *et al.*, 2017) reports 0.9 Dice (SUDRE *et al.*, 2017) in 10-fold 110 ADNI (PETERSEN *et al.*, 2010) volumes with a novel CNN input idea. Instead of using only the triplanes as patches, it also cuts the volume in six more diagonal orientations. This approach results in 9 planes, that are fed to 9 small modified U-Net (RONNEBERGER *et al.*, 2015) CNNs. The ensemble of these U-Nets constructs the final result.

(XIE; GILLIES, 2018) trains a voxel-wise classification method using triplanar patches crossing the target voxel. They merge features from all patches into a Deep Neural Network with a fully connected classifier alongside standard use of ReLU activations and softmax (KRIZHEVSKY *et al.*, 2012). The training patches come only from the approximate central area where the hippocampus is usually located, balancing labels for 1:1 foreground and background target voxels. Voxel classification methods tend to be faster than multi-atlas methods, but still slower than FCNNs.

DeepNat (WACHINGER *et al.*, 2018) achieves segmentation of 25 structures with a 3D CNN architecture. With a hierarchical approach, a 3D CNN separates foreground from background and another 3D CNN segments the 25 sub-cortical structures on the foreground. Alongside a proposal of a novel parametrization method replacing coordinate augmentation, DeepNat uses 3D Conditional Random Fields as post-processing. The architecture is a voxelwise classification, taking into account the classification of neighbor voxels. This work's results mainly focuses on the Multi-atlas Labeling Challenge dataset, with around 0.86 Dice in hippocampus segmentation.

Hippodeep (THYREAU *et al.*, 2018) uses CNNs trained in a region of interest (ROI). However, where this thesis applies one CNN for each plane of view, Thyreau et al. uses a single CNN, starting with a planar analysis followed by layers of 3D convolutions and shortcut connections. This study used more than 2000 patients, augmented to around 10000 volumes. Initially the model is trained with FreeSurfer segmentations, and later fine tuned using volumes which the author had access to manual segmentations, the gold standard. Thyreau's method requires MNI152 registration of input data, which adds around a minute of computation time, but the model is generally faster than multi-atlas or voxel-wise classification, achieving generalization in different datasets, as verified in (NOGOVITSYN *et al.*, 2019).

Quicknat (ROY *et al.*, 2019) and Atalaglou's method (ATALOGLOU *et al.*, 2019) are simultaneous, independent works, that used a similar idea to this thesis, namely the use of the consensus of a CNN per orthogonal view of the MRI volume.

QuickNat achieves faster segmentations than DeepNat by using a multiple CNN approach instead of voxel-wise classification. Its methodology follows a consensus of multiple 2D U-Net like architectures specialized in each slice orientation. The use of FreeSurfer (FISCHL, 2012) masks over hundreds of public data to generate silver standard annotations allows for much more data than usually available for medical imaging. Later, after the network already knows to localize the structures, it is fine-tuned to more precise gold standard labels. Inputs for this method need to conform to the FreeSurfer format.

Ataloglou et al. recently displayed another case of fusion of multiple CNN outputs, specialized into axial, coronal and sagittal orientations, into a final hippocampus

segmentation. They used U-Net like CNNs specialized in each orientation, followed by error correction CNNs, and a final average fusion of the results. They went against a common approach in training U-Nets of using patches during data augmentation, instead using cropped slices. This raises concerns about overfitting to the used dataset, HarP (BOCCARDI *et al.*, 2015), supported by the need of fine-tuning to generalize to a different dataset. Inputs for this method need to conform, manually, to a specific orientation, alongside going through a pre-processing pipeline with brain extraction and non-parametric non-uniform intensity normalisation (N3).

Dinsdale et al. (DINSDALE *et al.*, 2019) mixes knowledge from multi-atlas works with Deep Learning, by using a 3D U-Net CNN to predict a deformation field from an initial binary sphere to the segmentation of the hippocampus, achieving around 0.86 DICE on Harp. Trying an auxiliary classification task did not improve segmentation results.

It is known that Deep Learning approaches require a large amount of training data, something that is not commonly available specially with Medical Imaging. Commonly used forms of increasing the quantity of data in the literature include using 2D CNNs over regions (patches) of slices, with some form of patch selection strategy. The U-Net (RONNEBERGER *et al.*, 2015) FCNN architecture has shown potential to learn from relatively small amounts of data with their decoding, encoding and concatenation schemes, even working when used with 3D convolutions directly in a 3D volume (ISENSEE *et al.*, 2017).

Looking at these recent works, one can confirm the segmentation potential of the U-Net architecture, including the idea of an ensemble of 2D U-Nets instead of using a single 3D one, as an author work with colleagues (CARMO *et al.*, 2019b), some simultaneous recent work (ROY *et al.*, 2019; ATALOGLOU *et al.*, 2019), or even works in other segmentation problems (LUCENA *et al.*, 2018) presented. In this thesis, some of those methods were locally reproduced for comparison purposes in our final tests, namely (ROY *et al.*, 2019; THYREAU *et al.*, 2018), including a 3D architecture test from (ISENSEE *et al.*, 2017).

3 Materials

In this chapter, the involved datasets are presented in more detail. Also, implementation details are disclaimed.

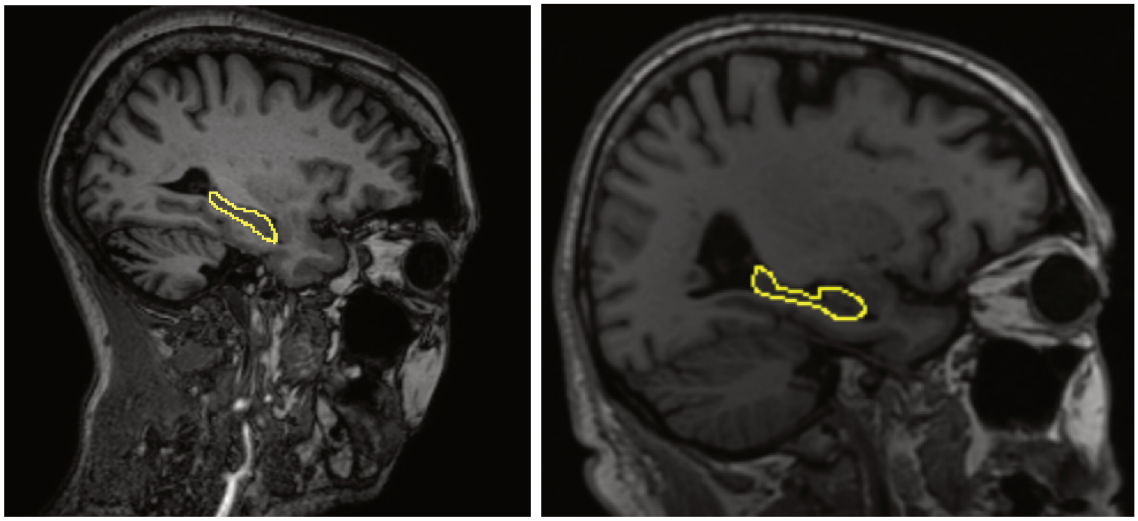
3.1 Data

This thesis uses two different datasets: one collected locally for an epilepsy study, named HCUnicamp; and a public one from the ADNI Alzheimer's study, HarP (BOCCARDI *et al.*, 2015). HarP is commonly used in the literature as a hippocampus segmentation benchmark. The main difference between the datasets is, the lack of one of the hippocampi in 70% of the epilepsy scans from HCUnicamp, due to surgical intervention, detailed in (GHIZONI *et al.*, 2015). Meanwhile, HarP has presence of atrophies due to Alzheimer's disease. Both datasets have control subjects. Both datasets also have scans from control groups.

Some initial experiments were performed with MNI152 (BRETT *et al.*, 2001) registered volumes from HCUnicamp and volbrain (MANJÓN; COUPÉ, 2016) silver standard masks. Our method needs input data to be in the MNI152 head orientation. In most experiments, data from those datasets is in native space and is not registered besides orientation correction. Due to that, when predicting in external volumes, an orientation correction by rigid registration is provided as an option, to avoid orientation mismatch problems.

3.1.1 HarP

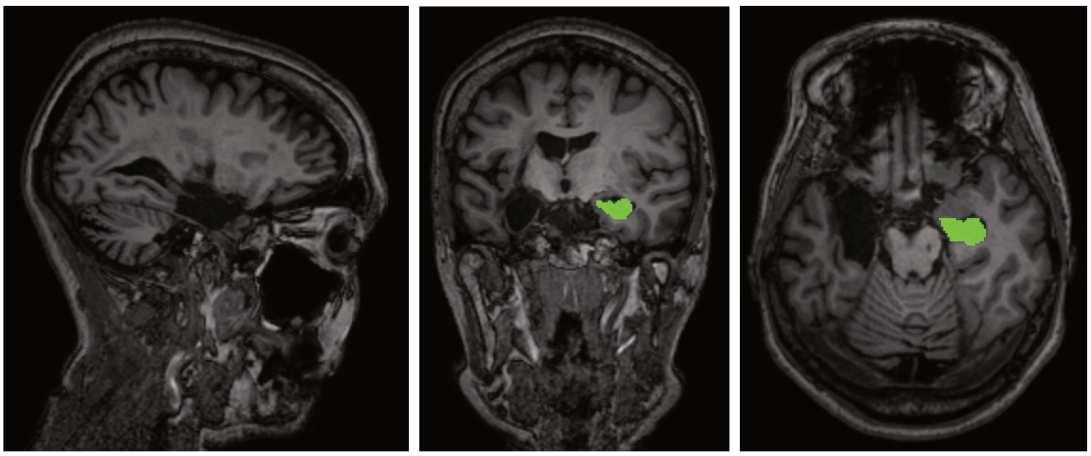
HarP (BOCCARDI *et al.*, 2015) is a widely used benchmark dataset in the hippocampus segmentation literature. The full HarP release contains 135 T1-weighted MRI volumes. HarP uses data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (<adni.loni.usc.edu>). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Alzheimer's disease classes are balanced with equal occurrence of CN, MCI and AD cases (PETERSEN *et al.*, 2010). The original volumes were minmax intensity normalized between 0 and 1, and no volumes were removed. When using this dataset for training, hold-out was employed with 70% training, 10%



(a) (b)

Figure 6 – Sagittal slice sample from a random control subject, with manual annotation's border in yellow, from (a) HCUnicamp and (b) HarP.

validation and 20% testing set. K-Folds cross validation, when used, consisted of 5 folds, with no overlap on the test sets.



(a) (b) (c)

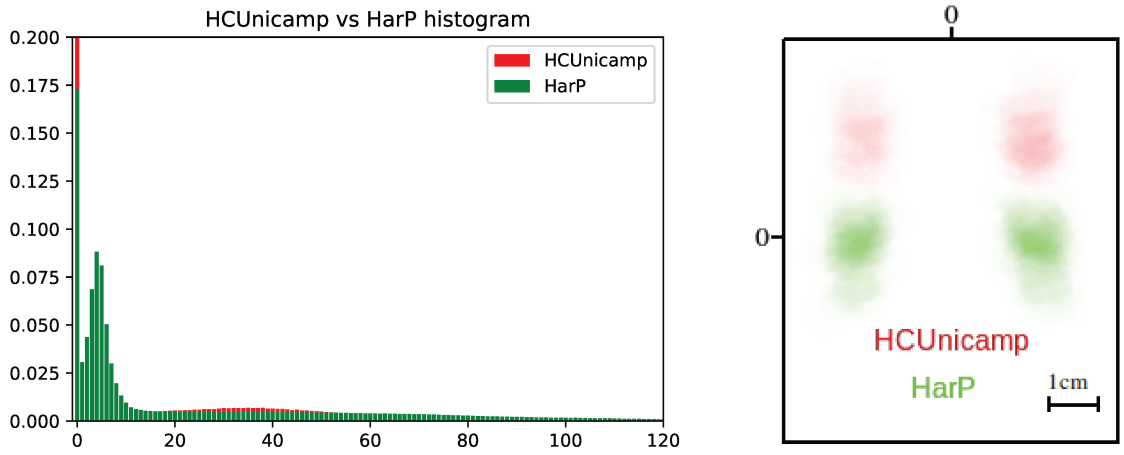
Figure 7 – (a) Sagittal, (b) Coronal and (c) Axial HCUnicamp slices from a post-operative scan, where one of the hippocampus was removed. Manual annotations in green.

3.1.2 HCUnicamp

HCUnicamp was collected inhouse, by personnel from the Brazilian Institute of Neuroscience and Neurotechnology (BRAINN) in UNICAMP's *Hospital de Clínicas*. This dataset contains 190 T1-weighted 3T MRI acquisitions, in native space. 58 are from

healthy controls and 132 are from epilepsy patients. From the epilepsy scans, 70% had one of the hippocampus surgically removed, resulting in a very different shape and texture than what is commonly seen in public datasets (Figure 7). More details about the surgical procedure can be found in (GHIZONI *et al.*, 2015; GHIZONI *et al.*, 2017). All volumes have manual annotations of the hippocampus, done by one rater. Post-processing includes voxel intensity normalization with min-max, between 0 and 1, per volume. This data acquisition was approved by the Ethics and Research Committee, while the specific use for this research is also approved, under CEP 3435027. Comparisons between the datasets can be seen in Figure 6 and Figure 8. The difference in mean mask position due to the inclusion of neck in HCUnicamp is notable, alongside with the lower presence of left hippocampus labels (Figure 8(b)) due to surgical intervention for Epilepsy. Moreover, the intensity histogram in Figure 8(a) shows background noise is very present in HarP, while HCUnicamp has mostly black backgrounds.

HCUnicamp is mainly used with the intention to see how hippocampus segmentation methods deal with the missing hippocampus and the presence of different textures. The native volumes with their manual annotations were only used in this thesis's final experiments, and not taken into consideration for the method's methodological choices. A final experiment in an attempt to learn from the Epilepsy data divides HCUnicamp in a balanced hold-out, which is called HCUnicamp-H for clarity.



(a) (b)

Figure 8 – (a) Shows a normalized intensity histogram for all volumes in both datasets. Differences due to presence of noisy background in HarP and neck present in the field of view of HCUnicamp are notable. In (b), a coronal center crop slice of the average hippocampus mask for HarP (in green) and HCUnicamp (in red). Zero corresponds to the center.

3.1.3 MNI-HCUnicamp

MNI152 space registered volumes of HCUnicamp with volbrain hippocampus annotations (MANJÓN; COUPÉ, 2016) were used in early experiments. After conformity pre-processings, the scans look quite different to the native ones and more normalized (Figure 9). Hold-out was employed with 80% of the data for training, 10% validation and 10% testing sets.

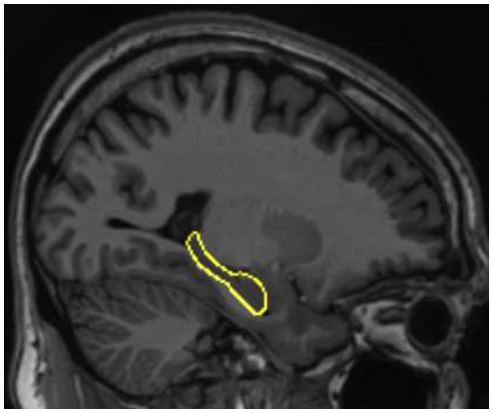


Figure 9 – Sagittal slice from MNI-HCUnicamp, with FreeSurfer segmentations borders as targets, in yellow.

3.2 Implementation Details

This work implements all discussed methodology and data usage in a Python 3.6 environment running in Ubuntu 18.04. Deep Learning tasks are performed using the PyTorch 1.x library. Other computer vision and machine learning tasks involved the sporadic use of other scientific libraries such as SciPy, scikit-image, scikit-learn, OpenCV, Numpy and matplotlib. NiBabel was used to handle MRI data. The ITKSnap tool (YUSHKEVICH *et al.*, 2006) was used for producing 3D and multi-view visualizations. Additional software was used when running other methods from the literature.

The hardware used included a E3-1220 v3 CPU, NVIDIA Titan X 12GB GPU and 32 GB of RAM.

For the use of this code by other researchers, a simple GUI and CLI were implemented in the public release of this thesis code, with easy to follow instructions (<github.com/MICLab-Unicamp/e2dhipseg>). A binary release version is also included.

4 Method

In this chapter, the methodology (Figure 10) for our hippocampus segmentation method is detailed. For hippocampus segmentation, the input is a volumetric MRI image of a subject, and the desired output is a volumetric segmentation mask.

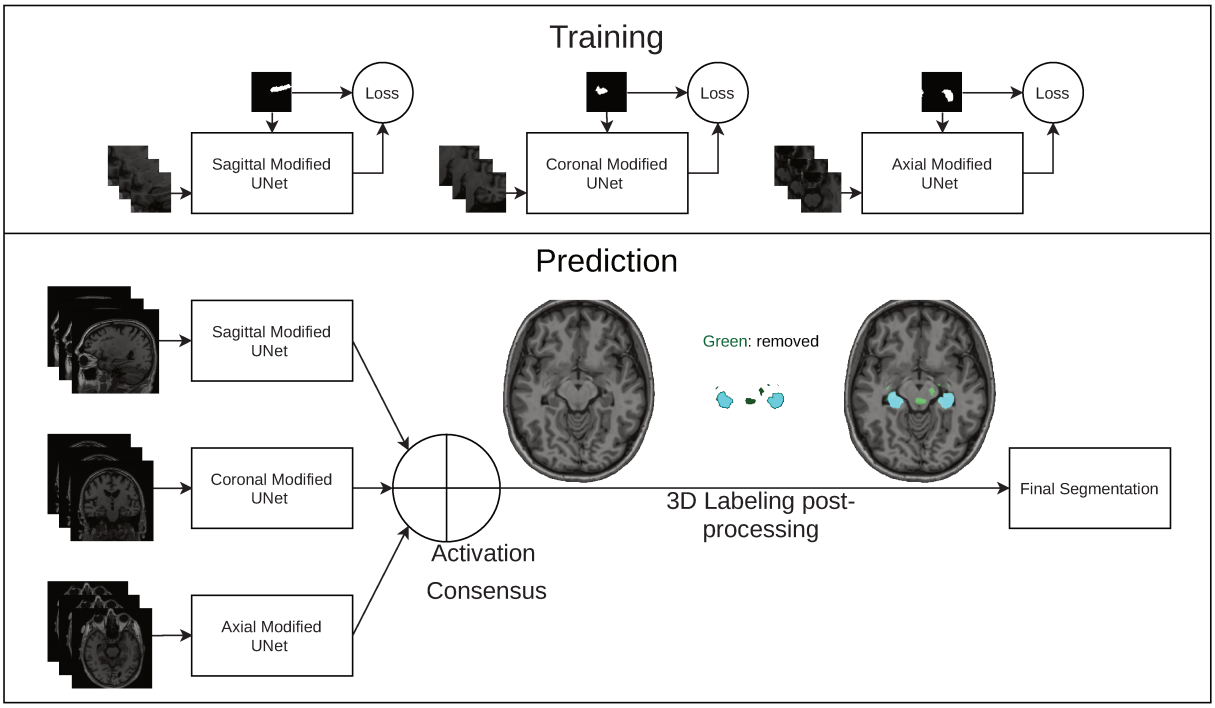


Figure 10 – The final segmentation volume is generated by taking into account activations from three FCNNs specialized on each 2D orientation. Neighboring slices are taken into account in a multi-channel approach. Full slices are used in prediction time, but training uses patches and their respective targets.

In summary, activations from three orientation specialized, 2D, modified U-Net CNNs are merged into an activation consensus. This approach is inspired by the work of (LUCENA *et al.*, 2018). Each network's activations for a given input volume are built, slice by slice. The three activation volumes, consisting of stacked 2D activations, are then averaged into a consensus volume, which is post-processed into the final segmentation mask. This methodology is inspired by how physicians analyze MRI in multiview visualization. Most experiments in this thesis consisted of adjusting the inner works of these CNNs and training parameters. All three networks follow the same architecture and hyperparameters, but are trained separately.

The following sections go into more detail in each part of the deep network's architecture and the method overall. Some ideas that didn't make it to the final method are

included. Experimental results will be presented in chapter 5, defining the final parameters of the methodology.

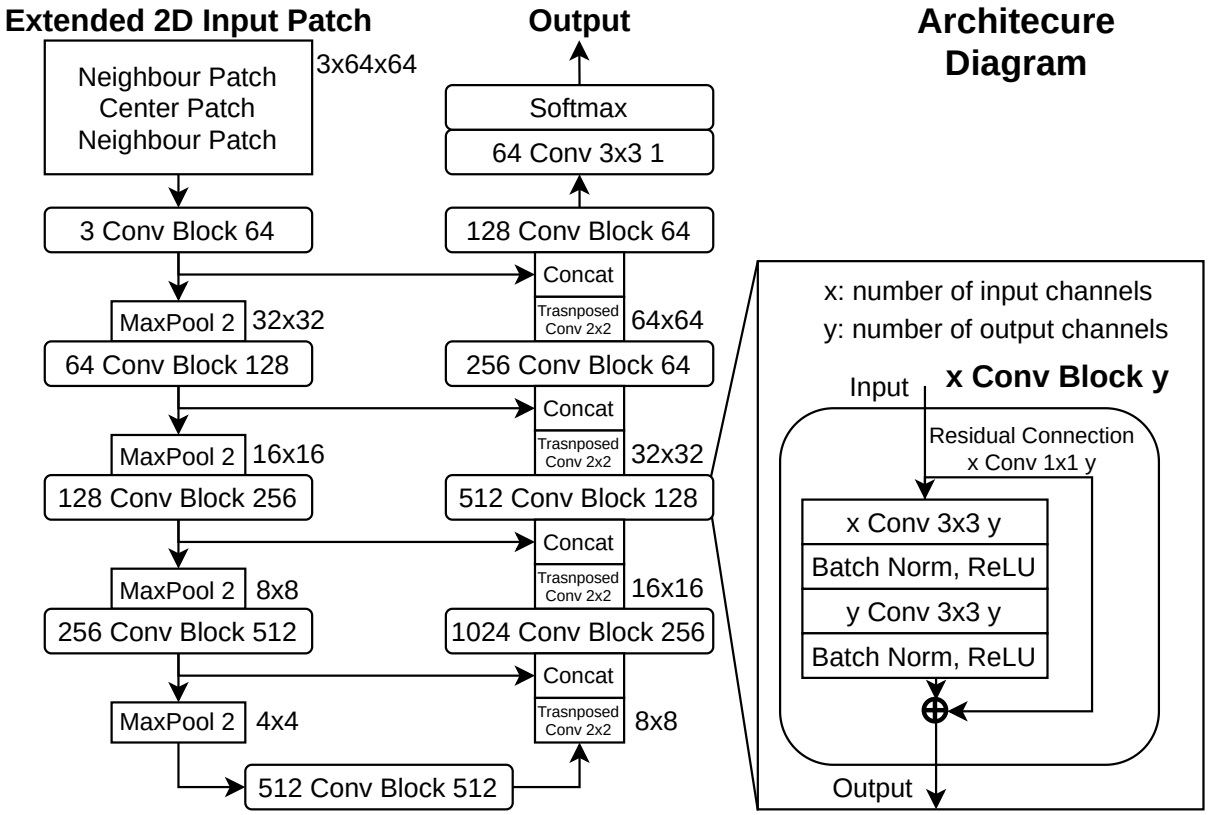


Figure 11 – Final architecture of each modified U-Net in figure 10. Of note in comparison to the original U-Net is the use of BatchNorm, residual connections in each convolutional block and the 3 channel neighbour patches input. Padding is also used after convolutions. In this work, the output might use a Softmax or the depicted Sigmoid layer. Spatial resolution changes are noted near to max pools or transposed convolutions.

4.1 U-Net architecture

The basic structure of each of the three networks depicted in Figure 10 is inspired by the 2D U-Net FCNN architecture (RONNEBERGER *et al.*, 2015). Note that in this work, 2D network refers to the input being 2D for the sake of visualization (as in a slice of a brain or a photo of an object), but considering the usage of channels and batches, the input will be technically 4D. In the same vein, a 3D network would process batches of volumes with possibly multiple channels, and the input would be 5D. The same idea is used when referring to 2D or 3D convolutions.

The U-Net architecture presents a U-shaped pattern where a step down is a series of two convolutional layers followed by a downsampling layer and a step up consists in a

series of two convolutional layers followed by upsampling. Connections are made between the downsample and upsample path at each scale, with concatenation of weights. Note that these downsample and upsample paths are also called encoder and decoder, respectively, in the literature, but they are not the same as encoders in e.g: auto encoders (BALDI, 2012). In this thesis some modifications based on other successful works were applied to the architecture (Figure 11). Those modifications include: instead of one single 2D patch as input, two neighbour patches are concatenated leaving the patch corresponding to the target mask in the center (PEREIRA *et al.*, 2019). Residual connections based on ResNet (HE *et al.*, 2016) between the input and output of the double convolutional block were added, as 1x1 2D convolutions to account for different number of channels. Batch normalization was added to each convolution inside the convolutional block, to accelerate convergence and facilitate learning (IOFFE; SZEGEDY, 2015). Also, all convolutions use padding to keep spatial dimensions and have no bias.

4.1.1 Residual Connections

Residual or shortcut connections have been shown to improve convergence and performance of CNNs (HE *et al.*, 2016). Either in the form of direct connections propagating past results to the next convolution input, by adding values, or in the form of 1x1 convolutions, to deal with different number of channels. An argument to its effectiveness is that the residual connections offer a way for a simpler propagation of values without any significant transformation. This is not a trivial task when the network consists of multiple non linear transformations, in the form of convolutions with non linear activations followed by max pooling.

In this work, residual connections were implemented in the form of an 1x1 convolution, adding the input of the first 3x3 convolution to the result of the batch normalization of the second 3x3 convolution in a convolutional block (Conv Block in Figure 11).

4.1.2 Weight Initialization, Bias and Batch-normalization

It has been shown that weight initialization is crucial in proper convergence of CNNs (KUMAR, 2017). In computer vision related tasks, having pre-initialized weights that already recognize basic image pattern recognition features such as border directions, frequencies and textures can be helpful. This work uses VGG11 (SIMONYAN; ZISSERMAN, 2014) weights, pre-trained on ImageNet, in the encoder part of the U-Net architecture, as in (IGLOVIKOV; SHVETS, 2018). Using initial layers of a pre-trained ResNet34 (HE *et al.*, 2016) as an encoder, or using Kaiming Uniform initialization (HE *et al.*, 2015) were also attempted.

4.2 Network Input

4.2.1 Prediction Input

During prediction time, slices for each network are extracted with a center crop, following each network's orientation. When building the output mask activation volume, the resulting activations are padded back to the original input size, and concatenated to result in a volumetric mask. In other words, let a given volumetric MRI input $I(x, y, z)$, where Z is the number of slices in the network's orientation. Let each network be represented by N_o where o is the orientation of such network: coronal c, sagittal s or axial a. The evaluation of volume I will be split in Z predictions of each slice $[N_o(I(x, y, 0)), N_o(I(x, y, 1)), ..., N_o(I(x, y, Z))]$. By concatenating those predictions on the z axis, one would have a volumetric activation mask M_o for network N_o where $M_o = f(x, y, z)$.

4.2.2 Training Input

For training, this method uses patches. One of the strong fits of the U-Net architecture is its ability to learn on patches and extend that knowledge to the evaluation of a full image, effectively working as a form of data augmentation. In this work, batches of random patches are used when training each network. Patches are selected in runtime, not as pre-processing. Patches can achieve many possible sizes, as long as it accommodates the number of spatial resolution reductions present in the network, e.g. division by 2 by a max pool or stride 2 convolutions.

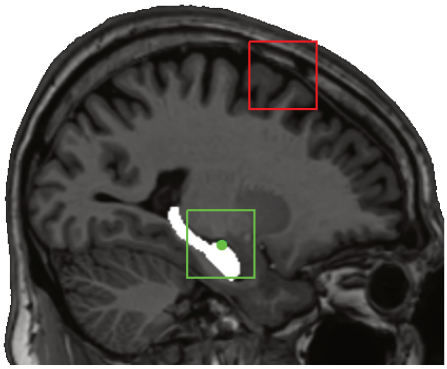
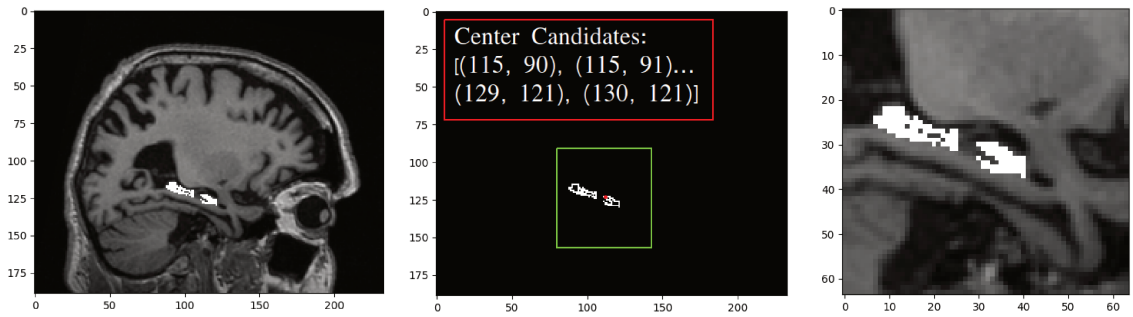


Figure 12 – In green, a positive patch, centered in a random point of the hippocampus border. In red, a random patch, named here negative patch. In this example patches have 32x32 mm. The hippocampus is highlighted with its target.

Patches are divided in negative and positive patches (Figure 12). Negative patches are selected from a random point of the brain, allowing for learning of what structures are not the hippocampus, and are not close to the structure, such as scalp, neck, eyes

and brain ridges. They do not necessarily have a completely zeroed target due to being random. On the other hand, positive patches are always centered on a random point of the hippocampus border. The hippocampus border is calculated using Canny edge detection (CHEN, 2015) over the binary target. The random center point for the patch is then selected by a random choice from the sparse matrix of the hippocampus border (Figure 13).



(a) (b) (c)

Figure 13 – Positive patch selection steps. (a) Overlap of sagittal slice with its target. (b) Target borders, with the red point being the selected patch center. (c) Resulting 64x64 positive patch.

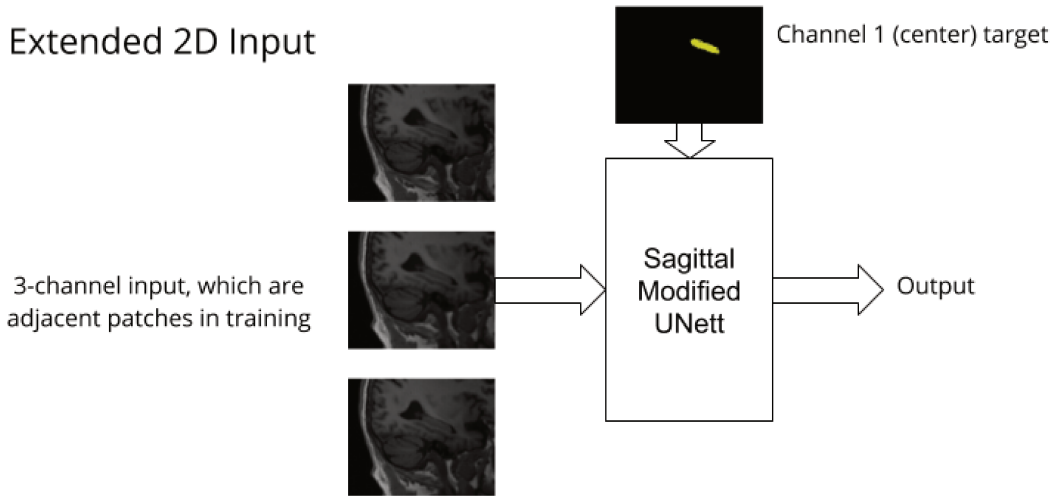


Figure 14 – Sample of Extended 2D input and target, when training with sagittal patches.

4.2.3 Extended 2D

In a similar approach to (PEREIRA *et al.*, 2019)'s Extended 2D, adjacent patches (slices on evaluation) are included in the network's input as additional channels (Figure 14). The intention is for the 2D network to take into consideration volumetric in-

formation adjacent to the region of interest, hence the name for the method, Extended 2D Consensus Hippocampus Segmentation (E2DHipseg). This methodological choice is also inspired by how physicians compare neighbor slices in multiview visualization, when deciding if a voxel is part of the analyzed structure or not.

4.3 Data Augmentation

Deep Learning algorithms usually require a big and varied dataset to achieve generalization (SHIN *et al.*, 2016). Manual segmentation by experts is used as a gold standard, but is often not enough for the training of Deep Networks. Data augmentation is used to try and improve our dataset variance and avoid overfitting, an excessive bias to the training data. Without augmentation, this method could overfit to MRI machine parameters such as magnetic field intensity, field of view and so on, or overfit to any bias present in the training dataset in hippocampus shape and size. All augmentations perform a random small modification to the selected patches, according to pre-defined parameters. Patches are augmented on runtime, not as pre-processing. A variety of combinations of data transformations were tested in this thesis for data augmentation purposes.

Let a patch $P(c, x, y)$ be a selected E2D patch, with $\{c \in Z : 0 \leq c < 3\}$ and $T(x, y)$ be the respective target. x and y represent rows and columns respectively. Since transformations to patches are applied to all c E2D channels, for the sake of simplicity, patches will be represented as $P(x, y)$. Additionally, the omission of (x, y) denotes an element wise operation and $P[C]$ denotes an element wise operation applied when condition C is true. After all transformations are applied, as a final step, values above 1 or below 0 are clipped, following $P[P > 1] = 1$ and $P[P < 0] = 0$. Following, a more detailed description of each transformation employed for data augmentation, with visualizations in Figure 15.

4.3.1 Intensity Transformation

There is a difference on voxel intensity on different scans, coming from different MR machines and configurations. Addition of intensity transformed data attempts to simulate this variation in voxel intensity.

This transformation takes as an argument a value $a \in [0, 1]$, to define the range of intensity modification. Let $\upsilon(-a, a)$ be a uniform distribution with values in the range $[-a, a]$, the intensity transformation applies to a patch P as:

$$P = P + i \tag{4.1}$$

Where $i \in \upsilon(-a, a)$ is a constant for every pixel.

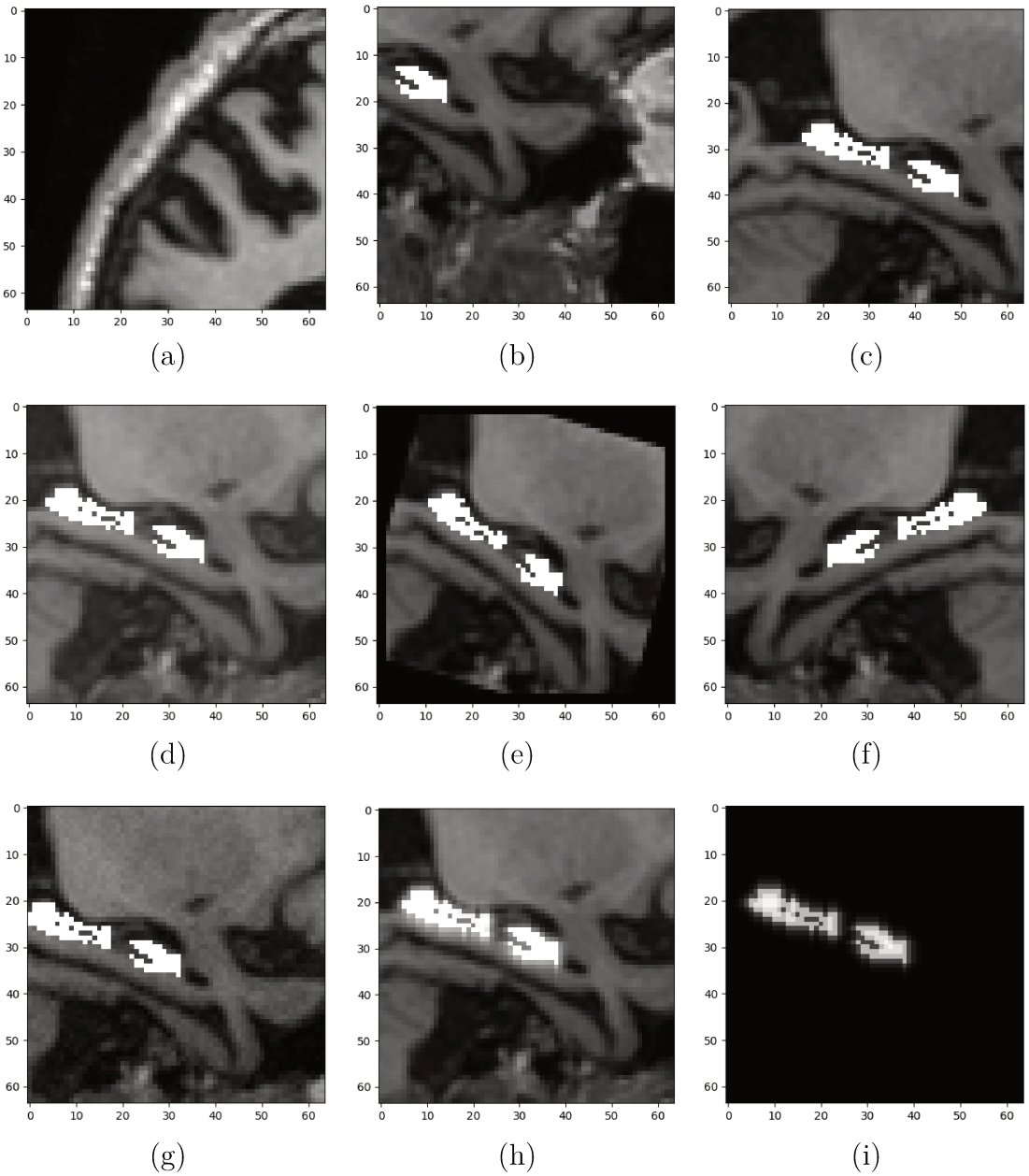


Figure 15 – Visual effects of transformations. All patches are selected from the same slice, but might be centered on a different point. The segmentation target is highlighted in white. (a) and (b) are negative patches, with (b) showcasing the possibility of hippocampus presence. A positive patch without transformations is in (c), and other transformations are: (d) intensity with $i = 0.1$; (e) rotation and scale with $r = -16$ and $s = 0.94$; (f) horizontal flip; (g) gaussian noise with $v = 0.0002$ and $\mu = 0$; soft target overlap (h) and mask (i) with $\lambda = 1$.

4.3.2 Rotation and Scale

The reason to use data augmentation with addition of rotated and scaled data is an attempt to simulate small rotations and differences on size of the hippocampus, among different subjects. This also includes small head orientation rotations due to subject position on the MR scanner.

This transformation could also be called an Affine Transformation. In a similar fashion to the intensity transform, given an input argument b, let $\upsilon(-b, b)$ be a uniform distribution with values in the range $[-b, b]$, the patch P is rotated by $r \in \upsilon(-b, b)$ or scaled by a factor of $s \in \upsilon(-b, b)$, both in the x and y axis. With P' being the transformed patch, pixel relocation could be expressed by the Affine Transformation $P' = AP + c$, with the augmented matrix A taking the form of:

$$A = \begin{bmatrix} s\cos r & -\sin r & 0 \\ \sin r & s\cos r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

The same operation is performed on the correspondent target T, using the same r and s. Intensity interpolation in patches is bicubic, while targets use nearest neighbor interpolation, to not generate intensity values different from 0 or 1. When the scale operations results in a zommed out patch, new pixels are filled with a symmetric strategy, mirroring the same pixels already present in the border.

4.3.3 Flips

Flips add additional synthetic data, exploring the horizontal symmetry of the hippocampus. The use of vertical flips would be useful in the sense of being able to recognize scans where the head is rotated, useful if global rotation invariance is intended for the model.

Consider $Ph(x, y)$ the result of a horizontal flip. Let X and Y represent the maximum number of rows and columns, respectively, in a patch. $Ph(x, y)$ can be expressed as:

$$Ph(x, y) = P(x, Y - y) \tag{4.3}$$

In a similar fashion, a vertical flip $Pv(x, y)$ can be expressed as:

$$Pv(x, y) = P(X - x, y) \tag{4.4}$$

4.3.4 Gaussian Noise

1.5T MR acquisitions generally appear more noisy and lower quality than 3T, although 3T images present more high frequency noise (SOHER *et al.*, 2007). Data augmentation with Gaussian Noise addition is an attempt to simulate the presence of lower quality scans, and make the networks more immune to noise.

Given a patch shape of X rows and Y columns, a noise matrix M corresponding to the same shape is generated, with all values z randomly selected from a Gaussian distribution G, with the following probability-density function (CHEN, 2015):

$$G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{4.5}$$

With mean μ, variance v, and standard deviation $\sigma = \sqrt{v}$. The noisy patch Pn is then generated following:

$$Pn(x, y) = P(x, y) + M(x, y) \tag{4.6}$$

4.3.5 Soft Target

Inter-rater variability is a important problem in segmentation applications. Disagreement between experts is common, specially in more complex structures such as the Hippocampus. Machine learning models learn the segmentation strategy of the rater who produced the ground-truth data, which might not agree with a different protocol present in, e.g. a different dataset (SOUZA *et al.*, 2018).

Segmentation masks are usually represented by binary masks. To express the uncertainty around the binary masks's border, this thesis proposes a transformation to the target mask, called soft target. The intention is to have a soft slope around the segmentation border. Instead of an abrupt change from 0 to 1, pixels approaching the border from the outside increase from 0 to 0.5 in the border, and from 0.5 to 1.0 from the border towards the center of the mask. To be able to determine the distances to the border, a $L1$ distance transform is used. The $L1$ distance from point $P_1(x_1, y_1)$ to $P_2(x_2, y_2)$ can be defined as $L1 = |x_1 - x_2| + |y_1 - y_2|$. Assume the operator D outputs the L1 distance map from every pixel with value 0 to the nearest pixel with value 1. Therefore, the soft target Ts can be defined as:

$$Ts = S(\lambda(D(1 - T) - D(T))) \tag{4.7}$$

$$S(x) = \frac{1}{1 + e^{-x}} \tag{4.8}$$

Where S is a sigmoid operator, applying the sigmoid function $S(x)$ (Figure 16) in a pixel-wise manner. The rate in which the values change, or the “softness" can be defined by the λ parameter, with a lower λ resulting in a softer slope.

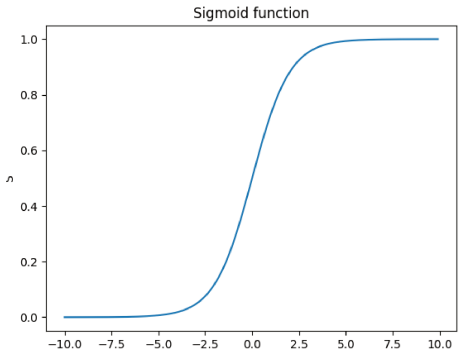


Figure 16 – Plot of $S(x)$. There is no need to clip intervals since the sigmoid output is already limited between 0 and 1. 0.5 ends up corresponding to 0 distance from the mask's border.

4.4 Network Output

In binary segmentation applications, two commonly used outputs for CNNs are a sigmoid layer, or a softmax layer. Consider a 2D segmentation case. The sigmoid layer consists of taking the last activation of the CNN and performing an element wise sigmoid, following equation 4.8. In this case, the output $O(x, y)$ has one channel, where the value 1 represents the foreground and 0 the background, essentially representing the probability of foreground. On the other hand, if the output has two channels with $O(c, x, y)$ and $c \in [0, 1]$, each channel can represent the foreground and background activations separately. The softmax layer ensures that each position (x, y) sums up to 1.0 over c, resulting in a probabilistic value for each channel. For a binary segmentation case, the softmax S_F can be calculated, for output O with:

$$S_F(c, x, y) = \frac{e^{O(c,x,y)}}{\sum_{i=0}^{2} e^{O(i,x,y)}} \tag{4.9}$$

The Hippocampus Segmentation problem has as an input a 3D volume, and a target a 3D mask. So far, most of the discussion of this thesis has focused on 2D patches and 2D CNNs. To extend the 2D problem to the original 3D one, each network N_o evaluates the volumetric input slice by slice, concatenating the output into a volumetric hippocampus activation mask M_o, for orientation o.

Recall that the 2D coronal network N_c outputs a 2D slice mask $O_c = f(x, y)$. To build the volumetric activation mask $M_c = f(x, y, z)$, the volumetric input I is evaluated slice by slice, as in: $M_c(x, y, 0) = N_c(I(x, y, 0))$, $M_c(x, y, 1) = N_c(I(x, y, 1))$, ...,

$M_c(x, y, Z) = N_c(I(x, y, Z))$. Note that M_c is still not a binary mask, and consists of Softmax or Sigmoid activations between 0 and 1. The final binary hippocampus mask M is only produced after the consensus and post-processing phase.

4.5 Loss Functions

A loss function when training a neural network is a function that penalizes the model for outputs that do not conform to the training set. It plays an important role in the optimization process, being what guides the steps taken by the optimizer (BENGIO *et al.*, 2015). In all described loss functions, consider $o_i \in O$ the output elements and $t_i \in T$ the target elements, for N pixels in 2D or voxels in 3D. Note that, when training, the batch can be composed of many patches. In that case, loss for that batch is calculated as the mean of the loss for each patch. Following are descriptions of loss functions used on this thesis.

4.5.1 MSE and BCE

When using a sigmoid output activation, Binary Cross Entropy (BCE), Mean Square Error (MSE) and Dice Loss are examples of commonly used functions in the literature.

MSE measures the average error between the output and the target by summing element-wise square of the differences between both.

$$MSE = \frac{\sum_i^N ((o_i - t_i)^2)}{N} \tag{4.10}$$

BCE is a special case of the Cross Entropy (BENGIO *et al.*, 2015), where the target is binary, and can be expressed as a single activation from 0 to 1, as in the sigmoid activation. BCE is calculated with:

$$BCE = \frac{\sum_i^N (t_i \log o_i + (1 - t_i) \log (1 - o_i))}{N} \tag{4.11}$$

4.5.2 Dice and Dice Loss

Dice (SUDRE *et al.*, 2017) is an overlap metric widely used in the evaluation of segmentation applications. Segmentation performance in this thesis is evaluated with Dice, by comparisons with the manual gold standard ground truth. Dice can be defined as:

$$Dice = 2\frac{\sum_i^N p_i t_i}{\sum_i^N p_i^2 + \sum_i^N t_i^2} \tag{4.12}$$

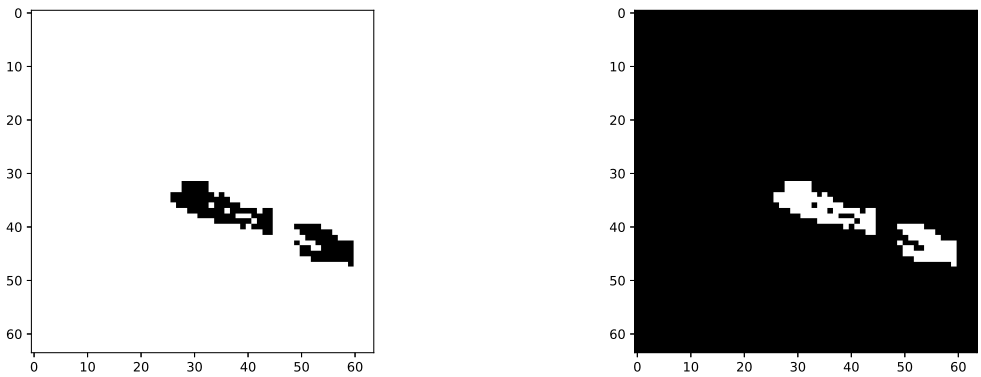
Where p and t are binary voxels from the prediction and target volume. To use Dice as a loss function, one can simply optimize $DiceLoss = 1 - Dice$, therefore optimizing a segmentation overlap metric.

$$DiceLoss = 1 - 2\frac{\sum_i^N o_i t_i}{\sum_i^N o_i^2 + \sum_i^N t_i^2} \tag{4.13}$$

Here, the binary p is replaced by the probabilistic sigmoid output of the network, o. DiceLoss works over probabilistic values from the output, while the metric uses strictly 0 and 1 binary values. This allows for optimization of a probabilistic output from the network, helping in early convergence and smoothness while training. When used as a Loss function in this work, O and T are 2D.

4.5.3 GDL and Boundary Loss

To take into account background information, a Softmax of two-channels representing background and foreground is used as an output (Figure 17). However, in many segmentation applications, there is an unbalance between foreground and background labels, requiring some form of weighting to the less represented label. One recently proposed function that satisfies that requirement is the Generalized Dice Loss (GDL) (SUDRE *et al.*, 2017).



(a) (b)

Figure 17 – A visual representation of (a) background target in channel 0 and (b) foreground target in channel 1, in a Softmax target. 1 is white and 0 is black. Note that summing both channels would sum to 1 in every pixel.

GDL weights the loss value by the presence of a given label in the target, giving more importance to less present labels. This solves a class imbalance problem that would emerge when using Dice Loss while including background as a class. Let t^c represents channel c in a $T(c, x, y)$ softmax target. Considering w_c the inverse contribution of channel

c in the evaluated slice or volume, defined by $w_c = 1/(\sum_i^N t_{ci})^2$, GDL is defined by, for the two channel case:

$$GDL = 1 - 2\frac{\sum_{c=0}^{1} w_c \sum_i^N t_{ci} o_{ci}}{\sum_{c=0}^{1} w_c \sum_i^N t_{ci} + o_{ci}} \tag{4.14}$$

In this case, o_c is channel c of the softmax output, where o_0 would be in practice the output softmax probabilities for the background and o_1 the foreground (hippocampus) channel.

In 2019 an improvement to GDL was proposed in the form of the Boundary Loss (KERVADEC *et al.*, 2019). Kervadec's work suggests that a loss functions that takes into account boundary information can improve results, specially for unbalanced datasets. Boundary Loss improves GDL by considering it a "regional" loss, and adding a second term, named Surface loss. In theory, the surface loss represents the sum of normal distances between the target border and prediction border. However, in a differentiable approximation, the surface loss L_S is defined by the sum over channels c of the element-wise multiplication between: D^c_{Si}, the Euclidian distance map of the target in channel c; and S^c_{Fi}, the softmax probabilistic output in channel c.

$$L_S = \sum_{c=0}^{1} \sum_{i}^{N} D_{Sci} S_{Fci} \tag{4.15}$$

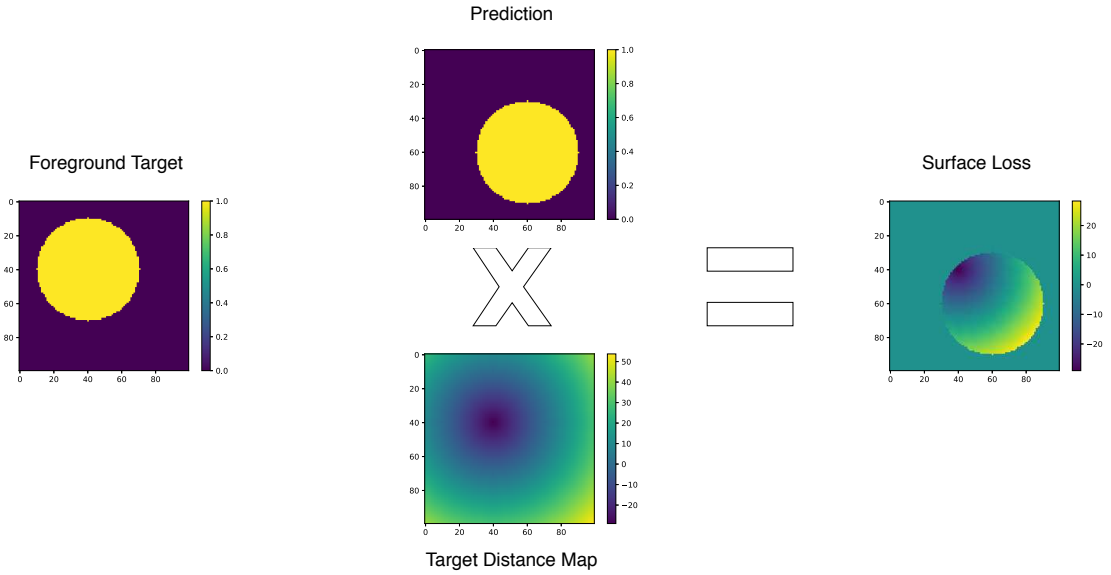


Figure 18 – Simulated Surface Loss, only in the foreground channel. The target's distance map is element-wise multiplied by the corresponding channel in the prediction, in this case, the foreground. Yellow corresponds to high values and purple low values. Notice the output loss has high values where the prediction distances itself from the target.

In the practical implementation, the Surface Loss has as inputs the distance maps to the target, and the corresponding softmax predictions. An illustration of this process in a simulated target, only in the foreground channel, is displayed in Figure 18. Notice that this does not gives weight to the overall area of the target, only to the distance between target and output borders. Consider the case that the prediction is completely inside the target but smaller (undersegmentation). The resulting value from the sum of the Surface Loss would still be higher than a perfectly aligned case, due to the negative values from the distance map when inside the target.

Finally, for the final Boundary Loss equation, it is necessary to balance the contribution of both components with a weight, defined as α in the following Boundary Loss (BDL) equation:

$$BDL = \alpha\ GDL + (1 - \alpha)\ L_S \tag{4.16}$$

Where GDL is the regional component of the loss function, and L_S is the Surface Loss. The weight factor α changes from epoch to epoch. The weight given to the regional loss is shifted to the surface loss, with α varying from 1 in the first epoch to 0 in the last epoch. The intention is to first optimize the localization and area of the target, and in later epochs, optimize the border distances with the Surface Loss. This thesis follows the original implementation in (KERVADEC *et al.*, 2019), where more detail can be found on the deduction of L_S.

4.6 Training Methodology

Each network (Figure 10) is trained separately from the other networks, but using exactly the same hyperparameters. Input batches while training are constructed with a fixed number of patches on the corresponding orientation. An epoch corresponds to the network seeing a single patch from every slice in the training set. The maximum number of epochs a training process is allowed to go is also fixed.

Neural network training performs adjustments to weights w present in the network, which in the case of a CNN are convolutional kernel values. These weights are changed based on values returned by the loss functions, that measures how "wrong" an output is in relation to a target, which is assumed to be the expected output. The gradient δ_w of a weight represents the change in loss caused by a change in the weights, in other words, a derivation, and guides the training process in the direction of minimizing the loss. In this thesis, the gradient for each weight was calculated using backpropagation in the form of PyTorch's Autograd (PASZKE *et al.*, 2017). The optimization process is controlled by an optimizer, and three were used on this thesis: Stochastic gradient descent (SGD), Adaptative momentum estimation (ADAM) and Rectified ADAM (RADAM).

With the SGD optimizer, the derivation of an updated weight w_t for a current discrete time t as a function of past weight w_{t-1} can be expressed as:

$$w_t = w_{t-1} - \alpha \Delta_w L(O, T) \tag{4.17}$$

For a loss function L over outputs O and targets T. SGD can also be implemented with momentum, where past w are also taken into consideration. α controls the speed of the optimization process, and is called the learning rate (LR). Finding the correct learning rate is a key factor in training a CNN. A high learning rate can make the model skip states where the minimum loss is achieved, where a low learning rate can lead to very slow convergence, and to the model being stuck in a local minimum. Some implementations change the learning rate during training with a function of the number of epochs passed. Some other optimizers try to change the learning rate adaptively, such as ADAM. ADAM computes a learning rate per parameter, instead of using a global learning rate, and takes into consideration a moving average of gradients. A weight update for ADAM can be expressed as:

$$w_t = w_{t-1} - \eta \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{4.18}$$

Where η is the step-size, that can vary between iterations. m_t and v_t are bias corrected first and second momentums, respectively. These momentums are a function of gradients and the square of the gradients, also in relation to an input and loss function. More details can be found in its original publication in (KINGMA; BA, 2014).

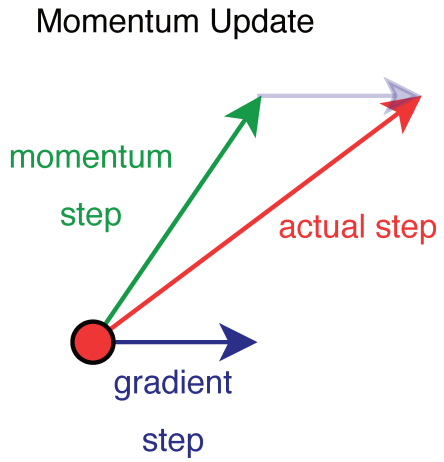


Figure 19 – A 2D illustration of momentum leading to a different step direction, in combination with the current gradient.

The usage of momentum and moving averages avoids the next step to be completely determined by the current batch of data, since batches can be very randomized.

The momentum keeps pointing to a general direction of minimization, where the current gradient points to the minimization for the current batch. The final step can be defined by a combination of the two (Figure 19).

Recently, RADAM was proposed as an improvement to ADAM. It exploits the idea that, in the initial phases of training, a “warm-up" period is advantageous to offset the initial high variance. In summary, the authors added a “rectifier" which disables the adaptative momentum part of ADAM when the variance is estimated to be high, stabilizing the initial phases of training. More details can be found in (LIU *et al.*, 2019). This work uses the implementation provided by the author.

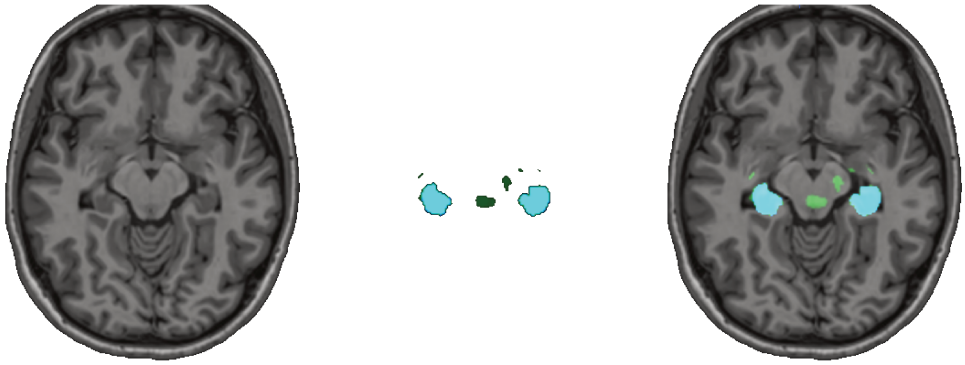


Figure 20 – 2D axial slice of the post processing results. In blue, the selected largest connected volumes, and in green, the discarded small volumes.

4.7 Consensus and Post-processing

The consensus depicted in Figure 10 consists of taking the average from the volumetric activations of all three CNNs. Recall that M_o refers to the activation volume produced by stacking 2D activations from orientation o network N_o. The consensus C can be expressed as:

$$C = \frac{M_s}{3} + \frac{M_c}{3} + \frac{M_a}{3} \tag{4.19}$$

For sagittal s, coronal c and axial a networks. After construction of the consensus of activations, a threshold is needed to binarize the segmentation. Thresholding is performed in a voxel-wise manner, for threshold value T, as in $C[C > T] = 1$ and $C[C <= T] = 0$. While developing this methodology, it was noticed that using patches, although improving generalization, resulted in small structures of the brain being recognized as the hippocampus. To remove those false positives, a implementation of 3D labeling of connected volumes from (DOUGHERTY; LOTUFO, 2003) was used, with subsequent removal of the smaller volumes. The two largest volumes are kept, or one

if only one is present (Figure 20). This post processing is performed after the average consensus of all networks and threshold application.

4.8 3D U-Nets and 3D fine tuning

For the sake of comparison, a 3D U-Net architecture from the literature was trained in our data (ISENSEE *et al.*, 2017). Training of a fully 3D deep network is more computationally intensive and convergence is harder due to the addition of a whole other dimension. The network was trained with batches of MRI volumes as input, and corresponding 3D masks as targets. The output also goes through the 3D Labeling post processing to remove small false positive volumes.

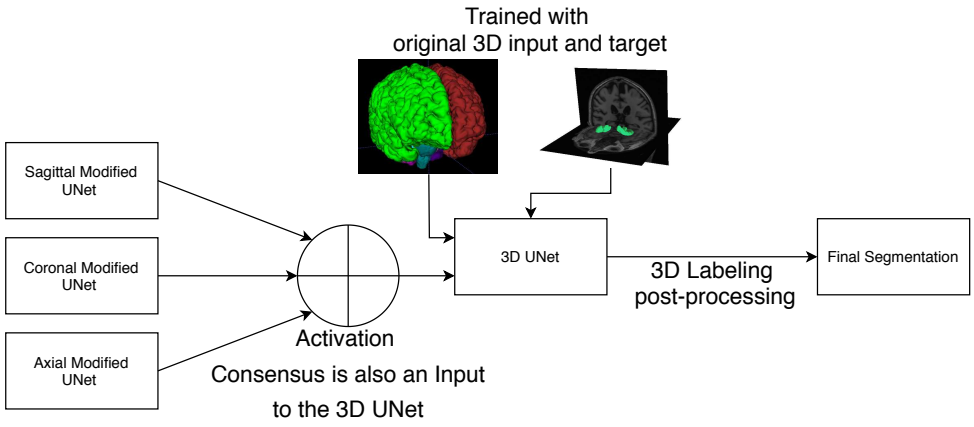


Figure 21 – A 3D U-Net added to the original methodology, as a fine-tuning step after building the consensus.

Also, experiments where performed on including the same 3D U-Net architecture as a fine tuning phase of this method. In this case, the 3D network is trained with the consensus activation mask generated by the previous method (Figure 21), before post processing, and the original image, in separate channels, forming a 4D input. Post processing is applied to the output of the 3D Fine tuning network.

5 Results

Many experiments were performed to validate the various ideas described in the previous Chapter. Dice in the test set after training of all involved networks is the evaluation metric used to define what is the better result. Dice is calculated considering the output volume from our full methodology and the target volumetric hippocampus mask, except for the first Early Experiment (Table 1). Many hyperparameters need to be tuned, including the choice of Loss function, optimizer, learning rate, augmentation strategy, architecture modifications and so on. The HarP dataset was used in most later experiments, while MNI-HCUnicamp was used in part of the early experiments.

After defining the best hyperparameters, the final methodology is put to the test against other methods from the literature, in the public HarP test set and on the whole HCUnicamp dataset. Qualitative and quantitative results are presented. Note that HCUnicamp was reserved as a final test set and was not involved on hyperparameters definition.

Finally, as a final experiment, adaptation to the HCUnicamp dataset was attempted, using a hold-out approach. The final methodology is trained and tested in different parts of the HCUnicamp, in some cases including the HarP dataset, to verify the generalization capabilities of this methodology.

5.1 Methodology Development

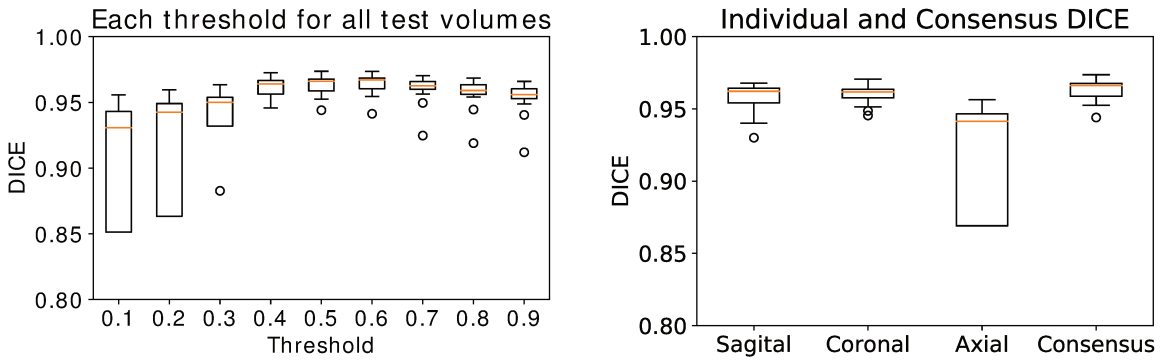
In this section, results in many hyperparameter modifications over the methodology are reported, leading to a final fixed method. Those choices include: optimizer of choice and related parameters; loss function; data augmentation strategies; consensus and post-processing parameters and so on. Whenever a hyperparameter is being experimented on, all other parameters are fixed, unless otherwise specified.

5.1.1 Early Experiments

Orientation	**DICE Loss (Dice)**	**BCE (Dice)**	**MSE (Dice)**
Sagittal	0.9474	0.9293	0.8991
Coronal	0.9406	0.9145	0.8787
Axial	0.9412	0.9111	0.8080

Table 1 – Initial experiments with loss. Validation Dice is reported for the U-Net output slice, with MNI-HCUnicamp as a dataset.

For these experiments, SGD was used as a training optimizer, with 0.005 learning rate. A momentum of 0.9 is used whenever using SGD in the next experiments, following the original U-Net paper (RONNEBERGER *et al.*, 2015). Early experiments with a base U-Net architecture confirmed that using only slices with hippocampus presence was advantageous, and that choice was kept for the rest of the thesis. Comparisons were also performed between the performance of each U-Net using input slices of different orientations, and using different Loss Functions (Table 1). For these experiments, inputs were 128x128 Center Cropped slices, from MNI-HCUnicamp, and targets the respective 2D annotations. Reported DICE is the mean over test slices.



(a) (b)

Figure 22 – (a) Dice values calculated in every binarization threshold (THS) values varying from 0.1 to 0.9 in all 22 MNI-HCUnicamp test volumes. (b) The consensus methodology combines knowledge from all orientations into a more stable final result.

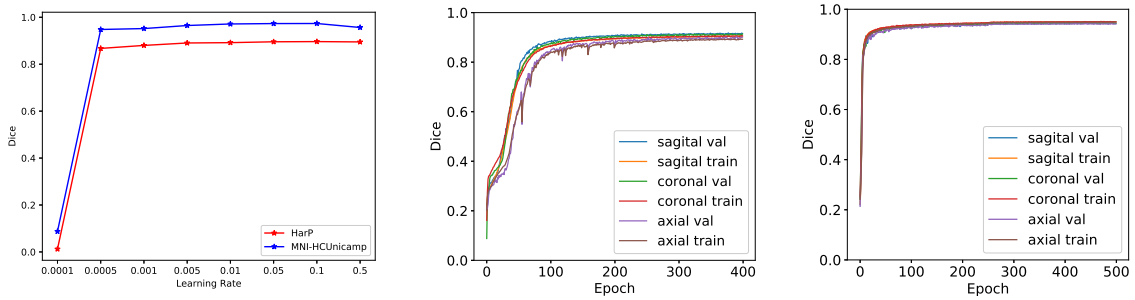
This experiment fixed DICE Loss as the loss function for early experiments. Using the full consensus methodology, with post-processing, the optimal threshold for thresholding after the consensus of activations from all three networks was studied (Figure 22(a)). 0.5 was selected as the threshold value, and from here onward all experiments use the full methodology. The next experiment confirms that the consensus methodology actually returns better results than individual networks. Comparisons were made with using only one of the networks, and performing post processing on its output, against post-processing the consensus of activations. The consensus methodology return more stable results, with less variance in Dice in comparison to individual networks (Figure 22(b)).

The next experiment tests the performance of the modifications performed on the original U-Net architecture, on MNI-HCUnicamp. Those include the addition of residual connections, the usage of VGG11 weights on the encoder and the Extended 2D input (Table 2). VGG11 weights worked better than ResNet34 or Kaiming Uniform initialization (HE *et al.*, 2015) (the default initialization). Improved convergence stability and reduced overfitting was perceived in general due to these architectural changes (Figure 23(b) and

Test Dice	Residual Connections	Extended 2D	ResNet34 Weights	VGG11 Weights
0.9333	-	-	-	-
0.9482	✓	-	-	-
0.9553	✓	✓	-	-
0.9584	✓	✓	✓	-
0.9630	✓	✓	-	✓

Table 2 – Showing the improvements on MNI-HCUnicamp's test set, volumetric Dice after including our changes to the U-Net base architecture of each network, and performing consensus. 32^2 input patchs were used.

(c)).



(a) (b) (c)

Figure 23 – (a) Grid search for the optimal learning rate in MNI-HCUnicamp (blue) and HarP (red). Training accuracy curves for (b) baseline U-Net architecture (c) modified U-Net Architecture in MNI-HCUnicamp, with both used the same training hyperparameters and 32^2 patch size.

Now, to select the best initial learning rate for SGD, in MNI-HCUnicamp, a grid-search was performed, fixing it at 0.005 for this specific data. At this point, training in the public HarP dataset was included in the method's development, with another grid-search performed to confirm the optimal learning rate for SGD (Figure 23). 0.9 momentum is still used. Although higher learning rates such as 0.05 returned slightly better results in the test set, that high learning rate often led to unstable networks and divergence in some cases (Figure 24). Hence, this work sets in 0.005 for SGD's initial learning rate.

Description	MNI-HCUnicamp (Dice)	HarP (Dice)
Center 128x128 crop	0.9485	-
Random 16x16 patch, 80% positive 20% negative	0.8676	-
Random 32x32 patch , 50% positive 50% negative	0.9178	-
Random 32x32 patch, 80% positive 20% negative	0.9482	-
Random 32x32 E2D patch, 80% positive 20% negative	0.9630	0.8546
Random 64x64 E2D patch, 80% positive 20% negative	0.9719	0.8748

Table 3 – Different ways used to perform patch selection, and early experiments results.

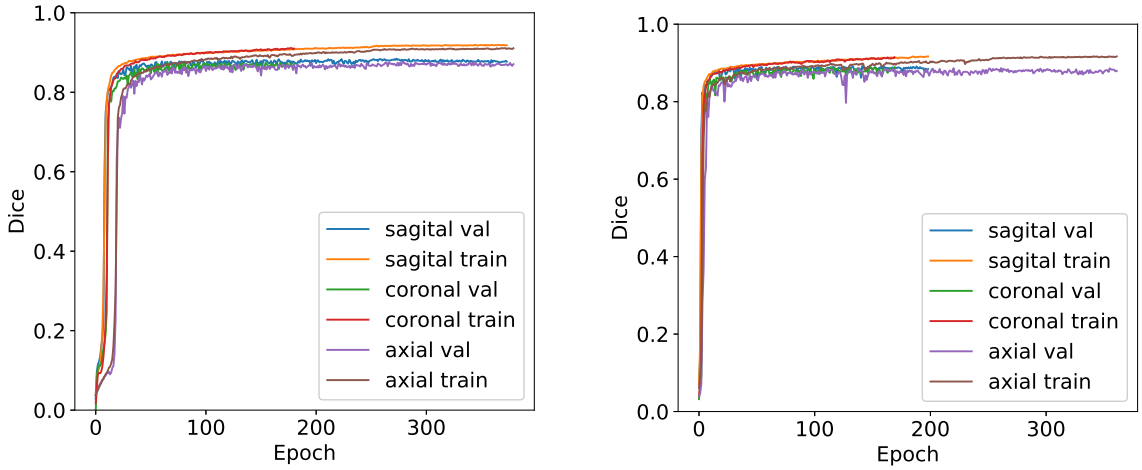


Figure 24 – (a) 0.005 initial learning rate for SGD in HarP provided more stable training than (b) 0.05 and other higher learning rates, even with slightly lesser Dice.

In Table 3, some options for patch selection used in this thesis are listed and tested. Changes between each option are on the size of the patches and the balance between positive and negative patches. For 16^2, one less U-Net layer was used, with 3 Max Pool/Transposed Convolutions instead of 4. Smaller patches resulted in less stable training. 80/20% balance between positive and negative patches, respectively, resulted in better convergence and less false positives than a 50/50% balance. With 64^2 presenting better results in both datasets, it was fixed as the patch selection strategy from here forward. The use of random patches with neighbour slices (E2D) instead of center crop 128^2 slices also reduced overfitting, while increasing the number of false positive activations. However, these false positives are handled by the 3D labeling post processing.

After noticing the volbrain silver standard masks presented in MNI-HCUnicamp were easy to learn from, and not completely representative of the problem at hand, experiments migrated to mainly using HarP as the training, validation and testing dataset. Modifications over the network architecture were also tested in HarP, including the addition of Batch Normalization (Figure 25). Not using batch normalization led to divergence in many cases.

5.1.2 Hyperparameter Experiments

Some of the most relevant hyperparameters experiments test results are showcased in this section, in a hold-out approach to HarP. While training on HarP with an 80% holdout training set, an epoch consisted of going through around 5000 sagittal, 4000 coronal and 3000 axial random patches extracted from slices with presence of hippocampus, depending on which network is being trained, with a batch size of 200. The max

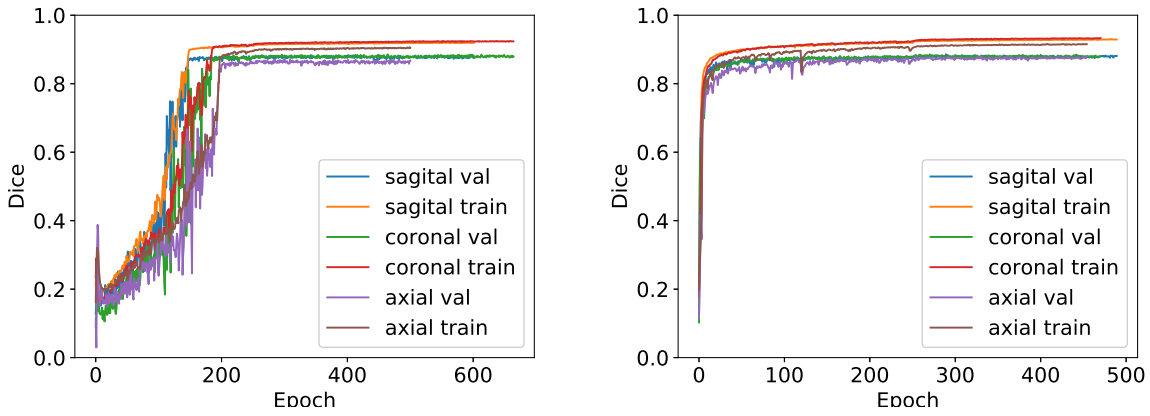


Figure 25 – Dice during training and validation in HarP, for (a) Base U-Net architecture (b) Our modified architecture. Both used the same training hyperparameters, with 32^2 patch size.

Aug.	**Chance (%)**	**Description**
1	100%	Intensity transformation with $a = 0.05$.
2	100%	Intensity transformation with $a = 0.10$.
3	20%	Rotation and scale with $b = 10$.
4	20%	Rotation and scale with $b = 20$.
5	20%	Horizontal flip.
6	20%	Gaussian noise with $\mu = 0$ and $v = 0.0002$.
7	20%	Soft Target with $\lambda = 1.0$
8	100%	Soft Target with $\lambda = 1.5$
9	20%	Random Horizontal and/or Vertical flip.

Table 4 – Description of specific transformation parameters used in hyperparameter experiments, with the % chance of application after patch selection and parameters description. Refer to Section 4.3 for more detailed descriptions.

number of Epochs allowed is 1000, with a patience early stopping of no validation improvement of 200 epochs. Weights are only saved for the best validation Dice.

A variety of combinations of transformations were tested for data augmentation purposes. In this section, a set of augmentations applied to a patch will be represented as $x, y, z...$, with x, y, z referring to the Aug. number in Table 4, in the order of application. If the transformation has a Chance lower than 100%, its application can be skipped. As an example, intensity modification followed by horizontal flips could be represented as $(1, 5)$, however, since 5 has a application chance of 20%, there is a 80% chance that the horizontal flip will not be applied. $-$ refers to using the output of the random E2D patch selection, with no additional augmentation.

From all the results found during the development of this work, it is notable that

Optimizer	**Loss**	**Augs.**	**HarP (Dice)**
SGD	Dice Loss	-	0.8760
SGD	Dice Loss	2, 4, 6	0.8748
SGD	Dice Loss	2, 4, 5, 6	0.8546
ADAM	Dice Loss	-	0.8829
ADAM	Dice Loss	2	0.8820
ADAM	Dice Loss	4	0.8827
ADAM	Dice Loss	6	0.8832
ADAM	Dice Loss	7	0.8675
ADAM	Dice Loss	8	0.8801
ADAM	GDL	-	0.8830
ADAM	GDL	1, 3, 6	0.8862
ADAM	Boundary	-	0.9068
RADAM	Boundary	1, 3, 6	0.9117
RADAM	Boundary	1, 3, 6, 9	**0.9127**
RADAM	Boundary	-	**0.9133**

Table 5 – Augs. refers to what data augmentation transformations were used, from Table 4. The bolded results represents the final models used in the next section. All tests in this table use 64^2 E2D patches and the modified U-Net architecture.

the random patches made the most impact in avoiding overfitting. Data augmentation techniques besides the random patch extraction only impacted overlap results in HarP slightly, in some cases even making results slightly worse in testing (Table 5). This was also verified using different optimizers and loss functions. Note that the Soft Target transformation did not result in improvements, and was not used in the final method. Data augmentation's most relevant impact was avoiding early stopping due to no validation improvements. This would lead to unstable networks in some cases. An example of an unstable network can be seen in Figure 22(b), where the axial CNN is performing under the other CNNs, causing more variance in the results. Using Horizontal and Vertical flips (Aug. 9) consisted of an attempt to achieve global rotation invariance, while testing with artificially rotated volumes and an orientation detector. However, global rotation invariance was not achieved. Thus, the requirement for a specific orientation for the head in the input volume is still maintained. This can be easily achieved with an automatic MNI152 registration as a pre-processing step, which is provided as an option in this method.

While using only one channel sigmoid activations as an output, recall that early experiments defined Dice Loss as the best convergence and results, beating MSE and BCE (Table 1). Using a softmax output and GDL gave similar results to Dice Loss. However, implementation of the recent Boundary Loss resulted in better test Dice, under the same hyperparameters (Table 5). Attempts at changing the GDL term on Boundary Loss to DICE Loss did not return any improvements. Additionally, changing the way the weight α is changed over the training epochs also did not return better results than the original

implementation.

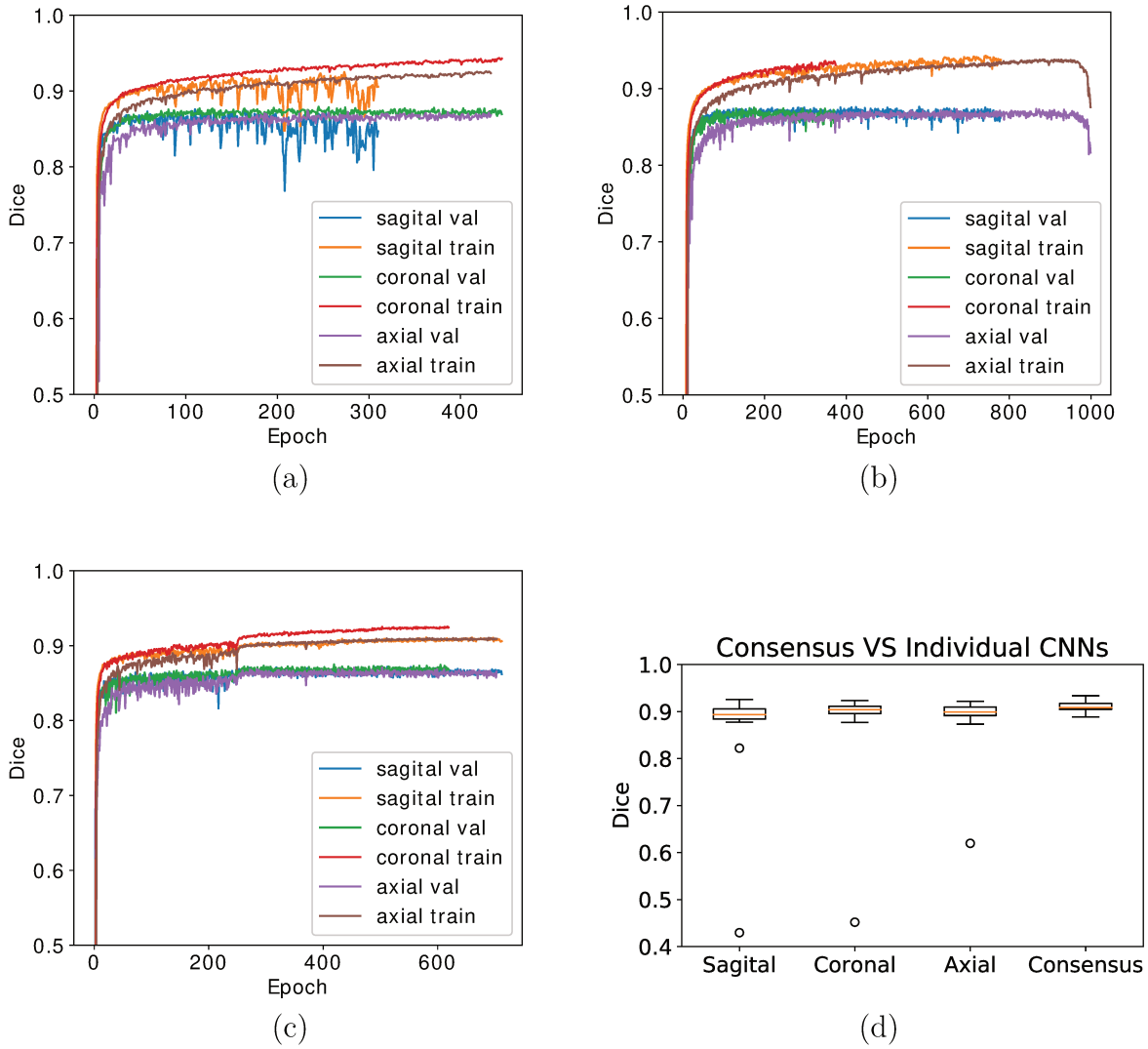


Figure 26 – Validation and training Dice for all networks, using: (a) ADAM (b) RADAM. Both with same hyperparameters and no LR stepping. Early stopping is due to patience. RADAM displays more stability. (c) Training and validation Dice curves for the best model, with RADAM and LR stepping after 250 epochs. (d) Boxplot for HarP test models, showing the improvement in variance and mean Dice from the Consensus compared to using only one network. In the individual network studies, post processing is also applied to remove false positives.

Optimal learning rates for SGD in HarP were confirmed before with grid-search (Figure 23(b)). Besides using SGD, attempts at using ADAM and RADAM were also performed in this thesis. For ADAM, 0.0001 initial LR was used as a default (recall that ADAM works with adaptative learning rates), delivering slightly better performance than SGD. The recent RADAM ended up being the optimizer of choice for the final method,

due to improved training stability and results (Figure 26). Although RADAM is robust to initial LR variance, a study comparing RADAM with 10^{-2} to 10^{-4} led to the choice of 10^{-3} as the initial LR. Some experiments were performed on learning rate scheduling. Between multiplying by 0.1 after 125, 250 or 500 epochs, or multiplying by 0.9 after every 100 epochs. The final choice in scheduling is multiplication by 0.1 after 250 epochs, its impact showcased in Figure 26(c).

For the experiments in the next section, both results in bold in Table 5 were used as representation of the best method, representing using augmentation or not using augmentation.

Thus, the final methodology uses RADAM as optimizer, with Boundary Loss, 0.001 initial LR, multiplied by 0.1 after 250 epochs, with 200 epochs of patience and a maximum of 1000 epochs. The modified U-Net architecture is employed, with the addition of residual connections, batch normalization, E2D 64^2 random input patches and VGG11 encoder weights.

Of notice is the improved stability of the final method (Figure 26(d)) in comparison to the early experiments (Figure 22(b)), showcased by the boxplots comparing individual networks to the consensus.

5.1.3 3D Unet and 3D Fine-tuning

For comparison's sake, this work also experiments with an off-the-shelf 3D U-Net architecture, from Isensee et al. (ISENSEE *et al.*, 2017), originally a Brain Tumor segmentation work. ADAM is used as an optimizer for DICE Loss, with 0.001 initial learning rate and HarP 160x160x160 center crops as input. SGD when used had 0.05 initial learning rate. Training of a 3D architecture requires much more memory due to the use of 4D kernels, (3D kernels with many channels). Hence, the batch size was limited to 2 volumes. Training had a maximum number of epochs of 200, with a patience of 20.

Method	Optimizer	**Augs.**	**HarP (Dice)**
3D U-Net	SGD	-	0.8493
3D U-Net	ADAM	2, 6	0.8596
E2D Consensus with 3D Fine tuning	ADAM	2, 6	0.8748
E2D Consensus with 3D Fine tuning	RADAM	2, 6	0.9077

Table 6 – Results from experiments with 3D architectures were not superior to the initial E2D Consensus methodology.

Using a fourth 3D U-Net as a consensus generator/error correction phase (Figure 21), the 3D fine tuning, returned better results than just training the 3D network alone. However, results are similar to the original methodology without the 3D U-Net (Table 6). Once again, the RADAM optimizer performs better than ADAM and SGD.

Due to the overhead introduced by involving the 3D architecture, the final methodology continues to be the E2D Consensus without 3D networks.

5.2 Quantitative Results

In this section, we report quantitative results of our final method and others from the literature in both HarP and HCUnicamp. The 3D U-Net experiment is also included in this comparison.

For the evaluation with the QuickNat (ROY *et al.*, 2019) method, a public implementation from the author was used, where volumes and target needed to be conformed to its required format, causing interpolation. As far as we know, the method does not have a way to return its predictions on the volume's original space. DICE was calculated with the masks on the conformed space. Note that QuickNat performs segmentation of multiple brain structures, not only the Hippocampus. For Hippodeep (THYREAU *et al.*, 2018), a public implementation made available by the author was also used. For more details, recall that QuickNat and Hippodeep were discussed in Chapter 2.

5.2.1 HarP Results

Deep Learning Methods	**HarP (DICE)**
3D U-Net (ISENSEE *et al.*, 2017)	0.86
Hippodeep (THYREAU *et al.*, 2018)	0.85
QuickNat (ROY *et al.*, 2019)	0.80
(ATALOGLOU *et al.*, 2019)	**0.90***
E2DHipseg (this work)	**0.90***
Label Fusion/Atlas-based Methods	
FreeSurfer v6.0 (FISCHL, 2012)	0.70
(CHINCARINI *et al.*, 2016)	0.85
(PLATERO; TOBAR, 2017)	0.85

Table 7 – Reported testing results for HarP. This work is named E2DHipseg. Results with * were calculated following a 5-fold cross validation.

The best hold-out mean Dice is 0.9133. When using a hold-out approach in a relatively small dataset such as HarP, the model can be overfitting to better results in that specific test set. With that in mind, we also report results with cross validation. 5-fold training and testing is used, where all three networks are trained and tested with each fold. With 5-fold our model achieved 0.90 ± 0.01 Dice. Results reported by other works are present in Table 7. Our methodology has similar top performance to Atalaglou

et al. recent, simultaneous work (ATALOGLOU *et al.*, 2019). Interestingly, the initial methodology of both methods is similar, in the use of multiple 2D CNNs.

5.2.2 HCUnicamp Test

HCUnicamp (Controls)					
Method	**Both (Dice)**	**Left (Dice)**	**Right (Dice)**	**Precision**	**Recall**
3D U-Net (ISENSEE *et al.*, 2017)	0.80 ± 0.04	0.81 ± 0.04	0.78 ± 0.04	0.76 ± 0.10	0.85 ± 0.06
Hippodeep (THYREAU *et al.*, 2018)	0.80 ± 0.05	0.81 ± 0.05	0.80 ± 0.05	0.72 ± 0.10	0.92 ± 0.04
QuickNat (ROY *et al.*, 2019)	0.80 ± 0.05	0.80 ± 0.05	0.79 ± 0.05	0.71 ± 0.11	$\mathbf{0.92 \pm 0.04}$
E2DHipseg without Aug.	0.82 ± 0.03	0.83 ± 0.03	0.82 ± 0.03	$\mathbf{0.78 \pm 0.10}$	0.88 ± 0.06
E2DHipseg with Aug.	$\mathbf{0.82 \pm 0.03}$	$\mathbf{0.83 \pm 0.03}$	$\mathbf{0.82 \pm 0.04}$	0.78 ± 0.10	0.89 ± 0.06
HCUnicamp (Patients)					
3D U-Net (ISENSEE *et al.*, 2017)	0.74 ± 0.08	0.48 ± 0.39	0.56 ± 0.36	0.66 ± 0.12	0.87 ± 0.07
Hippodeep (THYREAU *et al.*, 2018)	0.74 ± 0.08	0.48 ± 0.39	0.57 ± 0.37	0.63 ± 0.12	0.91 ± 0.06
QuickNat (ROY *et al.*, 2019)	0.71 ± 0.08	0.47 ± 0.38	0.56 ± 0.36	0.59 ± 0.12	$\mathbf{0.92 \pm 0.06}$
E2DHipseg without Aug.	$\mathbf{0.77 \pm 0.07}$	0.49 ± 0.40	0.58 ± 0.37	$\mathbf{0.69 \pm 0.11}$	0.88 ± 0.07
E2DHipseg with Aug.	0.76 ± 0.07	$\mathbf{0.50 \pm 0.40}$	$\mathbf{0.58 \pm 0.37}$	0.68 ± 0.11	0.89 ± 0.07

Table 8 – Locally executed testing results for HCUnicamp. All 190 volumes from the dataset are included, and no model saw it on training. The 3D U-Net here is using the same weights from table 7. Note that QuickNat performs whole brain multitask segmentation, not only hippocampus.

The HCUnicamp dataset was kept untouched during the thesis, for this final experiment. Without involvement in our method's hyperparameter optimization, its now used as a final test dataset, including other methods from the literature in the test. As described previously, the HCUnicamp dataset has lack of one of the hippocampi in many of it's scans (see Figure 7). Table 8 has mean and standard deviation Dice for all HCUnicamp volumes, using both masks, or only one the left or right mask, with multiple methods. "with Aug." refers to the use of augmentations 1, 3, 6 in training. We also report Precision and Recall, per voxel classification, where positives are hippocampus voxels and negatives are non hippocampus voxels. Precision is defined by $TP/(TP+FP)$ and Recall is defined by $TP/(TP+FN)$, where TP is true positives, FP are false positives and FN are false negatives. All tests were run locally. Unfortunately, we were not able to reproduce Atalaglou et al.'s method for local testing.

Our method performed better than other recent methods on the literature in the HCUnicamp dataset, even though HCUnicamp is not involved on our methodology development. However, no method was able to achieve more than 0.8 mean Dice in epilepsy patients. The high number of false positives due to hippocampus removal is notable by the low left and right DICE, and low precision. The impact of additional augmentations was not statistically significant in the epilepsy domain.

Our method takes around 15 seconds on a mid-range GPU and 3 minutes on a consumer CPU to run, per volume (Table 9). As other Deep Learning methods, E2DHipseg

Hippocampus Segmentation Method	Approximate Runtime Speed (s/volume)
FreeSurfer v6.0 (FISCHL, 2012)	28800
3D U-Net (ISENSEE *et al.*, 2017)	5
(PLATERO; TOBAR, 2017)	1020*
Hippodeep (THYREAU *et al.*, 2018)	60
QuickNat (ROY *et al.*, 2019)	20
(ATALOGLOU *et al.*, 2019)	15*
E2DHipseg	**15**
E2DHipseg (with orientation correction)	**75**

Table 9 – Comparison of runtime speed, in seconds per input volume, between recent hippocampus segmentation methods. E2DHipseg runs faster if orientation correction of the input with MNI152 pre-registration is not needed. *Reported times are from local testing in the same computer, using a GPU, except for Ataloglou and Platero's works, which are reported on the respective papers.

is many times faster than Atlas Based methods. Although the 3D-U-Net approach is heavy in terms of training difficulty, its inference speed due to most of the processing occurring in the GPU is notable.

All the code used on its development is available in <github.com/dscarmo/e2dhipseg>, with instructions for how to run it in an input volume. A free executable version for medical research use, without environment setup requirements, is available on the repository. Due to the need for correct head orientation, there is an option to use MNI152 registration when predicting in a given volume, to avoid problems with different head orientations (in a similar way to Hippodeep). Even when performing registration, the output mask will be back in the input volume's space, using the inverse transform. E2DHipseg does not require additional pre-processing. A GPU is recommended for faster prediction but not necessary.

5.2.3 Adaptation to HCUnicamp

After seeing the poor results of all methods in the hippocampus resection cases, additional experiments were performed involving HCUnicamp data in training, to try and learn to recognize the resection.

The experiments involved making a hold-out separation of HCUnicamp. In the previous experiment, all volumes were involved in the testing. In this one, hold-out is performed with balance between control and patients. Note that these results are not directly comparable with the previous results on HCUnicamp, since the whole dataset was included in Table 8's tests. To avoid confusion, the hold-out training/testing dataset will be refered to as HCUnicamp-H. Experiments were also performed including only control volumes or only patient volumes, with the same hold-out approach (Table 10). Results improve when training on HCUnicamp-H, but the high standard deviation still

shows that the method is failing to recognize resections.

Trained on	**Both (Dice)**	**Left (Dice)**	**Right (Dice)**
Patients	0.84 ± 0.04	0.60 ± 0.41	0.56 ± 0.42
Controls and Patients	0.86 ± 0.05	0.71 ± 0.36	0.74 ± 0.34
Controls	0.90 ± 0.01	0.89 ± 0.02	0.90 ± 0.01

Table 10 – E2DHipseg with networks trained in HCUnicamp-H. Test results for training in all volumes, only patients or only controls, using hold-out for testing. Results are in the respective test sets.

Another experiment attempts to generalize to both datasets's patients and controls, at the same time (Table 11). Training is performed concatenating the HarP and HCUnicamp-H datasets. The datasets where mixed together with a 70% training, 10% validation and 20% testing hold-out. The presence of patients and controls is balanced between the sets. Also displayed is performance from testing in a different domain while training in other.

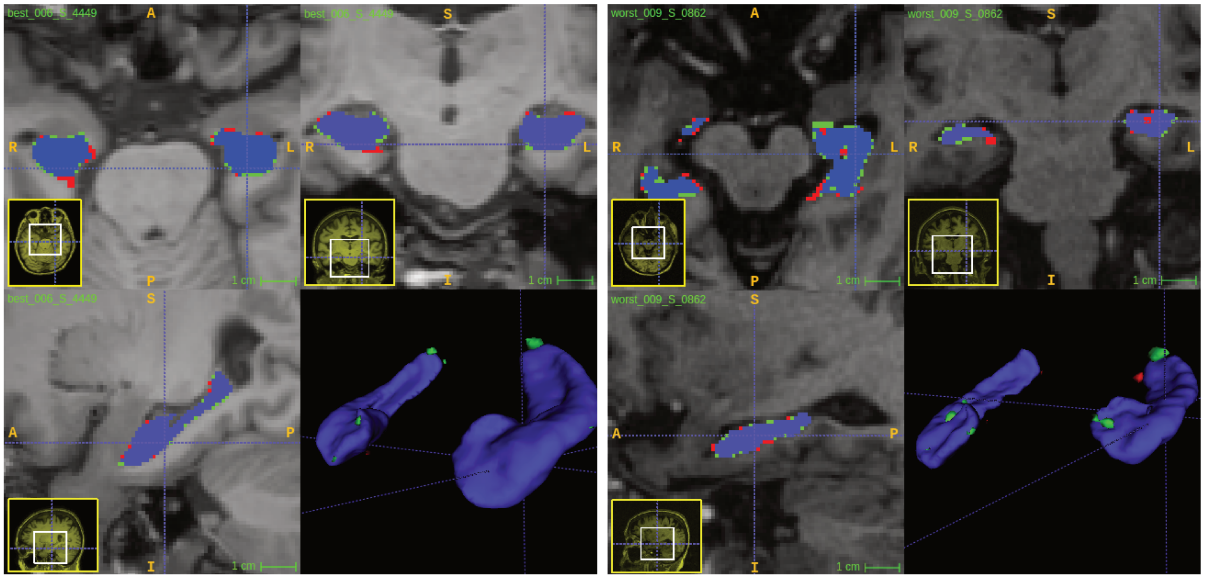
Trained on	**Tested on**	**Both (Dice)**	**Left (Dice)**	**Right (Dice)**
Harp	HCUnicamp-H	0.79 ± 0.07	0.65 ± 0.33	0.68 ± 0.31
HCUnicamp-H	HarP	0.50 ± 0.29	0.50 ± 0.31	0.50 ± 0.29
Harp + HCUnicamp-H	HarP	0.89 ± 0.01	0.89 ± 0.01	0.89 ± 0.02
Harp + HCUnicamp-H	HCUnicamp-H	0.85 ± 0.04	0.69 ± 0.35	0.73 ± 0.33

Table 11 – This table compares the generalization potential when training in one dataset's training set and testing in the other's separated test set. As expected betters results are achieved when involving both domains in training.

Although the model was able to achieve good overall Dice in both HarP and HCUnicamp-H when involving both in training, Dice standard deviation only in the left or right hippocampus still shows signals of problems when dealing with hippocampus resection. When training only in HCUnicamp-H and testing in Harp, in many cases the method predicted a resection was present, specially in darker scans, when it wasn't, resulting in high false negatives and very low mean Dice of around 0.5.

5.3 Qualitative Results

This section explores 2D and 3D visualizations of results from the presented method, E2DHipseg, and others.



(a) (b)

Figure 27 – Multiview and 3D render of E2DHipseg's results for (a) best and (b) worst cases in the HarP test set. Prediction in green, target in red and overlap in blue.

5.3.1 E2DHipseg's Best and Worst Results

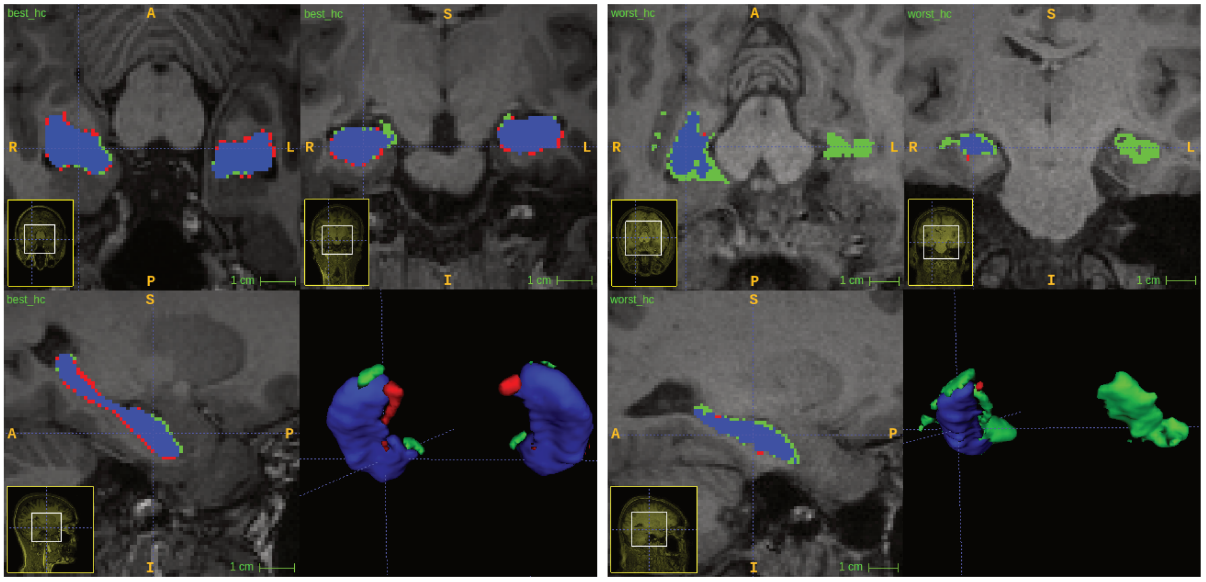
While visually inspecting HarP results, very low variance was found, without presence of outliers. This is indicated by looking at the low deviation in the consensus boxplot in Figure 26(d) and the best and worst segmentation in Figure 27. Other methods present similar, stable results.

In HCUnicamp, way more errors are visible in the worst segmentations from E2DHipseg in Figure 28(b). Specially where the hippocampus is removed. Other methods have similar results, with false positives in voxels where the hippocampus would be in a healthy subject or Alzheimer's patient. In general, the definition of the borders seems to be the hardest part of hippocampal segmentation.

As expected, the best segmentations, as displayed in Figure 28(a), were in control, healthy subjects, for all methods.

5.3.2 Comparison in the same Volumes

Additionaly, qualitative visualizations of QuickNat, Hippodeep and E2DHipseg in the same two randomly selected volumes from HCUnicamp are presented in Figure 29, Figure 30 and Figure 31. The patient in the left (a) with hippocampus resection and a control subject in the right (b). These volumes are referenced as Subject A and Subject B.



(a) (b)

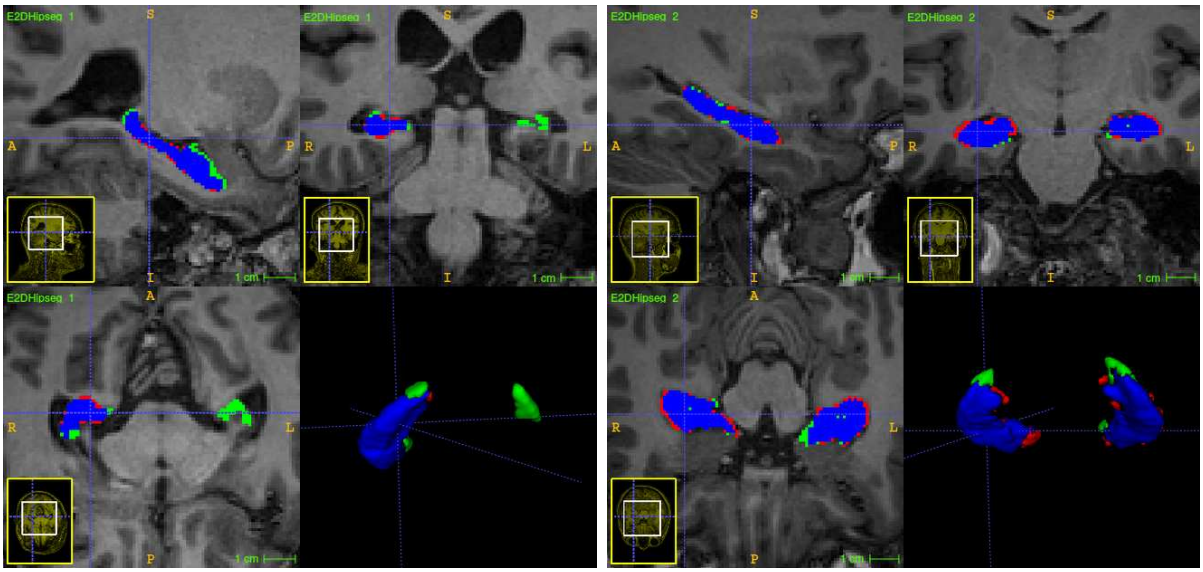
Figure 28 – Multiview and 3D render of E2DHipseg's results for (a) best and (b) worst cases in the HCUnicamp dataset. Prediction in green, target in red and overlap in blue.

The similar poor performance of all methods in noticing the lack of a hippocampus is notable. Visually, in general the methods seem to perform similarly.

5.3.3 Adaptation to HCUnicamp

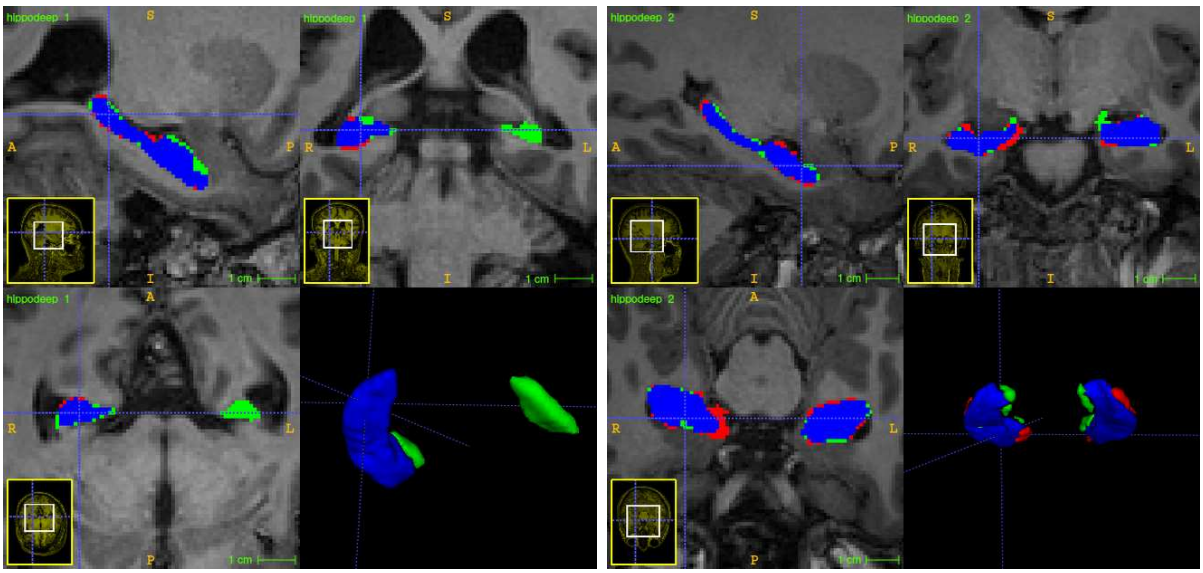
Finally, E2DHipseg trained in both HarP and HCUnicamp-H is put to the test in predictions in a HarP Alzheimer's Disease case named Subject C (Figure 32(b)), and Subject A (Figure 32(a)).

Even after involving epilepsy volumes in training, the method is still able to achieve good results in HarP, even in Alzheimer's Disease cases. Results of the test in HCUnicamp-H look more stable than when the Epilepsy patients with hippocampus resection were not involved, but the false positives still remain.



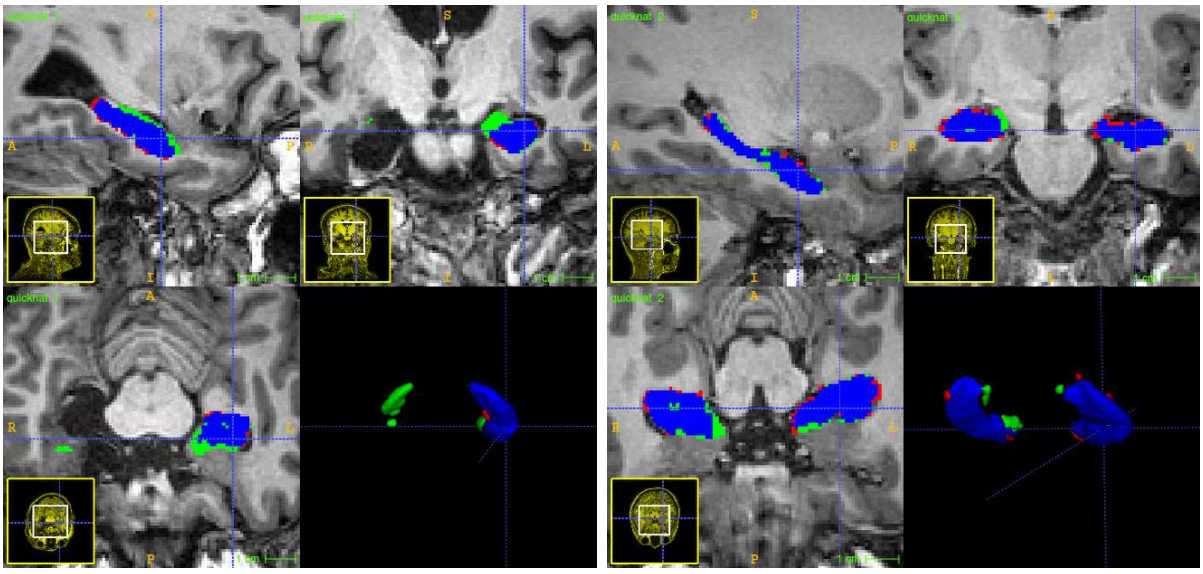
(a) (b)

Figure 29 – Multiview and 3D render of a (a) Subject A (HCUnicamp patient) and (b) Subject B (HCUnicamp control). Results are from E2DHipseg. Prediction in green, target in red and overlap in blue.



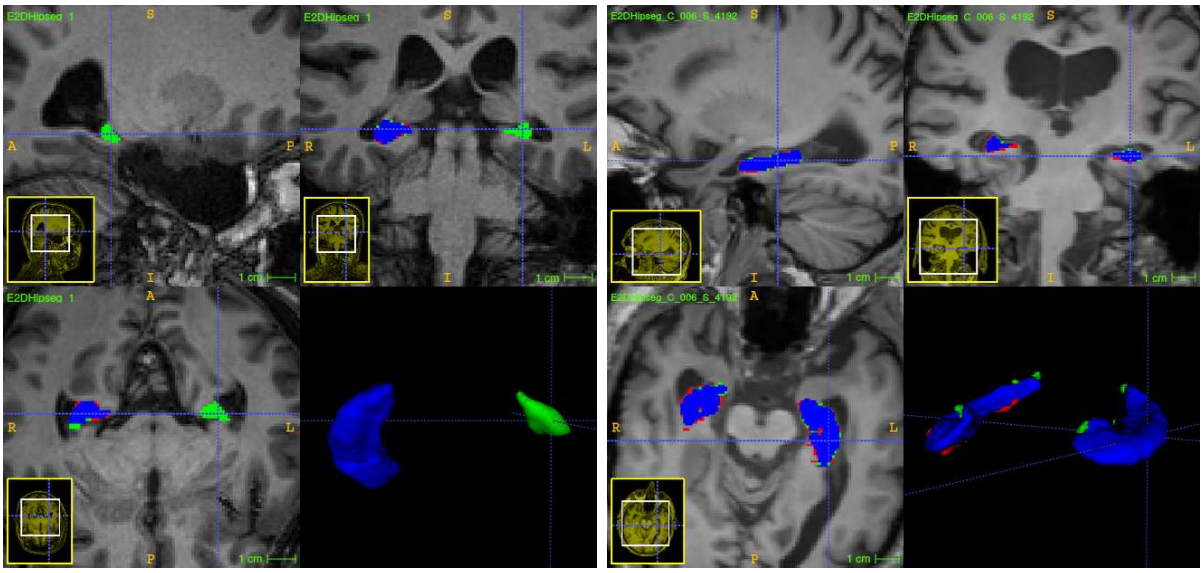
(a) (b)

Figure 30 – Multiview and 3D render of (a) Subject A and (b) Subject B. Results are from Hippodeep. Prediction in green, target in red and overlap in blue.



(a) (b)

Figure 31 – Multiview and 3D render of (a) Subject A and (b) Subject B. Results are from Quicknat. Differences in contrast and orientation are from the conformity processing required by Quicknat. Prediction in green, target in red and overlap in blue.



(a) (b)

Figure 32 – Multiview and 3D render of (a) Subject A and (b) Subject C, a Alzheimer's Disease case from HarP. Results are from E2DHipseg trained in both HCUnicamp (hold-out) and HarP. Prediction in green, target in red and overlap in blue.

6 Discussion and Conclusion

This chapter presents extended discussions of the previously presented results and concluding thoughts on this work.

6.1 Discussion

E2DHipseg uses a modified version of the now traditional U-Net architecture. Small modifications to the architecture can be advantageous, as showed in this work. Additionally, two recent publications in the field helped define E2DHipseg as a competitive method with the state-of-the-art in the HarP dataset, achieving over 0.90 Dice. Those were the use of the new RADAM optimizer and the Boundary Loss. This shows that the field is still open to advancements in optimizers, loss functions and network architecture. However they are not guaranteed to return better results in other datasets. Each case requires an specific study to verify if a modified architecture, RADAM or Boundary Loss will make a difference in relation to the base U-Net, the ADAM optimizer, GDL, or even other choices of loss, optimizers and hyperparameters in general.

It is noticeable that this work stopped using the MNI-HCUnicamp dataset after early experiments. The silver standard volbrain masks present on it where not helping with better performance on the public HarP benchmark, using gold standard annotations.

The fact that patches are randomly selected and augmented in runtime means they are mostly not repeated in different epochs. This is different to making a large dataset of pre-processed patches with augmentation. We believe this random variation during training is very important to ensure the network keeps seeing different data in different epochs, improving generalization and avoiding overfitting. The patch selection alone was enough to provide lengthy training without overfitting, and additional data augmentations did not make much difference in final results. The random patch selection in runtime is similar to the Dropout technique (SRIVASTAVA *et al.*, 2014), in that it will not use all data available in the dataset in every epoch. Better yet, even with all the data randomness, re-runs of the same experiment resulted mostly in the same final results, within 0.01 mean Dice of each other.

One of the first questions raised when using an ensemble of networks is if the ensemble brings advantages in comparison to using only one, well trained network. It was observed that most of the false positives some of the networks produce are eliminated by the averaging of activations. Also, in some cases, one of the networks fails and the other two "save" the result. All of this is visible looking at Figure 26(d), and the smaller variance of the consensus result. Post-processing phases are sometimes avoided when using CNNs,

but this work shows that tradittional post-processing is still relevant even with most works focusing on end-to-end Deep Learning approaches. Thresholding and 3D connected volumes labeling post processing allows the methodology to focus on good segmentation on the hippocampus area, without worrying with small false positives in other areas of the brain.

As visible on the results of multiple methods, Dice in the HCUnicamp dataset is not on the same level as what is seen on the public benchmark. Most methods have false positives on the removed hippocampus area, in a similar fashion to what is displayed in Figure 28(b). The fact that QuickNat and Hippodeep have separate outputs for left and right hippocampus does not seen to be enough to solve this problem. We believe the high false positive rate is due to textures similar to the hippocampus, present in the hippocampus area, after its removal. This observation called for confirmation with our medical sciences partners, if the hippocampus was really completely removed, which they confirmed. Although E2DHipseg got better performance than the other tested methods, QuickNat has better Recall. The recall metric is heavily connected to being able to recognize what is not the hippocampus (low false negatives). QuickNat's higher recall makes sense with its ability to recognize other structures on the brain. Its multitask approach seems to help with correctly identifying negatives in ambiguous cases.

Final results report attempts to adapt the methodology to HCUnicamp-H volumes, and test the generalization capabilities of the methodology. Training in HCUnicamp-H improves results, but the high standard deviation and mistakes on hippocampus resections is still present. A similar story is seen while analysing results from concatenating the HarP and HCUnicamp-H dataset in training. The method is able to achieve good overall Dice in both HarP and HCUnicamp-H, of 0.89 and 0.85, but analysing the structures separately shows the high standard deviation on missed resections. In cases of false positives in resections, the left or right Dice will be 0, pulling the mean Dice down drastically. This is confirmed in the qualitative results and does not happen when training and testing in HCUnicamp-H controls or Harp, as showcased by the similar, low standard deviation between overall Dice and left/right Dice.

6.2 Conclusion

This master thesis presents a deep learning based hippocampus segmentation method including consensus of multiple U-Net based CNNs and traditional post-processing, successfully using a new optimizer and loss function from the literature. The goal was to surpass the performance of traditional hippocampus segmentation methods and be competitive with the current, rapidly evolving, state-of-the-art of the field. Additionally, the hypothesis was raised that current automatic hippocampus segmentation methods would

not have the same performance on our in-house epilepsy dataset, HC-Unicamp, with some cases of hippocampus resection.

Quantitative and qualitative results show that E2DHipseg beats traditional methods in performance and speed. Competitive performance of the proposed method is observed in relation to state-of-the-art deep learning based hippocampus segmentation methods, in the public HarP benchmark. However, all methods failed to correctly take into account hippocampus resection, present in the HCUnicamp dataset. This raises the concern that current automatic hippocampus segmentation methods are not ready to these outliers. Even with poor performance when resections were present, E2DHipseg still showed superior metrics than other methods on HCUnicamp, without involvement of it in the methodology's development. The final experiment shows that results are improved when training on HCUnicamp data, but there is possibility of improvement when dealing with resection, with changes in the methodology.

6.3 Future Work

Future research plans include a better study of CNN adaptation to abnormalities such as hippocampus resection, with, as an example, a resection detection phase. More exploration of 3D networks and different architectures (CHEN *et al.*, 2018) is another future path. This method could be expanded for segmentation of other brain structures, or even multiple structures at the same time. Future research plans also include the use of attention masks (OKTAY *et al.*, 2018), and involvement of multimodal data such T2, T2-flair, DTI and so on.

6.4 Publications

The following works were prepared during the development of this thesis:

6.4.1 Journal Submission

A summarized version of this thesis was submitted to the Journal of Neuroscience Methods, titled Hippocampus Segmentation on Epilepsy and Alzheimer's Disease Studies with Multiple Convolutional Neural Networks, and is currently on the revision stage. This is the first manuscript that includes mention to HCUnicamp and the manual annotations of epilepsy data, and it showcases the difficulty of state-of-the-art hippocampus segmentation methods with the resections.

6.4.2 Full Paper

The modified U-Net architecture from this thesis's methodology was used in collaboration with Gustavo Pinheiro to test the usage of DTI data in brain structure segmentation. The resulting paper is titled Convolutional Neural Network on DTI data for Sub-cortical Brain Structure Segmentation (PINHEIRO *et al.*, 2019), published in the International Workshop on Computational Diffusion MRI from the 22nd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), in Shenzhen, China.

6.4.3 Short Papers

A 3 page short paper titled Extended 2D Consensus Hippocampus Segmentation (CARMO *et al.*, 2019a) was presented as a poster at the International Conference on Medical Imaging with Deep Learning (MIDL) in London, July 2019. This publication detailed an earlier version of the proposed methodology tested in HarP.

Another short paper titled, in portuguese, *Segmentação do Hipocampo com Múltiplas Redes Neurais Convolucionais 2D Estendido*, was published and presented for a local audience of the EADCA Workshop from UNICAMP's School of Electrical and Computer Engineering, on November 2019, also summarizing the findings of this thesis up until HarP experiments.

6.4.4 Abstracts

An abstract on the preliminary results of this work named Deep Volumetric Consensus Hippocampus Segmentation was selected for an oral presentation in the 6th BRAINN congress, at UNICAMP, on April 2019, displaying initial results of the consensus methodology.

Initial findings using 2D slices and one single U-Net architecture were published as an abstract, titled Deep hippocampus segmentation with 2D U-Nets over coronal view, at the São Paulo-Alberta Brainhack workshop, at UNICAMP, on October 2018.

HarP-ADNI Acknowledgement

Alzheimer's disease data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Ar-

aclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Bibliography

AMARAL, D.; LAVENEX, P. Hippocampal neuroanatomy. Oxford University Press, 2007. Citado na página 18.

ANDERSEN, P. *The hippocampus book*. [S.l.]: Oxford University Press, 2007. Citado 2 vezes nas páginas 17 and 18.

ATALOGLOU, D.; DIMOU, A.; ZARPALAS, D.; DARAS, P. Fast and precise hippocampus segmentation through deep convolutional neural network ensembles and transfer learning. *Neuroinformatics*, Springer, p. 1–20, 2019. Citado 6 vezes nas páginas 24, 25, 26, 56, 57, and 58.

BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 39, n. 12, p. 2481–2495, 2017. Citado 2 vezes nas páginas and 23.

BALDI, P. Autoencoders, unsupervised learning, and deep architectures. In: *Proceedings of ICML workshop on unsupervised and transfer learning*. [S.l.: s.n.], 2012. p. 37–49. Citado na página 33.

BENGIO, Y.; GOODFELLOW, I. J.; COURVILLE, A. Deep learning. *Nature*, v. 521, p. 436–444, 2015. Citado 2 vezes nas páginas 21 and 41.

BOCCARDI, M.; BOCCHETTA, M.; MORENCY, F. C.; COLLINS, D. L.; NISHIKAWA, M.; GANZOLA, R.; GROTHE, M. J.; WOLF, D.; REDOLFI, A.; PIEVANI, M. *et al.* Training labels for hippocampal segmentation based on the eadc-adni harmonized hippocampal protocol. *Alzheimer's & Dementia*, Elsevier, v. 11, n. 2, p. 175–183, 2015. Citado 2 vezes nas páginas 26 and 27.

BRETT, M.; CHRISTOFF, K.; CUSACK, R.; LANCASTER, J. *et al.* Using the talairach atlas with the mni template. *Neuroimage*, Elsevier Science, v. 13, n. 6, p. 85–85, 2001. Citado na página 27.

CARMO, D.; SILVA, B.; YASUDA, C.; RITTNER, L.; LOTUFO, R. Extended 2d volumetric consensus hippocampus segmentation. In: *International Conference on Medical Imaging with Deep Learning*. [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 22 and 67.

CARMO, D.; SILVA, B.; YASUDA, C.; RITTNER, L.; LOTUFO, R. Extended 2d volumetric consensus hippocampus segmentation. *arXiv preprint arXiv:1902.04487*, 2019. Citado na página 26.

CHEN, C.-h. *Handbook of pattern recognition and computer vision*. [S.l.]: World Scientific, 2015. Citado 2 vezes nas páginas 35 and 39.

CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 801–818. Citado 3 vezes nas páginas , 23, and 66.

CHEN, Y.; SHI, B.; WANG, Z.; ZHANG, P.; SMITH, C. D.; LIU, J. Hippocampus segmentation through multi-view ensemble convnets. In: IEEE. *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. [S.l.], 2017. p. 192–196. Citado 2 vezes nas páginas 19 and 24.

CHINCARINI, A.; SENSI, F.; REI, L.; GEMME, G.; SQUARCIA, S.; LONGO, R.; BRUN, F.; TANGARO, S.; BELLOTTI, R.; AMOROSO, N. *et al.* Integrating longitudinal information in hippocampal volume measurements for the early detection of alzheimer's disease. *NeuroImage*, Elsevier, v. 125, p. 834–847, 2016. Citado 2 vezes nas páginas 23 and 56.

DINSDALE, N. K.; JENKINSON, M.; NAMBURETE, A. I. Spatial warping network for 3d segmentation of the hippocampus in mr images. In: SPRINGER. *International Conference on Medical Image Computing and Computer-Assisted Intervention*. [S.l.], 2019. p. 284–291. Citado 2 vezes nas páginas 24 and 26.

DISORDERS, N. I. of N.; COMMUNICATIONS, S. U. O. of; LIAISON, P. *The Epilepsies and Seizures: Hope through Research*. [S.l.]: Department of Health & Human Services, NIH, National Institute of . . . , 2015. Citado na página 18.

DOUGHERTY, E. R.; LOTUFO, R. A. *Hands-on morphological image processing*. [S.l.]: SPIE press, 2003. v. 59. Citado na página 46.

FALK, T.; MAI, D.; BENSCH, R.; ÇIÇEK, Ö.; ABDULKADIR, A.; MARRAKCHI, Y.; BÖHM, A.; DEUBNER, J.; JÄCKEL, Z.; SEIWALD, K. *et al.* U-net: deep learning for cell counting, detection, and morphometry. *Nature methods*, Nature Publishing Group, v. 16, n. 1, p. 67, 2019. Citado na página 22.

FISCHL, B. Freesurfer. *Neuroimage*, Elsevier, v. 62, n. 2, p. 774–781, 2012. Citado 6 vezes nas páginas 17, 18, 23, 25, 56, and 58.

GHIZONI, E.; ALMEIDA, J.; JOAQUIM, A. F.; YASUDA, C. L.; CAMPOS, B. M. de; TEDESCHI, H.; CENDES, F. Modified anterior temporal lobectomy: anatomical landmarks and operative technique. *Journal of Neurological Surgery Part A: Central European Neurosurgery*, Georg Thieme Verlag KG, v. 76, n. 05, p. 407–414, 2015. Citado 4 vezes nas páginas 17, 18, 27, and 29.

GHIZONI, E.; MATIAS, R. N.; LIEBER, S.; CAMPOS, B. M. de; YASUDA, C. L.; PEREIRA, P. C.; FILHO, A. C. S. A.; JOAQUIM, A. F.; LOPES, T. M.; TEDESCHI, H. *et al.* Clinical and imaging evaluation of transuncus selective amygdalohippocampectomy. *World neurosurgery*, Elsevier, v. 100, p. 665–674, 2017. Citado 2 vezes nas páginas 17 and 29.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1026–1034. Citado 2 vezes nas páginas 33 and 49.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado 2 vezes nas páginas 20 and 33.

IGLESIAS, J. E.; SABUNCU, M. R. Multi-atlas segmentation of biomedical images: a survey. *Medical image analysis*, Elsevier, v. 24, n. 1, p. 205–219, 2015. Citado na página 23.

IGLOVIKOV, V.; SHVETS, A. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018. Citado na página 33.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. Citado na página 33.

ISENSEE, F.; KICKINGEREDER, P.; WICK, W.; BENDSZUS, M.; MAIER-HEIN, K. H. Brain tumor segmentation and radiomics survival prediction: contribution to the brats 2017 challenge. In: SPRINGER. *International MICCAI Brainlesion Workshop*. [S.l.], 2017. p. 287–297. Citado 7 vezes nas páginas 22, 26, 47, 55, 56, 57, and 58.

KAMNITSAS, K.; BAI, W.; FERRANTE, E.; MCDONAGH, S.; SINCLAIR, M.; PAWLOWSKI, N.; RAJCHL, M.; LEE, M.; KAINZ, B.; RUECKERT, D. *et al.* Ensembles of multiple models and architectures for robust brain tumour segmentation. In: SPRINGER. *International MICCAI Brainlesion Workshop*. [S.l.], 2017. p. 450–462. Citado na página 22.

KERVADEC, H.; BOUCHTIBA, J.; DESROSIERS, C.; GRANGER, E.; DOLZ, J.; Ben Ayed, I. Boundary loss for highly unbalanced segmentation. In: CARDOSO, M. J.; FERAGEN, A.; GLOCKER, B.; KONUKOGLU, E.; OGUZ, I.; UNAL, G.; VERCAUTEREN, T. (Ed.). *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*. London, United Kingdom: PMLR, 2019. (Proceedings of Machine Learning Research, v. 102), p. 285–296. Disponível em: <http://proceedings.mlr.press/v102/kervadec19a.html>. Citado 2 vezes nas páginas 43 and 44.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 45.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado 2 vezes nas páginas 21 and 25.

KUMAR, S. K. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017. Citado na página 33.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Research, v. 521, n. 7553, p. 436–444, 2015. Citado na página 21.

LEVI, G.; HASSNER, T. Age and gender classification using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2015. p. 34–42. Citado na página 21.

LIU, L.; JIANG, H.; HE, P.; CHEN, W.; LIU, X.; GAO, J.; HAN, J. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019. Citado na página 46.

LUCENA, O.; SOUZA, R.; RITTNER, L.; FRAYNE, R.; LOTUFO, R. Silver standard masks for data augmentation applied to deep-learning-based skull-stripping. In: IEEE. *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*. [S.l.], 2018. p. 1114–1117. Citado 2 vezes nas páginas 26 and 31.

MANJÓN, J. V.; COUPÉ, P. volbrain: an online mri brain volumetry system. *Frontiers in neuroinformatics*, Frontiers, v. 10, p. 30, 2016. Citado 2 vezes nas páginas 27 and 30.

MARTIN, J. Lymbic system and cerebral circuits for emotions, learning, and memory. *Neuroanatomy: text and atlas (third ed.). McGraw-Hill Companies*, p. 382, 2003. Citado na página 18.

MCCARTHY, C. S.; RAMPRASHAD, A.; THOMPSON, C.; BOTTI, J.-A.; COMAN, I. L.; KATES, W. R. A comparison of freesurfer-generated data with and without manual intervention. *Frontiers in neuroscience*, Frontiers, v. 9, p. 379, 2015. Citado na página 17.

MEHTA, R.; SIVASWAMY, J. M-net: A convolutional neural network for deep brain structure segmentation. In: IEEE. *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. [S.l.], 2017. p. 437–440. Citado na página 22.

NOGOVITSYN, N.; SOUZA, R.; MULLER, M.; SRAJER, A.; HASSEL, S.; ARNOTT, S. R.; DAVIS, A. D.; HALL, G. B.; HARRIS, J. K.; ZAMYADI, M. *et al.* Testing a deep convolutional neural network for automated hippocampus segmentation in a longitudinal sample of healthy participants. *NeuroImage*, Elsevier, v. 197, p. 589–597, 2019. Citado na página 25.

NOSEK, B. A.; ALTER, G.; BANKS, G. C.; BORSBOOM, D.; BOWMAN, S. D.; BRECKLER, S. J.; BUCK, S.; CHAMBERS, C. D.; CHIN, G.; CHRISTENSEN, G. *et al.* Promoting an open research culture. *Science*, American Association for the Advancement of Science, v. 348, n. 6242, p. 1422–1425, 2015. Citado na página 19.

OKTAY, O.; SCHLEMPER, J.; FOLGOC, L. L.; LEE, M.; HEINRICH, M.; MISAWA, K.; MORI, K.; MCDONAGH, S.; HAMMERLA, N. Y.; KAINZ, B. *et al.* Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. Citado 3 vezes nas páginas 21, 22, and 66.

PASZKE, A.; GROSS, S.; CHINTALA, S.; CHANAN, G.; YANG, E.; DEVITO, Z.; LIN, Z.; DESMAISON, A.; ANTIGA, L.; LERER, A. Automatic differentiation in pytorch. 2017. Citado na página 44.

PEREIRA, M.; LOTUFO, R.; RITTNER, L. An extended-2d cnn approach for diagnosis of alzheimer's disease through structural mri. In: *Proceedings of the 27th Annual Meeting of ISMRM 2019.* [S.l.: s.n.], 2019. p. na. Citado 2 vezes nas páginas 33 and 35.

PETERSEN, R. C.; AISEN, P.; BECKETT, L. A.; DONOHUE, M.; GAMST, A.; HARVEY, D. J.; JACK, C.; JAGUST, W.; SHAW, L.; TOGA, A. *et al.* Alzheimer's disease neuroimaging initiative (adni): clinical characterization. *Neurology*, AAN Enterprises, v. 74, n. 3, p. 201–209, 2010. Citado 4 vezes nas páginas 17, 18, 24, and 27.

PINHEIRO, G.; CARMO DIEDRE, Y. C.; LOTUFO, R.; RITTNER, L. Convolutional neural network on dti data for sub-cortical brain structure segmentation. In: *International*

Workshop on Computational Diffusion MRI from the 22nd International Conference on Medical Image Computing and Computer Assisted Intervention. [S.l.: s.n.], 2019. Citado na página 67.

PIPITONE, J.; PARK, M. T. M.; WINTERBURN, J.; LETT, T. A.; LERCH, J. P.; PRUESSNER, J. C.; LEPAGE, M.; VOINESKOS, A. N.; CHAKRAVARTY, M. M.; INITIATIVE, A. D. N. *et al.* Multi-atlas segmentation of the whole hippocampus and subfields using multiple automatically generated templates. *Neuroimage*, Elsevier, v. 101, p. 494–512, 2014. Citado na página 23.

PLATERO, C.; TOBAR, M. C. Combining a patch-based approach with a non-rigid registration-based label fusion method for the hippocampal segmentation in alzheimer's disease. *Neuroinformatics*, Springer, v. 15, n. 2, p. 165–183, 2017. Citado 3 vezes nas páginas 23, 56, and 58.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241. Citado 8 vezes nas páginas , 19, 21, 22, 24, 26, 32, and 49.

ROY, A. G.; CONJETI, S.; NAVAB, N.; WACHINGER, C.; INITIATIVE, A. D. N. *et al.* Quicknat: A fully convolutional network for quick and accurate segmentation of neuroanatomy. *NeuroImage*, Elsevier, v. 186, p. 713–727, 2019. Citado 6 vezes nas páginas 24, 25, 26, 56, 57, and 58.

SHIN, H.-C.; ROTH, H. R.; GAO, M.; LU, L.; XU, Z.; NOGUES, I.; YAO, J.; MOLLURA, D.; SUMMERS, R. M. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, IEEE, v. 35, n. 5, p. 1285–1298, 2016. Citado na página 36.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado 2 vezes nas páginas 20 and 33.

SOHER, B. J.; DALE, B. M.; MERKLE, E. M. A review of mr physics: 3t versus 1.5 t. *Magnetic resonance imaging clinics of North America*, Elsevier, v. 15, n. 3, p. 277–290, 2007. Citado na página 39.

SOUZA, R.; LUCENA, O.; BENTO, M.; GARRAFA, J.; APPENZELLER, S.; RITTNER, L.; LOTUFO, R.; FRAYNE, R. Reliability of using single specialist annotation for designing and evaluating automatic segmentation methods: A skull stripping case study. In: IEEE. *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*. [S.l.], 2018. p. 1344–1347. Citado na página 39.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 64.

SUDRE, C. H.; LI, W.; VERCAUTEREN, T.; OURSELIN, S.; CARDOSO, M. J. Generalised dice overlap as a deep learning loss function for highly unbalanced

segmentations. In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. [S.l.]: Springer, 2017. p. 240–248. Citado 3 vezes nas páginas 24, 41, and 42.

SUN, Y.; CHEN, Y.; WANG, X.; TANG, X. Deep learning face representation by joint identification-verification. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 1988–1996. Citado na página 21.

SUNDERMEYER, M.; SCHLÜTER, R.; NEY, H. Lstm neural networks for language modeling. In: *Thirteenth annual conference of the international speech communication association*. [S.l.: s.n.], 2012. Citado na página 23.

TANG, J.; LI, J.; XU, X. Segnet-based gland segmentation from colon cancer histology images. In: IEEE. *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. [S.l.], 2018. p. 1078–1082. Citado na página 23.

THYREAU, B.; SATO, K.; FUKUDA, H.; TAKI, Y. Segmentation of the hippocampus by transferring algorithmic knowledge for large cohort processing. *Medical image analysis*, Elsevier, v. 43, p. 214–228, 2018. Citado 7 vezes nas páginas 19, 24, 25, 26, 56, 57, and 58.

WACHINGER, C.; REUTER, M.; KLEIN, T. Deepnat: Deep convolutional neural network for segmenting neuroanatomy. *NeuroImage*, Elsevier, v. 170, p. 434–445, 2018. Citado 3 vezes nas páginas 19, 24, and 25.

WANG, H.; SUH, J. W.; DAS, S. R.; PLUTA, J. B.; CRAIGE, C.; YUSHKEVICH, P. A. Multi-atlas segmentation with joint label fusion. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 35, n. 3, p. 611–623, 2013. Citado na página 23.

XIAO, W.-T.; CHANG, L.-J.; LIU, W.-M. Semantic segmentation of colorectal polyps with deeplab and lstm networks. In: IEEE. *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*. [S.l.], 2018. p. 1–2. Citado na página 23.

XIE, Z.; GILLIES, D. Near real-time hippocampus segmentation using patch-based canonical neural network. *arXiv preprint arXiv:1807.05482*, 2018. Citado 3 vezes nas páginas 19, 24, and 25.

YUSHKEVICH, P. A.; PIVEN, J.; HAZLETT, H. C.; SMITH, R. G.; HO, S.; GEE, J. C.; GERIG, G. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, v. 31, n. 3, p. 1116–1128, 2006. Citado na página 30.