UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Analysis of medical images to support decision-making in the musculoskeletal field

Jonathan da Silva Ramos

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Jonathan da Silva Ramos

Analysis of medical images to support decision-making in the musculoskeletal field [1]

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Professor D.Sc. Agma Juci Machado Traina

USP – São Carlos
August 2021

[1] Funded by FAPESP (Process 2017/23780-2) and CAPES (Process 0487/17083480).

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

da Silva Ramos, Jonathan
d175a Analysis of medical images to support decision-making in the musculoskeletal field / Jonathan da Silva Ramos; orientadora Agma Juci Machado Traina. -- São Carlos, 2021.
141 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2021.

1. Image segmentation. 2. magnetic resonance imaging. 3. vertebral fragility fracture. I. Juci Machado Traina, Agma , orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Jonathan da Silva Ramos

Análise de imagens do músculo esquelético para apoio a tomada de decisão[1]

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Agma Juci Machado Traina

USP – São Carlos
Agosto de 2021

[1] Financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP – Processo 2017/23780-2) e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES – Processo 0487/17083480).

I dedicate this work to my brother Johnny Ramos (in memoriam),
whose existence was indispensable in my life.

ACKNOWLEDGEMENTS

This work was funded by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (Process 2017/23780-2 and 2016/17078-0), and by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil) - financial code 001 (Process 0487/17083480)

First and foremost, I would like to praise and thank the almighty God, who has granted me innumerous blessings, knowledge, and opportunities, so that I have been finally able to accomplish this thesis.

To my parents, Maria Aparecida and Rosalvo, for all the caring and support throughout the conduction of this research. Also, to my sisters, who always helped me stay positive in tough times. To all my relatives who never let me give up.

To my advisor, Professor Ph.D. Agma Traina, who taught me so much that I can not even numerate them all. To every professor, physician, researcher, and student who crossed my path and helped me grow intellectually and personally.

To Professor M.D. Marcello Nogueira-Barbosa who was so patient and caring with this project, aiding in so many ways with all the support necessary.

“As grandes idéias surgem da observação dos pequenos detalhes.”

“Big ideas come from observing small details.”
(Augusto Cury)

RESUMO

RAMOS, J. S. **Análise de imagens do músculo esquelético para apoio a tomada de decisão**. 2021. 139 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

Sistemas de diagnóstico auxiliado por computador, recuperação de imagens por conteúdo e a abordagem radiômica são grandes aliados no auxílio à tomada de decisão. Contudo, um passo comum entre estas abordagens é a segmentação de uma região de interesse, por exemplo, um corpo vertebral. Desenhar manualmente, de uma forma precisa, os contornos de um corpo vertebral é uma tarefa demorada e trabalhosa, tornando-se impraticável quando há uma quantidade considerável de exames a segmentar. Como consequência, ferramentas semiautomáticas, com uma interação manual mínima, se torna atrativa e impulsiona uma alta demanda para o lado computacional. Deseja-se que, em algum ponto, a interação seja mínima, com pouca ou nenhuma correção manual. Este doutorado tem a seguinte hipótese: "A segmentação de corpos vertebrais em RM pode ser realizada de forma computacionalmente rápida, com interação manual reduzida e, ao mesmo tempo, apresentando resultados precisos." A validação desta hipótese foi realizada em três cenários. No primeiro cenário, consideramos a segmentação de corpos vertebrais fraturados com apenas um corte de RM, no qual foi proposto o método BGrowth, o qual apresentou uma acurácia de 96.1% com um rápido tempo de processamento; No segundo cenário, trabalhamos com a segmentação de RM volumétricas, que é uma tarefa ainda mais desafiadora devido à considerável quantidade de cortes presentes nestes exames. Foi desenvolvida uma família de métodos de segmentação, apresentando-se abordagens cada vez mais rápidas e com menor interação manual. Ao final, apenas dois ou três cortes precisaram ser anotados, entre cerca de 100 cortes, atingindo 94% de *F-Measure*. Para tal, foi proposta a estimativa de anotações nos cortes intermediários (EANIS), juntamente com o método FastCleverSeg, o qual apresentou o tempo de execução mais rápido e, ao mesmo tempo, resultado igual ou superior aos competidores; No terceiro cenário, realizamos a análise de pacientes com fraturas por fragilidade usando exames de RM e a abordagem radiômica. Foi proposto um método de análise de textura (BEAUT), o qual obteve uma AUC de 97% na diferenciação de pacientes com e sem fratura por fragilidade. Portanto, este doutorado contribuiu para o estado-da-arte introduzindo métodos de segmentação que apresentaram resultados promissores nos três cenários mencionados acima. As contribuições dessa tese têm um potencial para auxiliar no processo de tomada de decisão e na produção de padrão ouro para o treinamento de modelos de aprendizado profundo.

Palavras-chave: Segmentação de imagens, ressonância magnética, fratura vertebral.

ABSTRACT

RAMOS, J. S. **Analysis of medical images to support decision-making in the musculoskeletal field**. 2021. 139 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

Computer-aided diagnosis, computer-based image retrieval systems, and the radiomics approach are great allies to aid in decision making. However, an ordinary and laborious step in those approaches is the segmentation of a region of interest, for example, a vertebral body. Unfortunately, manually drawing accurate and precise boundaries is time-consuming and impractical to perform for many exams. Consequently, semi-automatic segmentation tools, with minimal interaction, pose high and attractive demand to the computational end. The greater goal is that, at some point, the physician intervention in the segmentation would be minimal with just a few or even no manual corrections. This doctorate research has the following hypothesis: "The segmentation of vertebral bodies in MRI can be performed computationally faster with easier manual interaction and, at the same time, producing accurate results." We evaluate this hypothesis in three application scenarios as follows. First, we dealt with the challenging task of segmenting vertebral compression fractures in single MRI slices. We proposed Balanced Growth (BGrowth), which achieved 96.1% accuracy while keeping fast run-time performance. Second, we stepped into the segmentation of volumetric spine MRI exams, which is even more challenging due to several slices in the exams. We came up with a family of segmentation methods, presenting faster approaches with less manual interaction. Our final solution required annotating only two or three slices (among about 100 slices) and achieved 94% of F-Measure. To do so, we proposed the Estimation of ANnotation on Intermediary Slices (EANIS) along with the Fast Clever Segmentation (FastCleverSeg) method. Our approach was the fastest one and, at the same time, presented results similar to or better than the competitors. Finally, we assessed patients with bone fragility fractures using spine MRI and the radiomics approach. We proposed BonE Analysis Using Texture (BEAUT), which achieved 97% AUC for differentiating patients with and without vertebral body fragility fracture. Therefore, this doctorate research contributed to the state-of-the-art by introducing segmentation methods that presented promising results in the three scenarios mentioned above. We firmly believe that our contributions have a high potential to aid in the decision-making process and producing more ground truths for the training of deep learning models.

Keywords: Image segmentation, magnetic resonance imaging, vertebral fragility fracture.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SOURCE CODES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

2D	2-Dimensional
3D	3-Dimensional
ACC	Accuracy
AUC	Area Under the Receiver Operator Characteristic Curve
AVD	Average Hausdorff 's Distance
BEAUT	BonE Analysis Using Texture
BGrowth	Balanced Growth
EANIS	Estimating ANnotation on Intermediary Slices
FM	F-Measure
IAN	Inside Annotation
IFT	Image Foresting Transform
JAC	Jaccard Coefficient
k-NN	k-Nearest Neighbor
PAS	Percentage of Annotated Slices
OAN	Outside Annotation
PRE	Precision
SEN	Sensitivity
3DBGrowth	3D Balanced Growth
ANN	Artificial Neural Networks
BMD	Bone mineral density
CAD	Computer-Aided Diagnosis
CBIR	Computer-Based Image Retrieval
CT	Computed Tomography
CT	Computerized Tomography
DEXA	Dual-Energy X-ray Absorptiometry
DF	Distance Function
DICOM	Digital Imaging and Communication in Medicine
DL	Deep Learning
FEM	Feature Extraction Method
FN	False Negative
FP	False Positive

FPR	False Positive Rate
GT	Ground-Truth
GUI	Graphical User Interface
LBP	Low Back Pain
MAE	Mean Absolute Error
MRI	Magnetic Resonance Imaging
NSC	Negative Slope Coefficient
PACS	Picture Archiving and Communication Systems
QDA	Quadratic Discriminant Analysis
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
ST	SegmenTation
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
TPR	True Positive Rate
VCF	Vertebral Compression Fractures

LIST OF SYMBOLS

$\mathbb{Z}$ — Set of integer numbers.

$f(x,y)$ — Bi-dimensional digital image.

(x,y) — Entry coordinates of a bi-dimensional matrix.

r — Number of rows of a bi-dimensional digital image.

c — Number of columns of a bi-dimensional digital image.

$g(x,y)$ — Bi-dimensional digital image after reducing range of gray intensities.

$A \oplus B$ — Dilation operation between sets A and B.

$A \ominus B$ — Erosion operation between sets A and B.

$A \cdot B$ — Closing operation between sets A and B.

ξ — Represents a feature extraction method.

$\mathbb{S}$ — Complex objects domain.

s — Element in the complex objects domain ($\mathbb{S}$).

$\mathbb{V}$ — Feature vector domain.

v — Element in the feature vector domain ($\mathbb{V}$).

δ — Metric.

$< \mathbb{V}, \delta >$ — Metric space.

n — dimension of an object, element, or feature vector.

$\xi(v_a, v_b)$ — Distance between feature vectors v_a and v_b.

I — Digital image or exam represented as a matrix.

L — Matrix containing annotations or labels regions for a given exam (I).

W — Matrix containing weights for a given annotation matrix (L).

CONTENTS

CHAPTER

1

INTRODUCTION

Long before any technology took its place, humans have naturally used visual signs to communicate with each other, for example, through smoke signals and rustic drawings in the caves. As the technology evolved, we were able to capture and store digital images. Since then, we have applied digital images in several tasks, such as communication, security, and medicine. Specifically, in medicine, images from several exams have a relevant role in diagnosing and decision-making. In this case, Computer-Aided Diagnosis (CAD) and Computer-Based Image Retrieval (CBIR) systems are pioneers in supporting decision-making using images (GROBEL; TELLES, 2014; RAVAL; ZAFAR, 2013; GODINHO *et al.*, 2016).

A CAD system outputs a second-opinion based on quantitative analysis of the exam. In contrast, CBIR systems retrieve images based on their content. Thus, for example, given an exam with an anomaly, the system retrieves previous similar cases. However, medical imaging has progressively shifted from conventional visual image analysis to quantitative personalized medicine. Thus, CAD systems, which focus on a single output (as lesion or cancer), may not suit properly nowadays requirements. For this reason, the radiomics approach, which is a quantitative technique for medical imaging, was designed to improve quality and precision of models that aid decision-making (DURON *et al.*, 2019; KUMAR *et al.*, 2012; AHMAD *et al.*, 2019; XU *et al.*, 2016).

Radiomics combines the extraction of hundreds or thousands of features from the images and data from the patients not directly acquired from the images. The aim is to enhance the existing data to detect disease-specific processes that are not accessible through traditional visual inspection of the generated images. Therefore, there is no automation of the diagnostic processes. Instead, radiomics supports existing ones with additional information (TIMMEREN *et al.*, 2020).

1.1 Motivation

The lumbar or Low Back Pain (LBP) is a common complaint worldwide affecting people of all ages, compromising the quality of life, causing work absence (impairing mobility), imposing a high economic burden on individuals, families, communities, industry, and governments. Such issues and pain highlights the importance of caring for the human musculoskeletal system. According to the World Health Organization[1], over 80 percent of the world population may experience LBP at some time during life. Aging raises the prevalence or incidence of LBP, growing surgical treatments, requiring accurate and fast diagnoses to prevent complications. In general, to detect and diagnose several spinal diseases, Magnetic Resonance Imaging (MRI) exams frequently employed. Usually, computational approaches to support the decision-making (such as CAD, CBIR, and radiomics approach) limit/crop each exam to a region or volume of interest. In this way, digital segmentation can be applied, allowing a faster and more objective analysis of the issues, assisting in the characterization and quantification of abnormalities, surgery planning, and evaluation of suitable treatments (FATOYE; GEBRYE; ODEYEMI, 2019; FREBURGER *et al.*, 2009).

In this context, the digital segmentation of a region of interest in a medical image plays a significant role and can be performed manually, semi-automatically (sometimes with manual correction), or fully automatically (by specialized methods, often using deep learning). In manual segmentation, drawing accurate and precise boundaries is time-consuming and impractical to perform for many exams. Besides, manual segmentation may considerably suffer from observer bias, presenting intra- and inter-observer variations. In fully automatic segmentation, the use of traditional methods takes too much processing time, requires the support of powerful computing resources, and lacks consistent and precise results. Alternatively, automatic segmentation using deep learning approaches requires thousands of samples whose production requires laborious work, due to the issues above. On the other hand, semi-automatic segmentation tools, become highly attractive due to two main reasons (WANG *et al.*, 2017; HAMMERNIK *et al.*, 2015; KOREZ *et al.*, 2016; GAONKAR *et al.*, 2017):

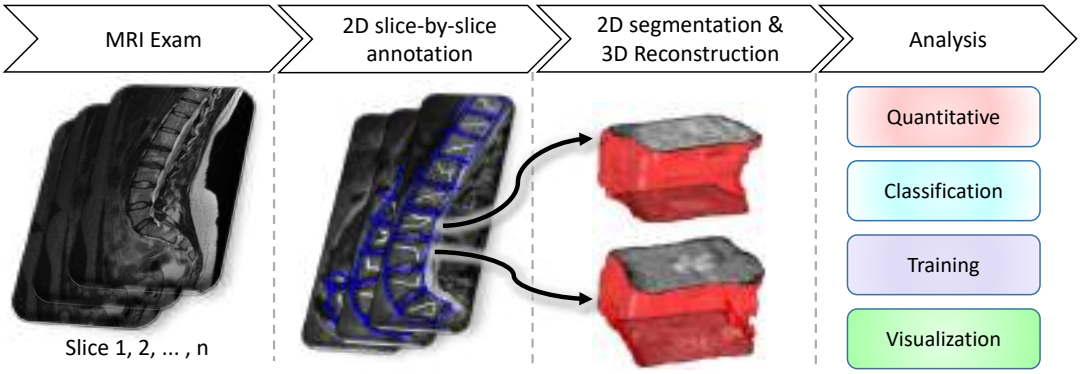
1. Reduction of the labor-intensiveness of the manual approach by using minimal user input;
2. Easier production of more samples allows scientists to collect a large amount of reference segmentations to be further used to train deep learning models to automatically and accurately segment new cases.

Moreover, the semi-automatic segmentation aids in several analysis, as depicted in Figure 1 and detailed as follows:

Quantitative: measures can be extracted, such as shape, texture, color (JUNIOR *et al.*, 2018).

[1] World Health Organization: <https://www.who.int/>

Figure 1 – Steps in a semi-automatic segmentation schema for volumetric MRI.



Source: Adapted from Ramos *et al.* (2019a).

Classification: machine learning techniques can be applied to characterize a given anomaly (CAD systems) or to retrieve similar cases (CBIR systems) (XUE *et al.*, 2011; CAZZOLATO *et al.*, 2019).

Training: Interactive segmentation tools can be meaningful during the education of new radiologists (KARIMI *et al.*, 2018). Students can learn how to correctly segment each vertebra and to detect spine pathologies (STEFAN *et al.*, 2018). This kind of training may avoid potential medical failures, which reduces further complications.

Visualization: 3D human structures can be visualized and used for simulation of medical and surgical procedures (BANERJEE *et al.*, 2017).

Semi-automatic segmentation methods enormously benefit physicians, as well as scientists (BURIAN *et al.*, 2019). Consequently, at some point, the physician intervention in the segmentation would be minimal with just a few or even no manual corrections. This Doctorate research investigated the following question: "*How can we ease manual interaction in semi-automatic segmentation of vertebral bodies in MRI exams and, at the same time, keep or improve segmentation quality?*". The solution to this research question is challenging due to vertebrae deformations/fractures, diversity of MRI acquisition parameters, the high computational cost of segmentation algorithms (mainly in volumetric exams) and how to embed the expert knowledge acquired over several years of experience without adding observer-bias.

1.2 Problem Statement

As previously mentioned, in this Doctorate research, we aim at advancing in the semi-automatic segmentation and analysis of vertebral bodies obtained by MRI, focusing on reducing the manual interaction. To this end, we propose novel segmentation methods and brand-new techniques to reduce the time spent in laborious manual tasks. Considering the diversity and

complexity of problems that may arise in this scenario, we propose this Doctorate thesis with the following hypothesis.

Hypothesis

The segmentation of vertebral bodies in MRI can be performed computationally faster with easier manual interaction and, at the same time, producing accurate results.

Our thesis's investigation considered data obtained from two distinct sources:

1. Exams obtained from the partners at the Ribeirao Preto Medical School Clinics Hospital - USP (HCFMRP), within the Thematic FAPESP project "*Mining, Indexing and Visualizing Big Data in Clinical Decision Support Systems*" – MIVIsBD[2], to which this Doctorate is a part of;
2. Public domain databases, such as from Zukić *et al.* (2014), Burian *et al.* (2019), Chu *et al.* (2015a), and Chu *et al.* (2015b).

Accordingly, we support our previously mentioned thesis by proposing methods to solve the three following research problems.

Research Problem 1. Segmenting fractured or deformed vertebral bodies in single slices.

A very recurrent disease among older adults is Vertebral Compression Fractures (VCF), which, in general, is caused by bone-weakening diseases. The literature reports the works (BARBIERI *et al.*, 2015; CASTI *et al.*, 2017; FRIGHETTO-PEREIRA *et al.*, 2015; FRIGHETTO-PEREIRA *et al.*, 2015; AZEVEDO-MARQUES *et al.*, 2015; FRIGHETTO-PEREIRA *et al.*, 2016a) that employ computational methods for VCF analysis from MRIs exams. The works used, unless stated otherwise, a database with 63 T1 weighted exams (detailed in Section 3.2.1). The database was divided into 102 lumbar vertebral bodies with VCFs (53 benign and 49 malignant) and 106 healthy lumbar vertebral bodies. Reference manual segmentations were performed in the central or most representative sagittal slice of the exam. The works can be divided into segmentation and classification:

Segmentation: In general, VCFs segmentation is a challenging task due to non-homogeneous gray intensities within the same vertebral body, which hardens boundaries delineation. The literature presents the works (BARBIERI *et al.*, 2015; CASTI *et al.*, 2017) that dealt with the challenge of VCF segmentation. Both techniques used only one seed point (a mouse click) approximately in the center of the vertebral body in a single slice.

Classification: the works (FRIGHETTO-PEREIRA *et al.*, 2015; FRIGHETTO-PEREIRA *et al.*, 2015; AZEVEDO-MARQUES *et al.*, 2015; FRIGHETTO-PEREIRA *et al.*,

[2] FAPESP Process Number 2016/17078-0.

2016a) dealt with the classification of VCFs using manually segmented vertebral bodies in 2D (slices) exams. Casti *et al.* (2017) proposed a cooperative strategy that exploited the dynamic contribution of the classification models, presenting an AUC of 94% using QDA classifier and considering the reference manual segmentation (ground-truth).

Although both segmentation works of Barbieri *et al.* (2015) and Casti *et al.* (2017) have presented promising results, the best classification reported results still depends on the ground-truth. Aimed at investigating the challenge of VCFs segmentation, we formulated the following question:

a) *How can we segment fractured vertebral bodies from MRI exams with accurate boundary delineation with an easier and faster manual interaction than the state-of-the-art approaches?*

Research Problem 2. Reducing manual interaction on the segmentation of lumbar vertebral bodies in volumetric MRI exams.

The segmentation of vertebral anatomical structures, such as paravertebral muscle, intervertebral discs, and vertebral bodies, supports measurement, quantification, and analysis of diseases related to the spine. In this context, segmentation tools, for example 3D Slicer[3] (KIKINIS; PIEPER; VOSBURGH, 2014), which provides friendly Graphical User Interface (GUI) are highly attractive. However, the semi-automatic segmentation algorithms commonly used in this GUI, frequently, require too much time from the radiologists on manual annotation of a few or all the slices in the exam (KOREZ *et al.*, 2016; GAONKAR *et al.*, 2017; ENGSTROM *et al.*, 2011; HILLE *et al.*, 2018, 2018; RAMOS *et al.*, 2019a). One example of such method is GrowCut (VEZHNEVETS; KONOUCHINE, 2005), which has been widely employed on the segmentation of medical imaging. Also, a faster but less accurate version of GrowCut, Fast GrowCut, has been widely used for segmentation of medical images in the 3D Slicer application. Considering this scenario, we formulated the following questions:

a) *How can we reduce the total time and effort spent on manual annotations of vertebral anatomical structures in MRI exams and maintain segmentation accuracy?*

Research Problem 3. Assessing bone fragility fractures using volumetric MRI and radiomic texture features.

According to the International Osteoporosis Foundation (IOF)[4], one in three women and one in five men over fifty years old will suffer a broken bone due to osteoporosis worldwide.

[3] <https://www.slicer.org/>
[4] IOF: <https://www.osteoporosis.foundation/>

Thus, osteoporosis is a major public health issue, which slowly evolves as an asymptomatic condition, with increasing probabilities of developing fragility fractures (MCCLUNG, 2005; LÓPEZ *et al.*, 2016). Studies have shown that a previous spine fracture increases the risk of future hip fracture, which in turn is associated with increased morbidity and mortality in elderly people (TELLACHE *et al.*, 2009; HAMMERNIK *et al.*, 2015).

As one of the most common fracture occurs in the vertebra (MEHTA; SEBRO, 2020; LÓPEZ *et al.*, 2016), predicting occurrence of this injury is crucial for prevention and early treatment. Bone mineral density (BMD) is one of the reference indicators employed to diagnose osteoporosis, and is associated to an increased risk of fractures, including lumbar and hip fractures (MCCLUNG, 2005; MEHTA; SEBRO, 2020; WU *et al.*, 2020). Among other bone measurements employed to estimate bone strength are the bone geometry, cortical porosity, collagen properties, bone turnover rates, trabecular microarchitecture, and percentage of microdamage, and bone marrow adiposity (MACIEL *et al.*, 2020). BMD, evaluated by Dual-Energy X-ray Absorptiometry (DEXA) imaging or bone densitometry, often fails on predicting osteoporosis fragility fractures (CHANG *et al.*, 2017), as well as bone mass measurements alone are insufficient to faithfully predict all osteoporotic fragility fractures (MEHTA; SEBRO, 2020). However, DEXA is still employed as an initial, but insufficient, step in the evaluation of new drugs efficacy (MACIEL *et al.*, 2020; WU *et al.*, 2020).

Studies from the literature have used intrinsic information extracted from Computerized Tomography (CT) scans and MRI to detect osteoporosis, combining the image features with bone measurements and structural information (TELLACHE *et al.*, 2009; WU *et al.*, 2020; CHANG *et al.*, 2017). Maciel *et al.* (2020) reported 13 texture attributes extracted from L3 vertebrae in MRIs that are strongly correlated to BMD, highlighting the potential relevance of texture features for fragility fracture assessment. Wu *et al.* (2020) reported a negative correlation of BMD with bone quantitative measurements in MRIs, which can potentially assess the severity of lumbar osteoporosis for scientific research. However, to the best of our knowledge, the literature lacks studies that address only feature-based MRI analysis for the classification of patients with fragility fractures risk. New techniques, such as radiomics, are necessary to achieve better prediction of patients at risk of developing fragility fractures. Previous literature showed that spine MRI texture features correlate well with DEXA measurements. We employed our proposed methods to segment and analyze bone fragility fractures caused by osteoporosis. We aimed at evaluating vertebral bone fragility using texture analysis and formulated the following questions:

a) *Can we assess bone fragility secondary to osteoporosis using MRI?*

b) *Is it possible to find potential biomarkers to predict bone fragility fractures?*

1.3 Contributions

The main contributions of this Doctorate research are three-fold, each of which addressing one of the aforementioned research problems.

Contribution 1. Segmenting fractured or deformed vertebral bodies in single slices.

We have investigated the challenge of Vertebral Compression Fractures (VCFs) segmentation and we proposed an efficient and accurate segmentation approach. Based on the motto "unity is strength", we take into account that all pixels in the path of an expanding region have influence on their growth. We achieve "unity" by balancing the weights along the growing path, so that the final segmentation is an "unanimity" among the pixels. This allows small intensities transitions, as occurs mostly often in fractured vertebral bodies, to be better delineated by our proposed method, named Balanced Growth (BGrowth). The experimental results on a database with 102 crushed and 89 healthy vertebrae showed that our approach significantly outperformed well-known methods from the literature such as GrowCut, LazySnapping, image foresting transform, otsu thresholding, among others. We have achieved 96.1% of accuracy while keeping fast run-time performance. Moreover, BGrowth presented the best results even with a rough/sloppy manual annotation. The work (RAMOS *et al.*, 2019) reports BGrowth's contribution.

Contribution 2. Reducing manual interaction on the segmentation of volumetric MRI exams.

The semi-automatic segmentation of vertebral bodies in a volumetric scenario is a challenging task, due to the large number of slices in the exams. We have assessed several possible solutions to this challenge and came up with three chronological contributions. We presented each time faster segmentation algorithm and less manual interaction approaches.

First, we introduced the 3D Balanced Growth (3DBGrowth) approach (RAMOS *et al.*, 2019a)[5] along with the slope coefficient to reduce the total number of annotated slices. The experimental results showed that, on average, only 37% of the slices had to be annotated, consequently making the process faster (on average, 36s for each vertebral body). 3DBGrowth significantly outperformed the competitors and kept comparable running time. Moreover, 3DBGrowth presented the best results even with simple/sloppy seed points, which demanded less effort from specialist on the annotation process.

Then, we proposed the Clever Segmentation (CleverSeg) algorithm (RAMOS *et al.*, 2019b), along with the mean absolute error between slices aimed at minimizing the number of annotated slices in each exam. In this work, we also optimized the manual annotation process by estimating the inside annotation based on the outside annotation. The previous state-of-the-art methods require inside and outside annotations. The experimental results showed that, on average, only 50% of the slices required outside annotations. The time

[5] This contribution won the conference's best paper award.

spent on overall annotations was 50% faster by using only the outside annotation and quickly estimating the interior annotation with our approach. CleverSeg presented better or similar results than the state-of-the-art while managing a statistically significant lower processing time. Moreover, we made CleverSeg available as a 3D Slicer extension[6] and incorporated it into URule and UTrack mobile applications (CAZZOLATO *et al.*, 2020; CAZZOLATO *et al.*, 2021), in a straightforward adaptation for the segmentation of colored images of skin wounds taken by mobile devices.

Finally, we proposed the Fast Clever Segmentation (FastCleverSeg) technique (RAMOS *et al.*, 2021), along with the Estimation of Annotation on Intermediary Slices (EANIS). EANIS requires the annotation of only two or three slices to produce an accurate volumetric segmentation. The experimental results showed that, combining FastCleverSeg and EANIS makes the segmentation process faster, regarding processing time and manual annotation, making it easier the production of reference manual segmentations. Moreover, our proposed segmentation technique is the fastest one and, at the same time, presented results similar to or better the state-of-the-art regarding segmentation methods. In average, EANIS performed in 6 ± 3 ms per ROI, while the manual annotation process took only 8.1 ± 2.2s. Moreover, FastCleverSeg presented an average f-measure of 94% with the fastest run-time: 25 ± 30 ms per ROI.

Contribution 3. Identifying patients with bone fragility using volumetric MRI exams and Radiomics texture features

We have assessed this novel research topic using data from a study group comprised of 47 patients submitted to DEXA and spine MRI (T2 weighted images). We segmented the lumbar vertebrae (L1 to L5) using our segmentation method (CleverSeg) in 3DSlicer. We divided the dataset into 97 volumetric vertebral bodies diagnosed as osteopenia/osteoporosis without fractures, and 97 volumetric vertebral bodies diagnosed as osteopenia/osteoporosis with fractures. From each vertebral body, we extracted 1,674 texture features using PyRadiomics[7], including a few variations using LoG and Wavelets. The experimental results, using traditional machine learning techniques, showed that texture analysis from spine MRI achieved high diagnostic performance for differentiating patients with and without vertebral body fragility fracture. The best results were obtained with feature selection, achieving up to 97% AUC and 92% accuracy. The works from Ramos *et al.* (2021), Ramos *et al.* (2021) report the results regarding this contribution.

[6] <https://github.com/JonathanRamos/SlicerCleverSegmentation>
[7] <https://pyradiomics.readthedocs.io/>

1.4 Outline

This chapter summarized our motivation, addressed the problems, objectives and main contributions of this doctorate research. We organized remaining of this monograph in eight chapters as follows.

Chapter 2: presents, concisely, fundamentals and relevant background essential to follow the work, considering a computational perspective as well as medical imaging perspective.

Chapter 3: details the results achieved by BGrowth, which tackles **Research Problem 1**.

Chapter 4: further explores the results achieved by 3DBGrowth and the negative slope coefficient, approached as *a first step* in **Research Problem 2**.

Chapter 5: presents the results achieved by CleverSeg along with the mean absolute error, approached as *a second step* in **Research Problem 2**.

Chapter 6: reports results achieved by FastCleverSeg and the estimation of annotations on intermediary slices (EANIS), approached as *a final step* in **Research Problem 2**.

Chapter 7: presents results achieved by BEAUT, which tackles **Research Problem 3**.

Chapter 8: concludes this thesis by stating our main contributions, publications yielded during the work as well as opportunities for future works.

Appendix A: details how we automatically estimated the inside annotation in a VCF study case.

Annex A: presents the source codes for our most relevant contributions.

CHAPTER

2

FUNDAMENTALS, BACKGROUND AND RELATED WORKS

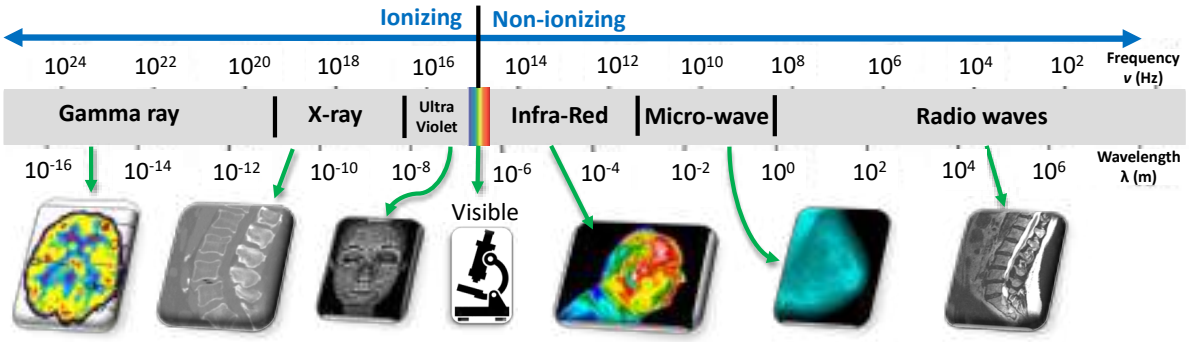
In this Chapter, first, we present the fundamentals of the representation of digital/medical imaging (Section 2.1). Next, we detail the research topics related to digital imaging, preprocessing, segmentation, feature extraction, classification, and measures to evaluate segmentation and classification results (Section 2.2). After that, we present, briefly, the anatomical structure of the human spine (Section 2.4). Then, we go further in the VCF and how a physician normally diagnoses this disease. We also present works on computational approaches for VCFs analysis, which includes segmentation and classification.

2.1 Fundamentals of medical and digital imaging

Over the centuries, the use of the electromagnetic spectrum, as depicted in Figure 2, has revolutionized the practice of medicine, aiding in the analysis, diagnoses and prognoses of several diseases. Note that the MRI takes place at the radio waves, not involving the use of ionizing radiation. Ionizing radiation can be harmful to patients if not used in a negligible dose.

In general, each range of the spectrum is represented as discretized values in a matrix, so that we can visualize, analyze, and store as digital images. Generally, a digital image can be represented as a 2-Dimensional (2D) matrix with "r" rows "c" columns. Each pair of coordinates or **pixel**, say entry (x,y) of the matrix, has a value that defines its intensity. Formally, a digital image is canonically represented as a bi-dimensional function, $f(x,y)$, with dimension $r \times c$, as

Figure 2 – Electromagnetic spectrum and common medical imaging modalities.



Source: Elaborated by the author.

follows (GONZALEZ; WOODS, 2011):

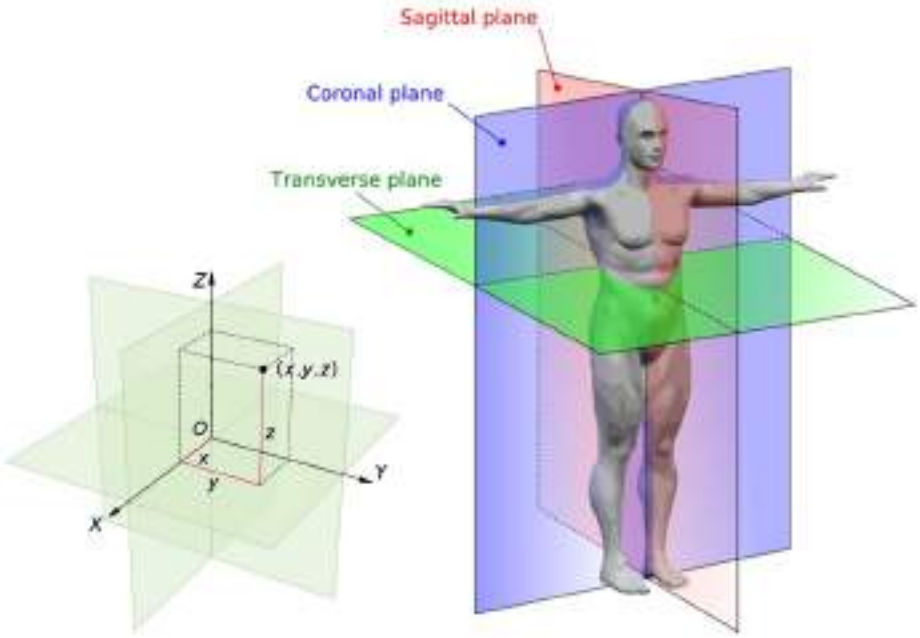
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,c-1) \\ f(1,0) & f(1,1) & \dots & f(1,c-1) \\ \vdots & \vdots & & \vdots \\ f(r-1,0) & f(r-1,2) & \dots & f(r-1,c-1) \end{bmatrix}. \tag{2.1}$$

Gray-scale images, in general, fit into this definition by presenting intensities varying between 0 and 255 (8-bits discretization in $[0,\ 255] \in \mathbb{Z}$) or 0 and 65,535 (16-bits discretization in $[0,\ 65{,}535] \in \mathbb{Z}$). A volumetric or 3-Dimensional (3D) image can also be represented with this definition by adding a third dimension ($f(x,y,z)$) so that each entry or **voxel** in the matrix, say (x,y,z), presents an intensity and the volumetric exam can be visualized in three distinct planes, as illustrated in Figure 3.

The sagittal or median plane (in red), is in the y and z Cartesian coordinates (perpendicular to the ground) which separates the mannequin's left from right. The coronal or frontal plane (in blue), is in the x and z Cartesian coordinates (perpendicular to the ground) which separates the mannequin's back (anterior) from the front (posterior). The transverse, axial, or horizontal plane (in green) is the x, y, and z Cartesian coordinates (parallel to the ground) which separates the mannequin's head (superior) from the feet (inferior) (MY-MS.ORG, 2008).

As a final remark, we highlight that, with the addition of channels, colored images can be represented by Equation 2.1 as well. However, in this Doctoral research we focus only 2D and 3D gray-scale images.

Figure 3 – Example of Cartesian coordinate system and the basics anatomical planes.



Source: My-MS.org (2008).

2.2 Digital imaging: preprocessing, segmentation, feature extraction and classification

Once an image is digitally stored as a matrix, a few techniques can be applied, isolated or in a pipeline, in order to achieve a specific goal. The techniques can be divided into four extensive research topics, preprocessing (Section 2.2.1), segmentation (Section 2.2.2), feature extraction (Section 2.2.3) and classification/analysis (Section 2.2.4) as well as measures employed to evaluate classification and segmentation results (Section 2.2.5). We briefly detail each topic in the following sections.

2.2.1 Preprocessing

The main goal of preprocessing is to suit the image for a specific application, so that to obtain visually informative images as well as to ease the subsequently processing and evaluation. In general, preprocessing can be performed in the **spacial domain** or in the **frequency domain**, detailed as follows (GONZALEZ, 2009).

Spatial domain: usually, employs convolutions (filters) on the image plane, such as high-pass and low-pass filters. Low-pass smooths the gray intensities and may smooths contours of the objects in the image as well, and, in general, blur and/or reduce noises. On the other hand, high-pass sharpen boundaries details, and, often, highlights details in blurred image.

Another simple, yet useful, preprocessing technique in this domain is the re-scaling of the gray intensities. For example, occasionally, 16-bits images are converted to 8-bits in order to reduce computational effort or to normalize the images in a given dataset. This conversion is accomplished as

$$g = \frac{f}{256}, \tag{2.2}$$

in which g represents the gray-scale with reduced range of intensities (GONZALEZ, 2009).

Frequency domain: analyze the rate of change among intensities using the Fourier transform. High frequencies represents portions of the image with several abrupt changes. For example, the portion of an image showing hair often contains high frequency data. Low frequencies represents the portions of the image with not much sudden changes, such as a flat wall. Thus, high and low-pass filters can be applied in this domain as well. High-pass filters can remove homogeneous details, while low-pass filters may remove heterogeneous details (GONZALEZ, 2009).

2.2.2 Segmentation

Generally, the segmentation is a crucial step and the most challenging one. Depending on the kind of analysis, a bad segmentation can directly impact the quality of the results in further steps. In most cases, the reference manual segmentation (performed by a specialist) is considered the Ground-Truth (GT) or gold-standard (GILLIES; KINAHAN; HRICAK, 2016). There are specific challenges on the segmentation of each region of the human body. For this reason, for each segmentation task, there is a better working approach. However, they all share some common goals (KUMAR *et al.*, 2012; GILLIES; KINAHAN; HRICAK, 2016):

i Present precise and replicable results;

ii Be as automatic as possible;

iii Require as minimum as possible of human interaction;

iv Present an acceptable processing time.

According to Gonzalez (2009), segmentation methods can be divided in 3 traditional groups:

Thresholding techniques that divide the image into groups of pixel intensities. In general, for this approach, the image is segmented in two groups, in which pixels with intensity values higher than the threshold is considered as foreground and the remaining pixels are considered as background. A well-known threshold techniques is the Otsu (OTSU, 1979) method, which automatically finds the best threshold value by using statistical measures.

The **Edge-detection** approach generally works by finding abrupt (local) changes in intensity. In general, the edges are found using a line detector, which frequently employs a mask or filter, such as Sobel, Prewitt, or Robert's, among others.

Region-growing methods work by growing seeds in the image using a determined criteria among seeds and non-seeds pixels. The seeds are points in the image that are set manually or automatically. The Superpixels (STUTZ; HERMANS; LEIBE, 2018) algorithm is the most known on the automatic seed category. In the manual seed category, there are several methods, in which, to the best of our knowledge, the GrowCut (VEZHNEVETS; KONOUCHINE, 2005) method has been the most used in the segmentation of vertebral bodies.

The three segmentation groups can be combined to obtain better results in a given application (GONZALEZ, 2009). In the last years, Deep Learning (DL) approaches, which employ neural networks, have presented promising results in the segmentation of images. However, DL techniques required hundreds or even thousands of segmented samples to learn how to correctly segment a given ROI. Depending on the kind of image or research topic, for example, medical images, acquiring that number of samples can be difficult due to the complexity of the images and demanded time to produce reference manual segmentation. Data augmentation methods and transfer learning could be applied to work with small databases. However, does not deal with the issues we tackle in this thesis. Also, when applying DL techniques, the interactive analysis guided by the understanding of the evaluation process is impaired. Moreover, depending on the kind of analysis, they may not generalize well enough to model the data. For these reasons, traditional methods, are still required to ease the production of reference manual segmentation.

With a given segmentation region in binary (background and foreground), a few post-processing can be performed using mathematical morphology, which we briefly detail next.

2.2.2.1 Mathematical morphology

In the mathematical morphology, each pixel (x,y) in the image is considered as an element in a set. Given a binary image, set A contains all the pixels from the foreground (ones) and set B contains all pixels from the background (zeros). A structuring element can present shape, such as a line or a square, so that the element can dilate (Definition 1) or erode (Definition 2) an object in the binary image. The dilation is an operation that "grows" the objects in the image, while the erosion "shrinks" it.

Definition 1 (Dilation)**.** The dilation between the binary object A in the image and an structuring element B can be performed as

$$A \oplus B = \{z|(\hat{B})_z \cap A \neq \emptyset\}, \tag{2.3}$$

which is the set of all elements whose center from A reflect and translated by B has at least one point in intersection.

Definition 2 (Erosion)**.** The erosion between the binary object in the image (A) and an structuring element B can be calculated as

$$A \ominus B = \{z|(B)_z \cap A^C \neq \emptyset\} \ , \tag{2.4}$$

in which $A \ominus B$ is the set of elements in A whose origin translated by B does not present any intersection.

Based on these two operations, erosion and dilation, the closing operation can be defined as the dilation followed by an erosion as

$$A \cdot B = (A \oplus B) \ominus B \ . \tag{2.5}$$

To better exemplify this, Figure 4 illustrates the closing operation. Note that, the closing operation removes protrusions from the binary object and, at the same time, keeps the original size of the object of interest.

2.2.3 Feature extraction

A digital image is considered a complex object/data. The feature extraction from complex data is usually performed by Feature Extraction Methods (FEMs) (Definition 3). FEMs yield numeric signatures of the image content, also known as feature vectors (Definition 4). Consequently, a digital image can be summarized/represented by its feature vector.

Definition 3 (Feature Extraction Method – FEM)**.** A **Feature Extraction Method (FEM)** is a non-bijective function ($\xi : \mathbb{S} \rightarrow \mathbb{V}$) that enables the representation of a complex object s_a from the complex data domain $\mathbb{S}$ into the feature vector domain $\mathbb{V}$ as v_a.

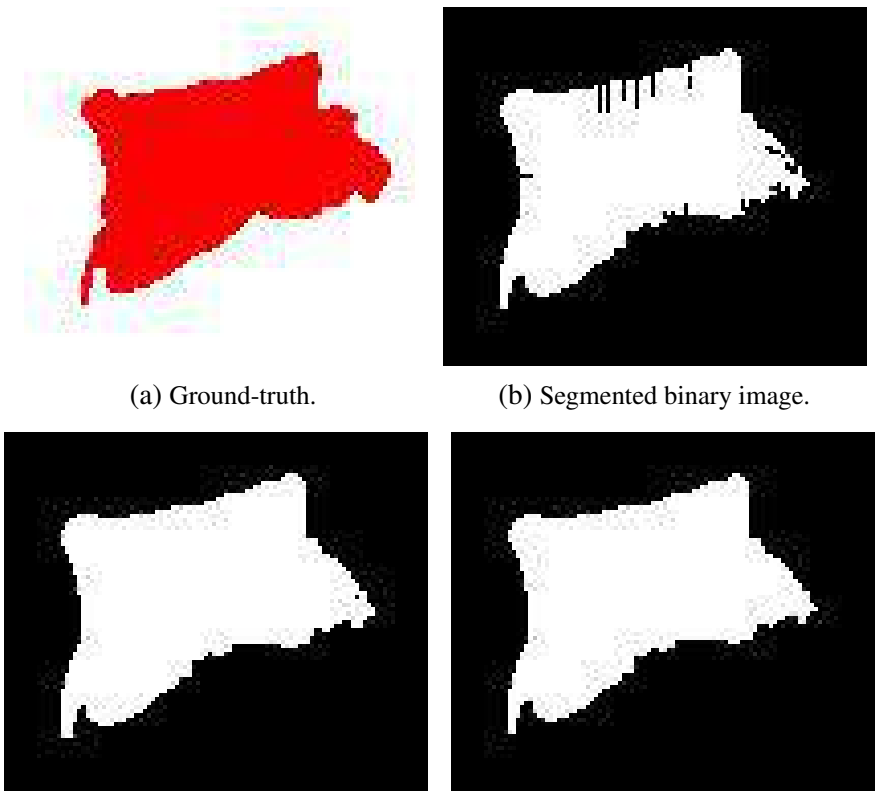
Definition 4 (Feature Vector)**.** A **feature vector**, say $v_a \in \mathbb{V}$, is generated by a FEM ($\xi(s_a) = v_a$) such that v_a summarizes/represents a complex object s_a from complex data domain $\mathbb{S}$ in the feature vector domain $\mathbb{V}$.

In general, FEMs work globally or locally considering features of color, texture or shape:

Global features consider every pixel of the image and may enable observers to rapidly perceive the gist of a scene. Features associated with specific types of scenes include degree of naturalness, degree of openness, degree of roughness, degree of expansion, and color.

Local features take into account only a part or a region of interest (ROI) of the image, which refer to a pattern or distinct structure found in an image, such as a point, edge, or small image patch. They are usually associated with an image patch that differs from its immediate

Figure 4 – Mathematical morphology applied over a binary image. We used a square structuring element of size 2×2 ([1 1; 1 1]).



(a) Ground-truth. (b) Segmented binary image.

(c) Dilation of (a). (d) Erosion of (b).

Source: Elaborated by the author.

surroundings by texture, color, or intensity. This category includes color, texture and shape features, such as the ones defined in the MPEG-7 standard by ISO/IEC JTC1 (MARTINEZ; KOENEN; PEREIRA, 2002). Local features .

In practice, several FEMs can be combined to improve the content representation being analyzed. Then, after the extractions of features vectors, the classification (Section 2.2.4) or comparison among other images can be employed. To compare the images, dissimilarity techniques ranks the images feature vectors using a distance function, which we cover next.

2.2.3.1 *Comparing images*

To compare pairs of images represented by their features vectors, a metric or Distance Function (DF) is usually employed. A metric is part of the definition of a metric space (Definition 5), which measures how far or close a pair of objects are. The closer the distance is to 0, the similar the images are. Reversely, the farther away from 0, the most dissimilar the images are (SAMET, 2006).

Definition 5 (Metric Space)**.** A **metric space** is an ordered pair $< \mathbb{V}, \delta >$, in which $\mathbb{V}$ is the domain of the feature vectors, and δ is a metric such that $\delta : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}^{+}$.

Definition 6 (Metric)**.** A **metric** δ calculates the distance between pairs of elements, for example $< v_a, v_b >$, such that $v_a, v_b \in \mathbb{V}$, following the properties of symmetry, non-negativity, triangular inequality, and the identity of indiscernibles. Therefore, $\forall v_a, v_b, v_c \in \mathbb{V}$, δ must hold the following properties to be considered metric:

Symmetry: $\delta(v_a, v_b) = \delta(v_b, v_a)$, the distance is a symmetric function;

Non-negativity: $\delta(v_a, v_b) \geq 0$, the distance value must be a positive number;

Triangular inequality: $\delta(v_a, v_c) \leq \delta(v_a, v_b) + \delta(v_b, v_c)$, the distance between two objects must not be higher than the distance between them passing through a third object;

Identity of indiscernible: $\delta(v_a, v_a) = 0$, the distance of an object to itself is 0.

There are several families of distance functions that follow the metric restrictions. However, the Minkowski family, or L_p norm, is the most used:

$$\xi(v_a, v_b) = \left(\sum_{i=1}^{n} (v_a^i - v_b^i)^p \right)^{1/p}, \tag{2.6}$$

in which n is total number of features from feature vectors v_a and v_b (both with the same dimension n). Note that when p have values 1, 2 and ∞, they are equivalent to Manhattan or city block, Euclidean and Chebyshev distances, respectively (CONDE; DOMÍNGUEZ, 2018).

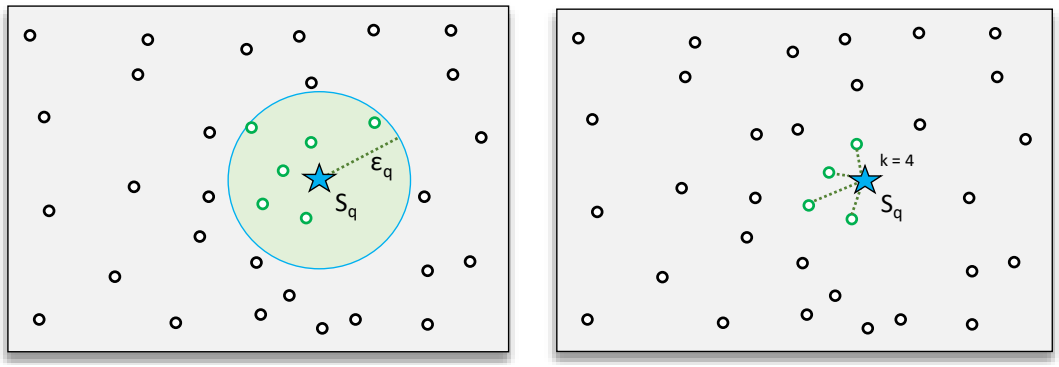
Based on the aforementioned concepts, the **similarity search or query** can be performed as a range query (Figure 5a) or as k-Nearest Neighbor (k-NN) query (Figure 5b). A range query return images (signatures) with distance equal or less than a threshold ε_q. A k-nn query returns the k nearest images in ascending order.

2.2.4 Classification

To classify an image, given its feature vector, in general, machine learning techniques are employed. To do so, a predictive classifier gets labeled features and learns the best way to classify them. The classification methods may use one of the following strategies (FACELI *et al.*, 2011):

Distance: one of the most popular in this category is the k-NN classification, which observes the distance among training instances (feature vectors). However, this type of method does not learn a compact model for the data and the computational cost for distance calculation between feature vectors grows as the number of instances expands.

Figure 5 – Graphical examples of similarity searches commonly used for querying images based on their content. S_q represents the query image, ε_q represents a range threshold for a range query, and k is the number of nearest neighbors for a k-nn query.



(a) Range query. (b) k-nn query.

Source: Elaborated by the author.

Probability: the frequency in which a pattern occurs in the feature vectors is highly relevant in this case. For example, the Naïve Bayes classifier estimates the probability of a feature based on how frequent it appears in the training data. The main disadvantage of this technique is that it assumes there is no dependence among the features, ignoring any correlation among them. On the bright side, the technique only requires a few instances in the training phase.

Search: takes advantage of the "divide-and-conquer" technique, in which a problem is divided in sub-problems, composing a decision path. Therefore one or more paths lead to the classification of each class. Methods in this category are widely kown as Decision Tree techniques.

Optimization: aims at minimizing or maximizing an objective function. For example, the Artificial Neural Networks (ANN) that employs an error correction constraint. At each stage of the learning phase, the network recurs to the error optimization between the expected classes and the predicted classes.

A few techniques integrate and take advantage of the aforementioned strategies:

Radial Basis Function (RBF): networks uses distances and an objective function approximation, which can be understood as an elementary ANN (BROOMHEAD; LOWE, 1988).

Quadratic Discriminant Analysis (QDA): separates two or more classes in a quadratic surface using statistical measures. The goal is to minimize an objective function (quadratic), assuming normally distributed data (JAMES *et al.*, 2014).

Support Vector Machines (SVM): calculates the hyper-plane that best separates the classification groups in the training data. To this end, the method uses a sparse kernel decision machine that avoids computing posterior probabilities in the phase of building its learning model (AWAD; KHANNA, 2015).

Boosting techniques: use a set of weak learners, for example, a decision tree, to compose a strong learner. For example, AdaBoost method, which gets the outputs of weak learner and combines the predictions into a weighted sum that represents the final output. The AdaBoost method can be divided in two versions, M1 and M2, in which M stands for "Multiclass". For binary classification problems (our case), M1 and M2 are equivalent (FREUND; SCHAPIRE, 1996). Another example is the LogitBoost method, which takes AdaBoost as a generalized additive model and applies logistic regression as a cost function. Therefore, LogitBoost can be seen as a convex optimization approach aiming at minimizing the logistic loss (FRIEDMAN; HASTIE; TIBSHIRANI, 2000).

2.2.5 Segmentation and classification evaluation measures

For the sake of simplicity, let us consider a binary segmentation and classification scenario. In the segmentation, regions are segmented in foreground (positive, object of interest, for example, a mass, a bone or a tissue) and background (negative, anything but the object of interest). In the classification, the goal is to differentiate between the positive class (for example, a bone fracture) and the negative class (for example, absence of bone fracture). In this context, generally, the comparison measures employ the following cases:

True Positive (TP): counts the number of correctly identified positive cases:

- Segmentation: foreground entries[1] correctly segmented as part of the foreground;
- Classification: positive cases correctly predicted as positive.

True Negative (TN): holds the number of correctly identified negative cases:

- Segmentation: background entries correctly segmented as part of the background;
- Classification: negative cases correctly predicted as negative ones.

False Positive (FP): comprises the total of negative cases identified as positive ones:

- Segmentation: background entries miss-segmented as part of the foreground;
- Classification: negative cases miss-predicted as positive ones.

False Negative (FN): represents the total number of positive cases identified as negative ones:

- Segmentation: foreground entries miss-segmented as part of the background;

[1] Pixels or voxels.

- Classification: positive cases miss-predicted as negative ones.

In order to compare the resulting segmentation or classification, a few measures, such as Accuracy, Precision, Sensitivity, Jaccard, F-Measure, and Area Under the Receiver Operator Characteristic Curve (AUC), can be calculated considering the aforementioned cases. Those measures present percentage values ranging from 0 to 1, $[0,1]$, defined as follows (TAHA; HANBURY, 2015; SOKOLOVA; LAPALME, 2009).

Accuracy (ACC): takes into account all aforementioned cases to compute the ratio of correct positive (TP) and negative predictions (TN) to the total number of input samples:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} . \tag{2.7}$$

The Accuracy provides an overall effectiveness of the segmenter or classifier. However, sometimes, it is interesting to know how the positive and negative classes are being handled separately. Often, the positive class are more relevant to the analysis. Therefore, to that end, the Precision and Sensitivity present insights on the positive class.

Precision (PRE): Considers the FPs to measure the percentage of TPs correctly identified:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} . \tag{2.8}$$

The higher the number of FPs, the lower the Precision, which indicates:

- Segmentation: how much is over-segmented outside of the ground-truth.
- Classification: class agreement of the data labels with the positive labels.

Sensitivity (SEN): uses the FNs to measure the percentage of TPs correctly identified:

$$\text{Sensitivity} = \text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} . \tag{2.9}$$

The higher the number of FNs, the lower the Sensitivity, indicating:

- Segmentation: how much is not segmented within the ground-truth.
- Classification: effectiveness of a classifier to identify positive labels.

The Sensitivity is also known as Recall or True Positive Rate (TPR).

Jaccard Coefficient (JAC): also known as the intersection over union, calculates the intersection of the reference manual segmentation and segmentation yielded by a segmentation algorithm (TP) divided by their union (TP + FN + FP).

$$\text{Jaccard} = \frac{|TP|}{|TP + FN + FP|} . \tag{2.10}$$

F-Measure (FM): measures the balance between Precision and Sensitivity by calculating their harmonic mean:

$$\text{F-Measure } = 2 \times \frac{\text{Precision } \times \text{Sensitivity}}{\text{Precision } + \text{Sensitivity}} \ . \tag{2.11}$$

The F-Measure is also known as the Sørensen–Dice coefficient or Dice Score.

AUC: Measures the area underneath the entire Receiver Operating Characteristic (ROC) curve plot. The ROC is a graph showing the performance of a classification model at all classification thresholds. The graph is composed of the Sensitivity values against the False Positive Rate (FPR):

$$\text{FPR} = \frac{FP}{FP+TN} \ . \tag{2.12}$$

Lowering the classification threshold leads to predicting more positive cases, increasing both FP and TP. Reversely, higher classification threshold values lead to predicting more negative cases, increasing both FN and TN. AUC can be understood as the probability that the model ranks a random positive case more highly than a random negative case.

For the case of a binary segmentation, the AUC can be calculated using the False Positive Rate (FPR) against the False Negative Rate (FNR):

$$\text{AUC} = 1 - \frac{\text{FPR} + \text{FNR}}{2} \ , \ \text{FNR} = \frac{FN}{FN+TP} \ . \tag{2.13}$$

When it comes to analyzing specifically 3D segmentation results, a widely employed measure is the **Average Hausdorff 's Distance (AVD)**, which indicates how far away in voxels, on average, the Ground-Truth (GT) and SegmenTation (ST) are:

$$\text{AVD}(\text{GT},\text{ST}) = \max\{ \ \xi(\text{GT},\text{ST}), \ \xi(\text{ST},\text{GT}) \ \} \ , \tag{2.14}$$

in which $\xi(\text{GT},\text{ST})$ is the directed Average Hausdorff:

$$\xi(\text{GT},\text{ST}) = \frac{\sum \min ||gt_i - st_i||}{N} \ , \forall gt_i \in \text{GT} \ , \ \forall st_i \in \text{ST} \ , \ i = 1,2,...,N \ . \tag{2.15}$$

An AVD of 0 indicates comparable segmentations(MAROŠEVIĆ, 2018). The directed Hausdorff Distance is generally sensitive to noise and outliers. In the medical context, noise and outliers are commom in the exams. Therefore, Average Hausdorff is more indicated to compare segmentation results, since it is more stable and less sensitive to outliers (TAHA; HANBURY, 2015).

2.3 Medical imaging: systems that aid decision-making

One of the most used application to manage and retrieve medical exams (digital images and metadata) is the Picture Archiving and Communication Systems (PACS), which, in general, allows a fast and efficient access to the exams. Usually, a PACS adopt the Digital Imaging and Communication in Medicine (DICOM) as the standard file format (LAROBINA; MURINO, 2014). DICOM incorporates protocols for the techniques of radiography, CT, MRI, ultrasound, and, radioactive therapy. Moreover, DICOM defines standardization of information sharing, such as via portable medias, image compression, 3D visualization, presentation, and reports (KAHN *et al.*, 2007).

A few systems use the aforementioned architecture and concepts to aid in the decision-making process. In the next section, we explore two of them: CAD and CBIR systems.

2.3.1 CAD and CBIR systems

CAD systems **output a second-opinion** to the physician. The aim is to aid in the decision-making by providing an output based on automatic quantitative analysis of the exam (AHMAD *et al.*, 2019). In general, CAD systems can be divided into four steps, as shown in Figure 6 and detailed as follows.

1. First, the image may be preprocessed (Section 2.2.1) for several reasons, such as to remove noises introduced in the acquisition phase or to enhance the segmentation phase.

2. Next, the region of interest (ROI) is isolated/segmented (Section 2.2.2) so that unnecessary regions are not included in the analysis.

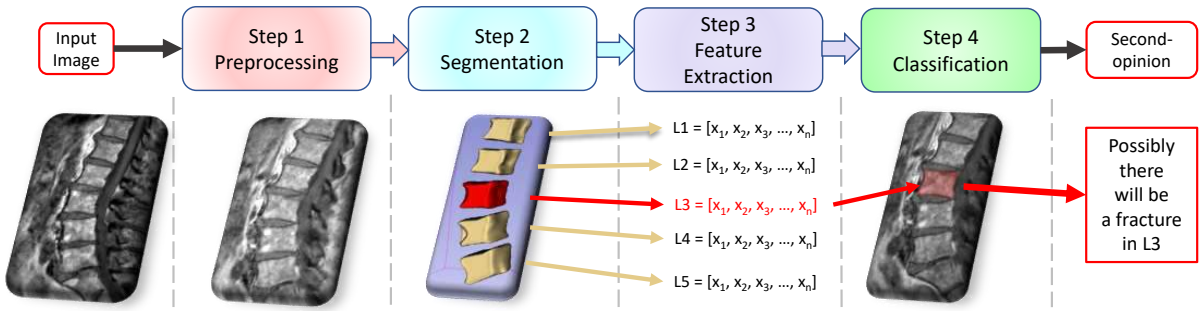
3. After that, quantitative features are extracted from the ROI (Section 2.2.3). The features can be characteristics of texture, shape or color.

4. Finally, the extracted features are used to perform the classification (Section 2.2.4), which yields a second-opinion to the physician.

CBIR systems **retrieve images** based on their content. The goal is to minimize the *semantic-gap*, which is the gap between high-level characteristics, for example objects as we interpret (car, house, cat, and so on) and low-level characteristics such as shape, texture, and color (VIKHAR; KARDE, 2017).

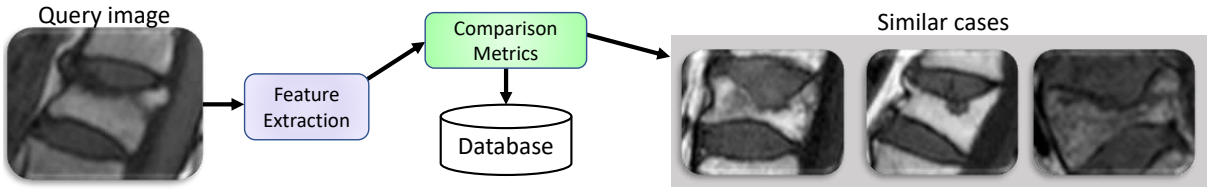
In a medical context, a CBIR system takes an input image as query, for example a fractured vertebral body, as exemplified in Figure 7, and returns similar cases based on the following steps.

Figure 6 – Fundamental steps in a CAD system.



Source: Elaborated by the author.

Figure 7 – Illustration of the fundamentals steps in a CBIR system.



Source: Elaborated by the author.

1. First, the system extracts features from the query image (Section 2.2.3). In this case, the image may or may not be segmented, depending on the kind of features that will be extracted from the image. In general, shape features require segmented ROIs.

2. Next, the feature vector of the query image is compared to the feature vector of the images in the database. The comparison is made using a similarity query, as explained in Section 2.2.3.1.

3. Finally, the result is the rank (k-NN query) or grouping (range query) of the similar images in the database.

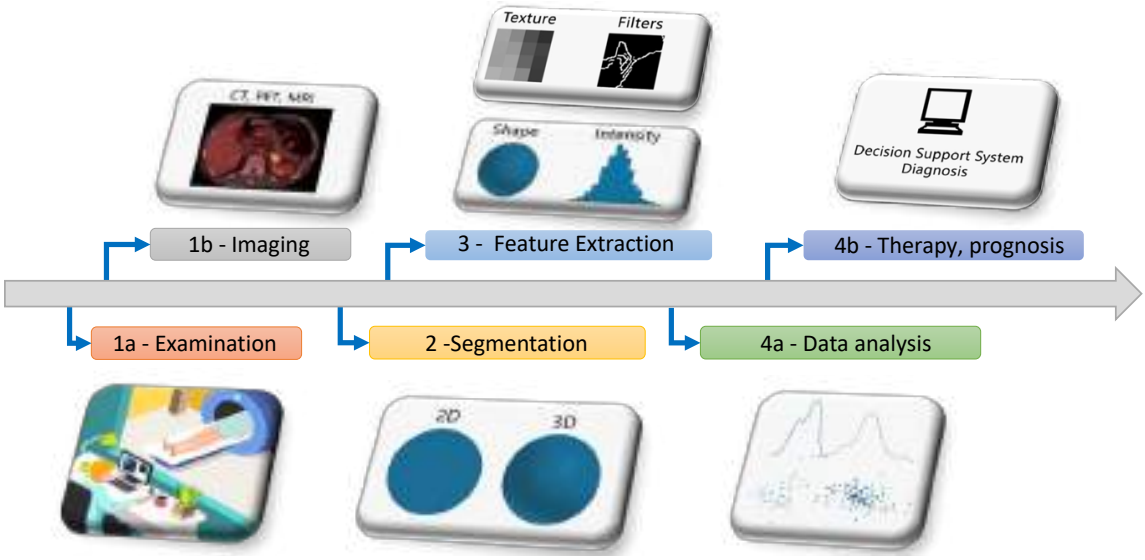
Although CAD and CBIR systems have presented promising results, medical imaging has progressively shifted from conventional visual image analysis to quantitative personalized medicine. Furthermore, the field of medical image analysis has grown exponentially, which implicates in increased data set sizes, consequently growing the number of pattern recognition tools to analyze such data. For this reason, the development of processes for high-throughput extraction of quantitative features were facilitated. Thus, allowing the conversion of images into mineable data and the subsequent analysis of these data for decision support (DURON *et al.*, 2019; KUMAR *et al.*, 2012; AHMAD *et al.*, 2019; XU *et al.*, 2016). To take advantage of this high-throughput extraction of quantitative features, the Radiomics approach was designed to improve quality and precision of models that aid decision-making, as we explain next.

2.3.2 Radiomics approach

Radiomics is a quantitative technique to aid medical imaging, which combines the extraction of hundreds or thousands of features from the images, as well as data from the patients not directly acquired from the images (DURON *et al.*, 2019; KUMAR *et al.*, 2012; AHMAD *et al.*, 2019; XU *et al.*, 2016). The aim of radiomics is to support decision making by enhancing the existing data, in order to detect information of disease-specific processes that are not accessible through traditional visual inspection of the generated images. Thus, improving quality and precision of models that aid in the decision making. The main idea is not to automate the process, but rather supports existing ones with additional information (TIMMEREN *et al.*, 2020).

Radiomics can be divided into four main parts, as depicted in Figure 8. Each part presents its own intrinsic challenges, which harden the implementation in a clinical set-up (GILLIES; KINAHAN; HRICAK, 2016), detailed as follows.

Figure 8 – The Radiomics approach workflow.



Source: Adapted from Lambin *et al.* (2012), Gillies, Kinahan and Hricak (2016).

1. **Examination and Imaging:** the exams can vary, coming from Computed Tomography (CT), MRI, Positron Emission Tomography (PET), among others. Moreover, the exams may present diverse acquisition parameters, such as resolution, patient position, 3-D reconstruction, slice thickness. For those reasons, it is difficult to compare results from distinct hospitals with different acquisition equipments. Furthermore, finding exams with the similar clinical parameters, for example disease stage, is a challenging task. Thus, standardization is the greatest challenge in the acquisition phase (LAMBIN *et al.*, 2012; GILLIES; KINAHAN; HRICAK, 2016).

2. **Segmentation:** poses as a crucial, challenging and also controversial step. The proceeding analysis relies on the accuracy of the segmentation, making it crucial. In some cases, the

boundaries of the ROI are not always well-defined in the image, making it challenging to segment. It is controversial because there is no consensus about manual or automatic (reproducible) segmentation, or which one between both is more reliable (GILLIES; KINAHAN; HRICAK, 2016).

3. **Feature extraction:** the goal is to extract as many features as possible, and, when possible, include information of the patient anamnesis, biopsy data, genomics and proteomics data. The total number of feature extracted are then selected, discarding irrelevant features, in which machine learning techniques are employed. The challenges in this step includes imprecise features due to issues aforementioned in earlier steps.

4. **Analysis:** one of the challenges in this step is the number of available samples for the training methods. In general, the data is filtered and a few of them are discharged due to missing or inconsistent data, which reduces the total number of sample size. The analysis in a preprocessed database (with removed irrelevant cases, missing or inconsistent data) employs data mining techniques (artificial intelligence, machine learning or statistics analysis).

According to Gillies, Kinahan and Hricak (2016), when building a database for Radiomics analysis:

``[...] numbers are king and quality is queen.''

This implicates that the accuracy of the predictive classifier model relies on having sufficient data. In general, the best models can accommodate additional clinical or genomic covariates, which increases the need for large high-quality data sets. Most of the times, the curation of a small dataset, say 100 patients, takes over hundreds of hours. Here, the curation refers to segmentation, validation through charts, review of pathology and image acquisition parameters, and so on (GILLIES; KINAHAN; HRICAK, 2016). For those reasons, in this Doctorate research we focus on easing the time spent in laborious manual tasks by proposing faster and accurate segmentation methods and, at the same, with fewer and less demanding interactive and/or manual annotations.

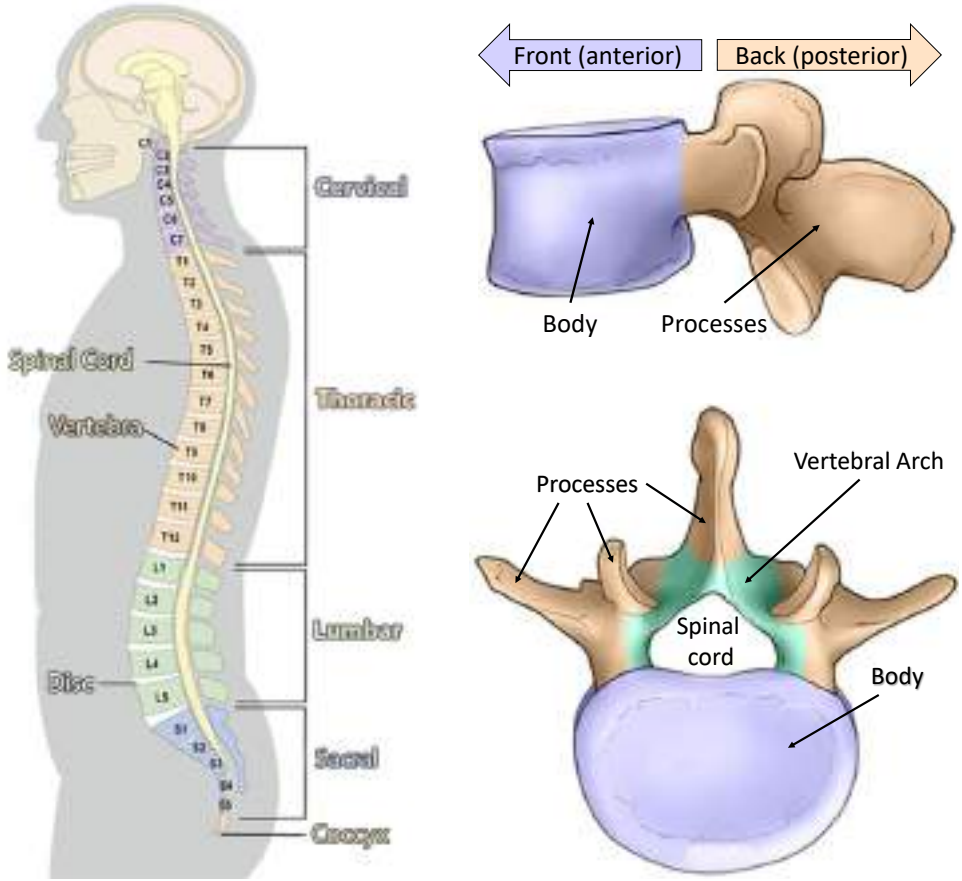
2.4 Brief spine anatomy

The human body has 206 bones, in which 33 individual bones, stacked one on top of the other, composes the vertebral column, also known as backbone or spine. One of the most important role of the spine is to protect the spinal cord from injury. The spinal cord, roughly, performs the transmission of neurological signs from the brain to the whole body, controlling the voluntary muscles of the trunk and upper and lower extremities and receives sensory input from these areas of the body. Other important roles of the spine includes the main support for

our body, allowing us to stand upright, bend, and twist (BICAN; MINAGAR; PRUITT, 2013; SPINE, 2018).

A healthy spine depends on strong muscles and bones, flexible tendons and ligaments, and sensitive nerves. We can feel pain if any of these structures gets affected by strain, injury, or disease. The spine is divided into five main regions, so that each region has unique features that help in performing its main purpose, as illustrated in Figure 9a and detailed as follows (SPINE, 2018). Starting from top to bottom:

Figure 9 – Spine groups and vertebra parts.



(a) The spine is grouped in five parts: cervical, thoracic, lumbar, sacral, and coccyx.

(b) The three main parts of a vertebra: body, vertebral arch, and processes for muscle attachment.

Source: Adapted from Spine (2018).

Cervical (numbered C1 to C7): holds the weight of the head, which, in average, weights 4.5 Kilograms. The five cervical vertebrae (C1 to C5) presents the wider range of movements due to two vertebrae connected to the skull (C1 and C2). The first vertebra (C1), presents a ring shape and directly connects to the skull, which allows us to give the “yes” sign with our head. The second vertebra (C2), presents a peg-shaped axis, which has a projection called the odontoid. The movement of C2 allows us to give the “no” sign with our head.

Thoracic (numbered T1 to T12): presents the main purpose to hold the rib cage and protect the heart and lungs. The thoracic spine has a limited range of motion.

Lumbar (numbered L1 to L5): mostly bear the weight of the body and absorb the stress of lifting and carrying heavy objects. The lumbar vertebrae are larger in size when compared to the other vertebrae. For this reason, most of the diseases/fractures caused by bone weakening takes place in the lumbar spine.

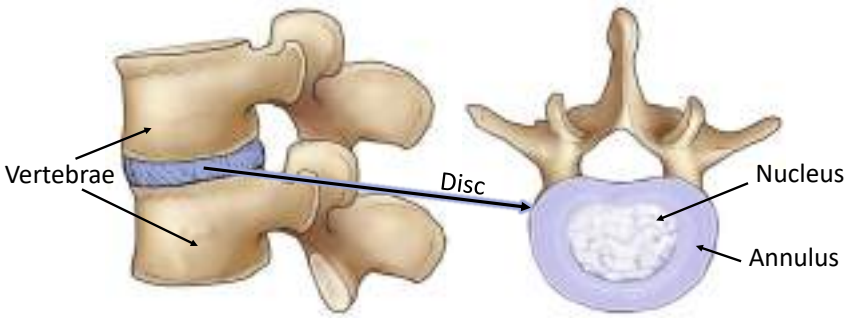
Sacral (numbered S1 to S5): connect the spine to the hip bones (iliac). The five vertebrae are fused together and, joined with the iliac bones, they form the pelvic girdle in the shape of a ring.

Coccyx: also known as tailbone, presents four fused bones provide attachment for the ligaments and muscles of the pelvic floor.

While each one of the five regions of the spine has unique regional features, every vertebra has three functional parts, as depicted in Figure 9b. The **body** is designed to bear weight and withstand compression and presents the shape of a drum. The **vertebral arch** bone that protects the spinal cord. The **processes** have the shape of a star and are designed for muscle attachment.

As Figure 10 shows, between each pair of vertebrae there is an intervertebral disc, which "link" or cushion the vertebrae and prevents the bones from rubbing each other. The discs are

Figure 10 – Discs are made of two main parts, in which the annullus pulls the vertebrae bones together against the resistance of the nucleus.



Source: Adapted from Spine (2018).

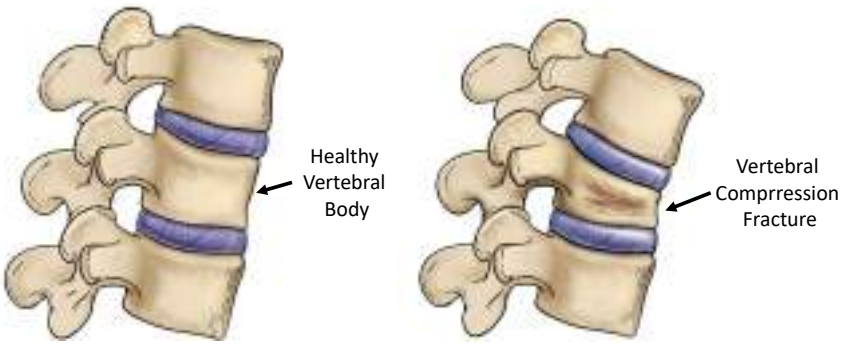
designed like a radial car tire and is composed of two main parts, the nucleus and the annulus. The annulus (outer ring), has crisscrossing fibrous bands, which attach between the bodies of each vertebra. The nucleus (inner ring) contains mostly fluid, which is absorbed during the night as we lie down and is pushed out during the day as we move upright. Our discs increasingly lose the ability to reabsorb fluid and become brittle and flatter as we age. This is one of the reasons why we get shorter as we grow older.

In this Chapter, we present the relevant background related to medical imaging. From the acquisition and representation (Section 2.3), in which the utilization of the electromagnetic spectrum of light has revolutionized the practice of medicine, to systems that support the decision-making, such as Computer-Aided Diagnosis (CAD) and Content-based Image Retrieval (CBIR) systems (Section 2.3.1), as well as the Radiomic approach (Section 2.3.2).

2.4.1 Vertebral Compression Fracture – VCF

Among several spine diseases, VCFs, exemplified in Figure 11, often occurs in older adults. In general, it is caused by bone weakening diseases, such as osteoporosis (benign VCF) or bone metastases (malignant VCF). Usually, VCFs affects the lumbar vertebrae since they bear the weight of the body. In a clinical context, VCFs caused by osteoporosis are diagnosed as benign while VCFs caused by bone metastases are diagnosed as malignant. In both cases, patients present similar complaints. Thus, the differentiation between them is crucial for the right diagnose and treatment (PEREIRA, 2016).

Figure 11 – Example of healthy vertebral body and vertebral compression fracture. (VCF)



Source: Adapted from Spine (2018).

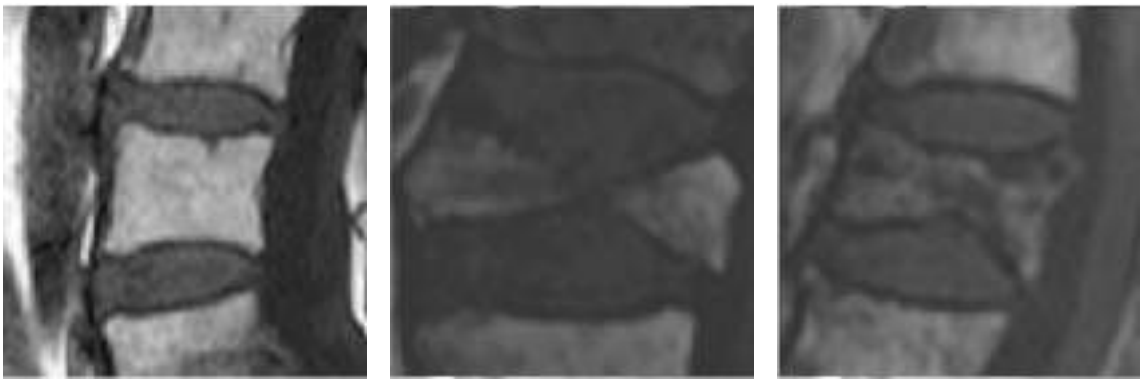
To investigate a VCF's etiology, the physician can use MRIs of the spine. The analysis of gray intensities within the vertebrae is performed to differentiate between a benign or malignant VCF. In the clinical routine, the physician compares the gray levels of the intervertebral disc with the gray levels as well as the shape of the vertebral body. Frequently, the following criteria are used to investigate the etiology of a VCF (considering MRI exams with T1-weighted sequences) (PEREIRA, 2016; FRIGHETTO-PEREIRA *et al.*, 2016b):

Healthy: the vertebral body frequently presents an approximately rectangle shape in the sagittal plane, which may include osteophyte that causes small deformations in the shape of the vertebral body itself, due to outgrowth and marginal bone proliferation. Usually, the inner part of the vertebral body presents higher gray levels than the intervertebral disc (CUÉNOD *et al.*, 1996), as shown in Figure 12a.

Benign: In this case, the posterior wall may present angulated or irregular contours with fragmented retropulsion on the the vertebral body. Accentuate concave shape can be observed in vertebral plateaus, as well as rough contours with indentations caused by subchondral bone impaction, as shown in Figure 12b

Malignant: may present a posterior bulge or convexity of the posterior vertebral body wall along with concave deformation of the vertebral plateaus. Malignant processes and the bulging neoplastic tissue may cause the contours of vertebrae to be relatively rounded or smoother, as shown in Figure 12c.

Figure 12 – Example of vertebral bodies with and without VCFs in MRI exams (T1-weighted sequence).



(a) Healthy. (b) Benign fracture. (c) Malignant fracture.

Source: Adapted from Casti *et al.* (2017).

Sometimes it is not possible to achieve a precise diagnose analyzing the characteristics aforementioned. In those cases, the investigation of the patient's history, biopsy exams and computational methods can be employed.

2.5 Final Considerations

In this chapter we presented the fundamental concepts of digital imaging related to the research developed in this Doctoral thesis. All concepts and related works not presented herein will be covered in the next chapters as demanded by each context. As stated in the beginning of the chapter, the literature covered in this chapter is quite broad regarding the subjects presented. We did not intent to deeply discuss each topic. Rather, we briefly and concisely presented the key-points of each subject, so that to cover the required background and knowledge to understand the contributions of this doctoral research. Since, most systems that support decision-making, such as CAD, CBIR and Radiomics, in general, requires laborious manual interaction in some steps, for example, in the segmentation. Therefore, in this Doctorate research we focused mostly on easing the manual annotation time in the segmentation process. In the next Chapters we cover

our contributions, in a chronological order, towards advancing the semi-automatic segmentation for musculoskeletal region, specifically for vertebral bodies and muscles segmentation.

CHAPTER

3

SEGMENTING VERTEBRAL COMPRESSION FRACTURES IN MAGNETIC RESONANCE IMAGING USING BALANCED GROWTH

In this chapter, we present the challenging task of segmenting VCFs, which, in general, present heterogeneous gray intensities within the same vertebral body, which difficult its segmentation. To cope with this segmentation challenge, we present our proposed Balanced Growth (BGrowth) segmentation method (Section 3.1). BGrowth takes advantage of the motto "unity is strength" to balance the weights along the growing path of the segmentation region (the vertebral body in this case). Thus, our method achieves a better delineated border at both dark and bright regions even with rough/sloppy seeds (manual annotation). The experimental results (Section 3.3) on a dataset with 102 crushed and 89 healthy vertebrae show that our approach significantly outperforms well-known and state-of-the-art methods from the literature. We have achieved an accuracy up to 95% while keeping acceptable processing time performance, that is equivalent to the state-of-the-art methods. Finally, we present the concluding considerations (Section 3.4).

3.1 Balanced Growth Method

Based on the motto "unity is strength", **B**alanced **Growth** (BGrowth) takes into account that all pixels in the path of an expanding region have influence on its growth. BGrowth achieves "unity" by balancing the weights along the growing path, so that the final segmentation is a "unanimity" among the pixels. As a consequence, small intensities transitions are better delineated, as occurs mostly often in fractured vertebral bodies.

For simplicity's sake, we consider the segmentation of two regions, foreground as vertebral body and background as the remaining parts. Since images come from MRIs exams,

we consider gray-scale images. However, the method can be easily adapted to more than two regions and for color images.

Let us consider a digital image as a matrix, as we defined in **??**, and represented it by I. Then, let us consider a matrix of annotations or labels regions and represented it by L, which is the matrix that contains the annotations/seeds manually drawn by the user. Both matrices I and L have dimension $r \times c$, in which r is the number of rows and c is the number of columns. Each entry in L has value -1, 0 or 1, representing background, unlabeled, and foreground, respectively.

Algorithm 1: BGrowth method.

Input: Image I and labels matrix L.
Output: Segmented binary image L == 1.

`/* Step 1: Set initial weights */`
1 **for** $L(i,j) \neq 0, \forall(i,j)$ **do**
2 $\quad W(i,j) \leftarrow 1.0$ `// Equation 3.1`
3 $maxIntensity \leftarrow \max_{\forall i,j} I(i,j)$
`/* Step 2: For each labeled pixel and its 8-Neighbors */`
4 **for** $L(i,j) \neq 0, \forall(i_n,j_n), \forall(i,j)$ **do**
5 $\quad intensityDiff \leftarrow 1 - (|I(i,j) - I(i_n,j_n)| \ / \ maxIntensity)$
6 $\quad strength \leftarrow W(i,j) \times intensityDiff$ `// Equation 3.3`
7 $\quad$ **if** $(strength - W(i_n,j_n)) > 0.01$ **then**
8 $\quad\quad L(i_n,j_n) \leftarrow L(i,j)$
9 $\quad\quad W(i_n,j_n) \leftarrow 0.5 \times W(i_n,j_n) + 0.5 \times strength$

BGrowth method is divided into two main steps, as shown in Algorithm 1. The **first step** is to set up a weight matrix W (Algorithm 1) as in Equation 3.1, in which $i = 1,2,...,M$ and $j = 1,2,...,N$ are the matrices indexes and W has the same dimension as I and L. We assign the weight 1.0 for labeled pixels and the weight 0.0 for unlabeled pixels $L(i,j) \neq 0$ (line 2). Then, we find the highest gray intensity in image I, represented as $maxIntensity$ (line 3).

$$W(i,j) = \begin{cases} 1.0 & \text{if } L(i,j) \neq 0. \\ 0.0 & \text{otherwise.} \end{cases} \tag{3.1}$$

The **second step** is to analyze each labeled pixel at index (i,j) (line 4) and its 8-neighbors (i_n,j_n) in the image (we visit all pixels in the image). For each neighboring pixel at index (i_n,j_n), first, the absolute pixel intensity difference (line 5) is normalized by the maximum pixel value in the image and subtracted by 1, which produces values within the range $[0,1]$:

$$intensityDiff = 1 - \frac{|I(i,j) - I(i_n,j_n)|}{\max_{\forall(i,j)} I(i,j)} \ . \tag{3.2}$$

Next, a strength factor $strength$ (line 6), which is the product of the central pixel weight and the intensity difference, is calculated as

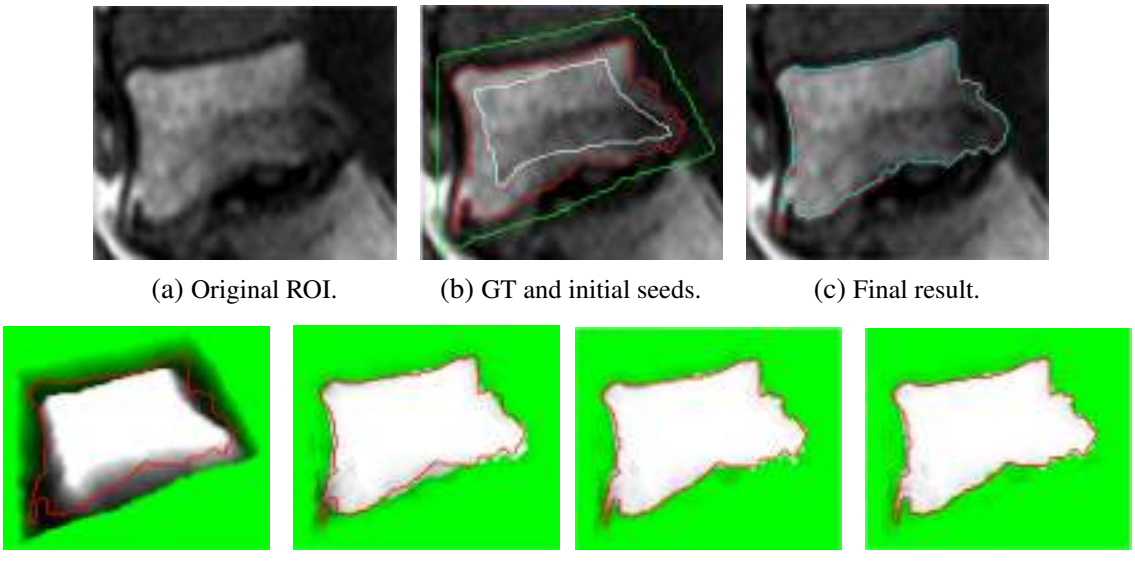
$$strength = W(i,j) \times intensityDiff\ , \tag{3.3}$$

If the difference between the $strength$ and the neighboring pixel weight (line 7) is greater than a threshold (empirically set to 0.01 in this case), then the central pixel (i,j) expands to the neighboring pixel at index (i_n, j_n): its label is changed (line 8) and the new weight is balanced with the old one (line 9). The algorithm iterates until no neighbor label was updated.

Without a threshold, the balancing may go back and forth, which may prevent the algorithm from converging or to run unnecessary iterations. Thus, we avoid averaging values over the second decimal place, consequently, managing the number of iterations and speeding up the processing time.

The balance induced by the weights produces smoother borders at regions with small pixel intensities. In our experimental evaluation, we determined that balancing the weights (old and new one) equally produces the best results. Moreover, as long as the interior annotation contains small portions of the dark and bright regions, it can work well even with rough interior and exterior seed points. To better illustrate this, Figure 13 shows a few iterations of BGrowth for a single vertebral body segmentation. Note that, at iteration 8 (Figure 13e), a few dark regions (outside of the vertebral body) are still part of the foreground and, as the balancing goes on, the foreground shrinks towards the ground-truth. The final result is quite close to the manual segmentation.

Figure 13 – Example of BGrowth's iterations. Ground-Truth in red, interior annotation in white, exterior annotation in green and final segmentation in cyan.



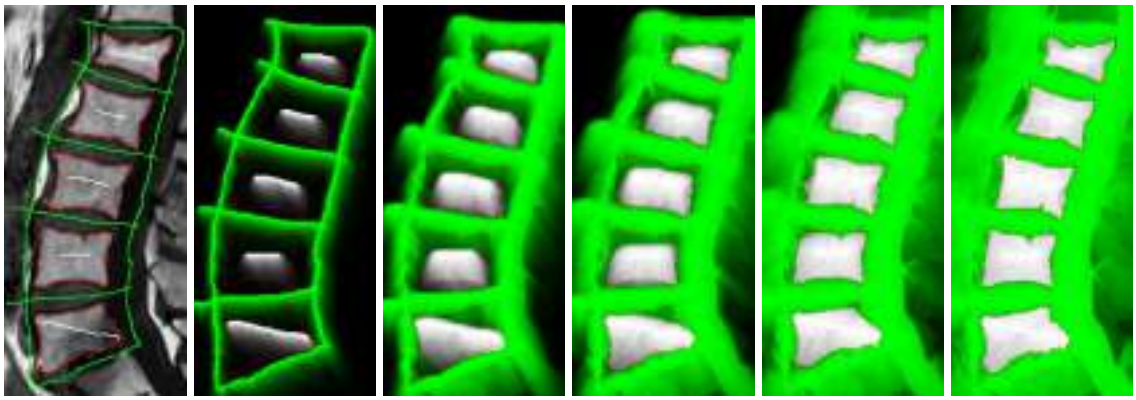
(a) Original ROI. (b) GT and initial seeds. (c) Final result.

(d) Iteration 1. (e) Iteration 8. (f) Iteration 15. (g) Iteration 30.

Source: Adapted from Ramos *et al.* (2019).

Figure 14 shows another example of BGrowth's iterations on five lumbar vertebrae. Note that, the method works well even when a simple line is given as seed points. The only constraint is that the interior annotation has to comprise the dark/bright regions within the vertebral body.

Figure 14 – Example of BGrowth's iterations on five lumbar vertebral bodies: Ground-Truth, interior annotation and exterior annotation boundaries are outlined in red, white and green, respectively.



(a) Annotations. (b) Iteration 1. (c) Iteration 5. (d) Iteration 10. (e) Iteration 20. (f) Iteration 50.

Source: Adapted from Ramos *et al.* (2019).

As we work with the balance of old and new strength factors, it might happen that the process of expansion goes back and forth on the same pixel. Therefore, we would recommend the definition of a maximum number of iterations depending on the kind of image being segmented. For the vertebral body segmentation, we have empirically used a maximum of 30 iterations for the segmentation of a single vertebral body. For a whole exam, as exemplified in Figure 14, a few more iterations might be necessary.

3.1.1 Time complexity

In the worst case scenario, the algorithm has a time complexity of $r \times c \times 8 \times it = O(r \times c \times it)$, where it is the maximum number of iterations.

3.2 Materials and Methods

In this section, we first describe the experimental database in Section 3.2.1, which is composed of healthy, malignant and benign vertebral bodies. Then, we present the experiment design, starting with the parameters settings in Section 3.2.2, which presents the computational set-up as well as how the manual annotations were performed. Next, we detail the state-of-the-art considered in the analysis in Section 3.2.3. Finally, in Section 3.2.4, we present the considered measures to validate the experimental results.

3.2.1 Patient database

We have used a meaningful database of lumbar vertebral bodies (L1, L2, L3, L4, and L5), including 89 healthy vertebral bodies and 102 with VCFs. Bone metastases, confirmed by biopsy and histopathological analysis, occurred in 49 VCFs (malignant VCFs). The remaining 53 VCFs were benign fractures and, following clinical guidelines, not all of them were confirmed by biopsy or histopathological analysis. On these cases, a musculoskeletal radiologist with over 20 years of experience carefully reviewed the radiological information system (RIS) and the hospital information system (HIS) and supervised the manual segmentation (ground-truth). This study was approved by the Ethics Research Committee of the Ribeirão Preto Medical School - USP, where the database was acquired.

The exams were acquired using the Achieva 1.5T MRI system (Philips Medical Systems, Best, the Netherlands). T1-weighted MRI sequences were produced using the following imaging parameters: mean echo time (ET) = 9.83 ± 0.82 ms, mean repetition time (RT) = 489±75.58 ms, mean field of view (FOV) = $32.25 \times 33.33 \pm 3.98 \times 6.39$ cm, acquisition type 2D, flip angle 90∘, and a slice thickness of 4 mm. The images used for the analysis have a spatial resolution of 353 μm/pixel. To compose the database, the central sagittal slices of 63 patients were considered, which included 38 women and 25 men, with a mean age of 62.25 ± 14.13 years.

To assure the same conditions to every segmentation method, all exams were normalized from 65,535 gray levels (16 bits/pixel) into 256 gray levels (8 bits/pixel) using Equation 2.2.

We divided the database into four groups as follows.

Healthy: contains only normal vertebral bodies, without any fracture, summing up to 89 healthy vertebral bodies.

Benign: has only osteoporosis fractures (53 benign VCFs).

Malignant: presents only metastasis fractures (49 malignant VCFs).

All: comprises the three earlier groups, which sums up to 191 vertebral bodies.

3.2.2 Parameters settings

The parameters for all methods that used superpixels were empirically set to $m \times n \times 0.25$ superpixels (25% of the total pixels in the image) for each ROI, in which m and n are the number of rows and columns of the ROI, respectively. The maximum number of iterations was set to 300 for Chan-Vese technique. For BGrowth and GrowCut, no maximum number of iterations was set. The remainder of the parameters for all methods were set to default settings to avoid loss of generalizability. No pre or post-processing technique were applied to assure the same conditions for all segmentation methods.

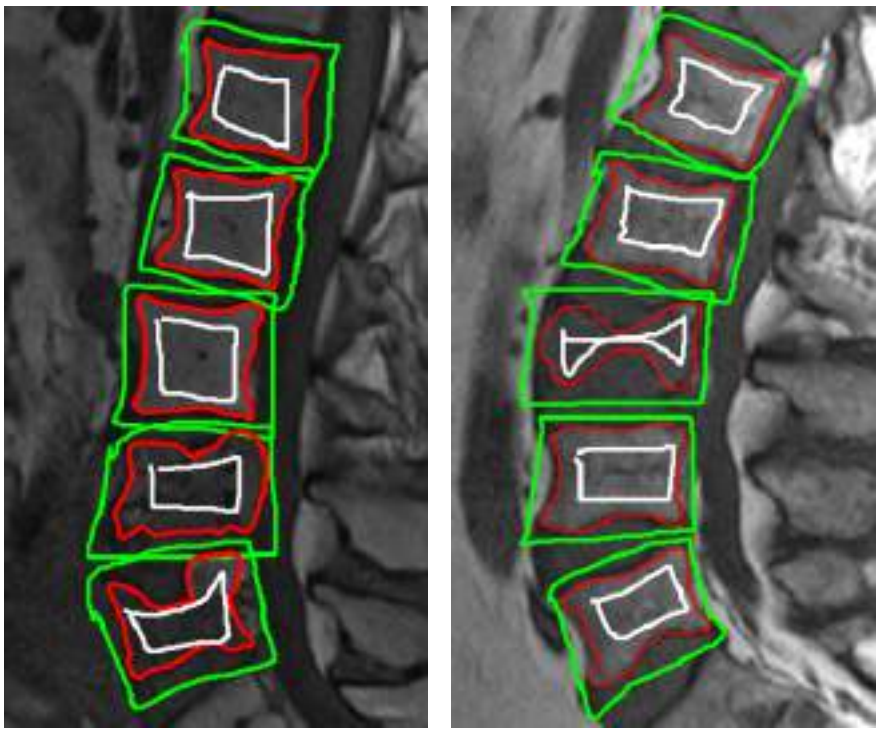
3.2.2.1 Computational set-up

The experiments were performed on a desktop with a 3.60GHz Intel(R) Core(TM) i7 CPU and 16GB RAM using Matlab(R) version 2021a.

3.2.2.2 Manual annotations

As Figure 15 shows, the initial interior and exterior seeds annotation were performed in a "sloppy" way, i.e., no detailed boundary for accentuated curves were drawn. In general, the annotation looks like a rectangle.

Figure 15 – Example of sloppy annotations (seed points): ground-truth (GT) in red; interior and exterior annotations in white and green, respectively.



(a) Exam 42. (b) Exam 50.

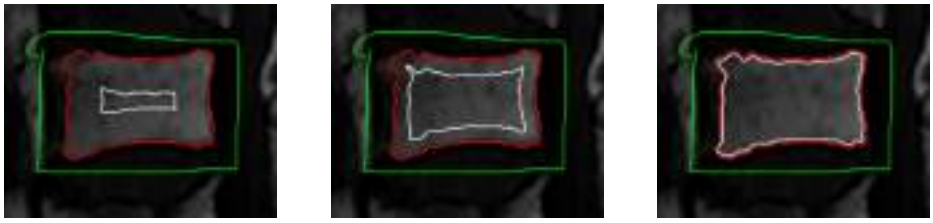
Source: Adapted from Ramos *et al.* (2019).

3.2.2.3 Variations in the inside/outside annotations

We performed interior and exterior variations on the annotations, which we detail as follows.

Interior variation: We keep a sloppy exterior annotation and vary the percentage of interior annotation on each vertebral body, as exemplified in Figure 16. A percentage of the ground-truth is extracted for each vertebral body. We always keep at least a line that goes through the dark and bright regions. The percentage starts at 10% (Figure 16a), increasing by 10%, up to 90% (Figure 16c), summing up to 9 variations.

Figure 16 – Examples of variations in the interior annotation.

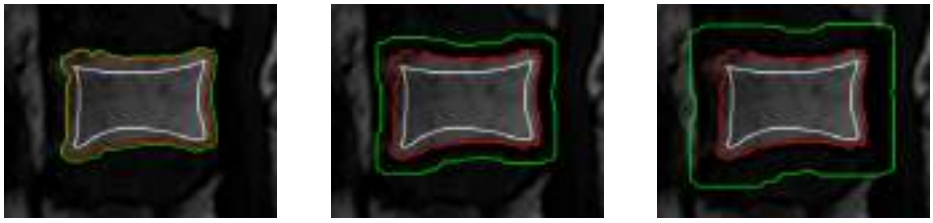


(a) 10%. (b) 50%. (c) 90%.

Source: Elaborated by the author.

Exterior variation: We keep a sloppy interior annotation and vary the distance, in pixels, of the exterior annotation in relation to the ground-truth, as exemplified in Figure 17. Starting at a distance of 3 pixels (Figure 17a) from the ground-truth, increasing by 3 pixels up to 27 pixels (Figure 17c), which sums up to 9 variations. We crop the ROI according to the convex hull of the exterior annotation. Therefore, in the exterior annotation variation, the ROIs size vary according to the size of the exterior annotation.

Figure 17 – Examples of variations in the exterior annotation.



(a) 3 pixels. (b) 15 pixels. (c) 27 pixels.

Source: Elaborated by the author.

3.2.3 State-of-the-art segmentation methods

In order to evaluate the correctness of BGrowth, we compared it with the methods GrowCut, GrabCut, LazySnapping, VBSeg (BARBIERI *et al.*, 2015), Image Foresting Transform (IFT)[1] (FALCÃO; STOLFI; LOTUFO, 2004), Chan-Vese or Snakes (CHAN; VESE, 2001) and Otsu threshold (OTSU, 1979). To the best of our knowledge, although most of the aforementioned methods have not been used for the segmentation of VCFs, they have been adapted or used in several medical applications (YONG *et al.*, 2017; LU *et al.*, 2017; WU; NAKAO; MATSUDA, 2016; JUNIOR *et al.*, 2018). We consider those segmentation approaches in our analysis to address a wider assessment of VCFs segmentation over the state-of-the-art methods, considering they are the most representative approaches for this kind of analysis/segmentation. We briefly describe each method as follows.

[1] <https://github.com/joakimlindblad/FuzzConn/blob/master/ift.m>

GrowCut: Egger, Nimsky and Chen (2017) segmented normal vertebral bodies using the GrowCut (VEZHNEVETS; KONOUCHINE, 2005) algorithm, since the faster version, named as Fast GrowCut (ZHU *et al.*, 2014), presents slightly lower segmentation performance than the original GrowCut (ZHU *et al.*, 2014). Both approaches employ several seeds points inside and outside of the ROI/VOI.

GrabCut: uses Gibbs energy (GEMAN; GEMAN, 1984) and Gaussian Mixture Models as soft segmentation for the background and foreground (ROTHER; KOLMOGOROV; BLAKE, 2004).

LazySnapping: LazySnapping works as an interactive image cutout tool and also uses Gibbs energy, combining graph cut with a pre-computed over-segmentation, e.g. Superpixels (LI *et al.*, 2004).

IFT: is a graph-based approach to the design of image processing operators based on connectivity, dealing with how different operators relate to each other (FALCÃO; STOLFI; LOTUFO, 2004).

Chan-Vese: uses level-set functions to overcome common issues with Snakes segmentation methods (CHAN; VESE, 2001), such as sensibility to initial seeds (the seed need to be placed closer to the boundaries of the segmentation object) and difficulties dealing with protrusions and topological changes (JIANG; ZHANG; NIE, 2012).

Otsu: binarizes a gray-scale image by automatically finding the best threshold that maximizes the intra-class variance (OTSU, 1979).

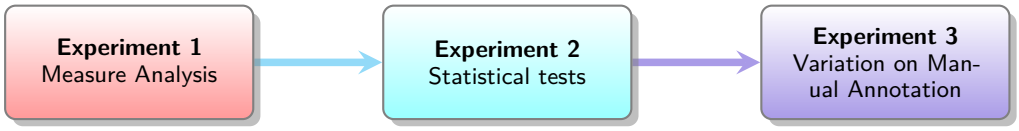
3.2.4 Evaluation measures

To evaluate BGrowth's segmentation results, we employed the measures of Accuracy (Equation 2.7), Precision (Equation 2.8), Sensitivity (Equation 2.9), Jaccard (Equation 2.10), and F-Measure (Equation 2.11).

3.3 Experiments, results and discussion

We divided our experimental design in three main parts, as depicted in Figure 18 and detailed as follows. First, in Section 3.3.1, we present the experimental results considering a measure analysis. Then, in Section 3.3.2, we analyzed resulting statistical difference for each segmentation method against our proposed BGrowth. Finally, in Section 3.3.3, we assess the impact of changing the percentage of manual annotation inside and outside each vertebral body for every approach that uses this kind of annotation.

Figure 18 – Experimental design to validate BGrowth.



Source: Elaborated by the author.

3.3.1 Measures analysis

We conducted the measure analysis in two main parts, first, we assess the results for group All, then for groups Healthy, Benign, and Malignant.

Table 1 reports the average results for group All, taking into account each segmentation method considered in our experiments. Overall, BGrowth presented the best results, the only exception was in the Run-Time, in which Otsu was the fastest and BGrowth the second faster. However, although Otsu was the fastest, it presented one of the lowest Precision and Sensitivity. GrowCut presented the second best results for most measures.

Table 1 – Results of the segmentation assessment considering group All.

	Measure (%)					Run-Time
Method	Accuracy	Precision	Sensitivity	F-Measure	Jaccard	(ms/ROI)
BGrowth	**95.8 ± 07**	**96.0 ± 05**	**96.8 ± 06**	**96.4 ± 06**	**90.6 ± 11**	12.8 ± 07
GrowCut	95.5 ± 07	95.7 ± 05	96.6 ± 07	96.1 ± 06	89.9 ± 11	18.1 ± 11
IFT	94.4 ± 07	95.1 ± 05	95.4 ± 07	95.2 ± 06	87.6 ± 11	22.9 ± 08
LazySnapping	93.7 ± 07	94.6 ± 05	94.8 ± 07	94.6 ± 06	86.1 ± 12	48.6 ± 40
GrabCut	66.8 ± 16	92.4 ± 14	53.6 ± 28	62.1 ± 21	55.9 ± 19	26.9 ± 58
Chan-Vese	79.6 ± 16	89.4 ± 10	76.6 ± 23	79.6 ± 20	65.0 ± 21	500.5 ± 70
Otsu	80.5 ± 13	85.4 ± 13	83.6 ± 13	83.5 ± 11	63.3 ± 23	**0.6 ± 04**
VBSeg	71.0 ± 11	80.0 ± 14	87.0 ± 06	–	–	–

Source: Research data.

Note – **Bold** and underlined values highlight the best and second-best column results, respectively.

Table 2 reports the average results for groups Healthy, Benign, and Malignant. BGrowth presented the highest results for every measure on each group. GrowCut presented the second-best results for every measure on every group. IFT and LazySnapping presented promising results as well. Note that, the segmentation methods drop the segmentation performance when there are VCFs. Moreover, each method tends to present a default segmentation behavior or pattern, as illustrated in Figure 19. The dark and bright regions within the same vertebral bodies difficult the segmentation for all methods. Thus, most methods present spiked borders and over-segmentation on those regions due to the neighbors structure intensities (very similar , dark). On these cases, even the manual segmentation is challenging and requires much experience of the specialist. On the other hand, sometimes the segmentation methods, such as LazySnapping (Figure 19f)

Table 2 – Results of the segmentation assessment for groups Healthy, Benign, and Malignant.

		Measure (%)				
	Method	Accuracy	Precision	Sensitivity	F-Measure	Jaccard
Healthy	BGrowth	**96.1 ± 07**	**95.5 ± 06**	**97.3 ± 07**	**96.4 ± 07**	**92.4 ± 11**
	GrowCut	95.7 ± 08	95.1 ± 06	97.2 ± 08	96.1 ± 07	91.8 ± 11
	IFT	95.1 ± 07	94.9 ± 06	96.2 ± 08	95.5 ± 07	90.6 ± 11
	LazySnapping	94.3 ± 07	94.1 ± 06	95.5 ± 08	94.7 ± 07	88.7 ± 11
	GrabCut	70.6 ± 15	94.3 ± 13	53.7 ± 27	63.5 ± 21	63.7 ± 15
	Chan-Vese	87.8 ± 08	92.6 ± 06	84.9 ± 10	88.3 ± 08	77.8 ± 12
	Otsu	87.9 ± 06	92.3 ± 05	85.4 ± 09	88.4 ± 06	77.7 ± 10
	VBSeg	–	–	–	–	–
Benign	BGrowth	**95.2 ± 08**	**96.1 ± 06**	**96.7 ± 07**	**96.4 ± 06**	**87.9 ± 15**
	GrowCut	94.9 ± 09	95.9 ± 06	96.4 ± 07	96.1 ± 06	87.1 ± 15
	IFT	93.9 ± 09	95.0 ± 06	95.8 ± 07	95.3 ± 06	84.5 ± 15
	LazySnapping	92.8 ± 09	94.8 ± 06	94.4 ± 07	94.5 ± 06	81.9 ± 15
	GrabCut	62.7 ± 18	92.9 ± 12	50.7 ± 28	60.0 ± 23	46.2 ± 20
	Chan-Vese	79.7 ± 16	86.8 ± 11	82.5 ± 18	83.4 ± 15	58.4 ± 22
	Otsu	82.3 ± 11	86.6 ± 10	87.5 ± 09	86.7 ± 08	58.1 ± 23
	VBSeg	75.0± –	85.0± –	86.0± –	–	–
Malignant	BGrowth	**95.7 ± 01**	**96.9 ± 02**	**95.9 ± 02**	**96.4 ± 01**	**89.7 ± 04**
	GrowCut	95.5 ± 01	96.6 ± 02	95.8 ± 02	96.2 ± 01	89.1 ± 04
	IFT	93.5 ± 02	95.5 ± 03	93.5 ± 04	94.4 ± 02	84.9 ± 05
	LazySnapping	93.7 ± 02	95.7 ± 03	93.7 ± 04	94.6 ± 02	85.4 ± 06
	GrabCut	63.4 ± 15	88.0 ± 18	56.6 ± 30	61.4 ± 19	50.4 ± 19
	Chan-Vese	62.5 ± 13	85.5 ± 13	53.0 ± 31	57.4 ± 24	45.3 ± 15
	Otsu	63.1 ± 11	69.5 ± 12	75.4 ± 20	70.0 ± 11	35.2 ± 19
	VBSeg	62.0± –	68.0± –	88.0± –	–	–

Source: Research data.

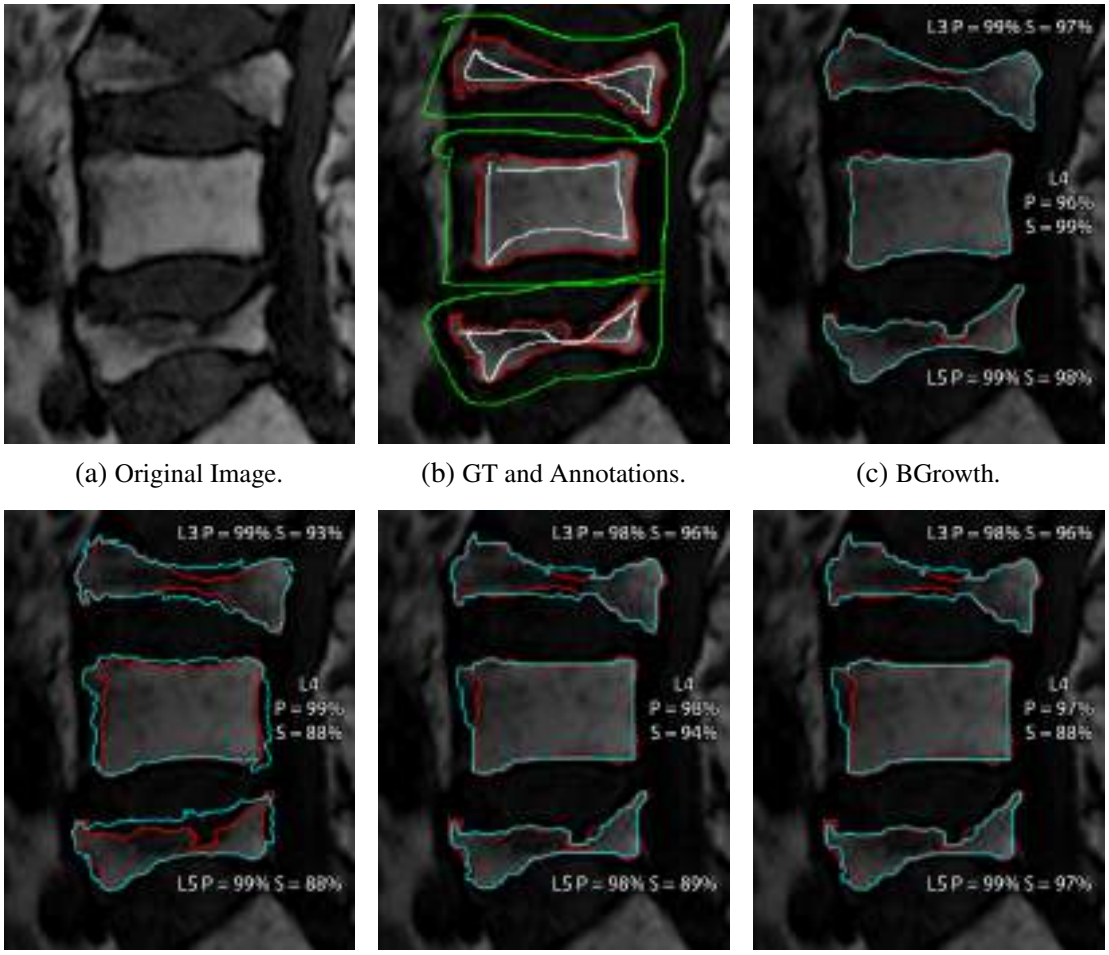
Note – **Bold** and underlined values highlight the best and second-best column results for each group, respectively.

and IFT (Figure 19e), tend to produce straight line borders, as exemplified by the results for L4. Although most competitors presented promising results, sometimes they fail to delimit borders even for healthy vertebral bodies (L4). In general, BGrowth (Figure 19c) generated smother and more delineated borders (better balance between Precision and Sensitivity, which are closer to the ground-truth.

3.3.2 Statistical Analysis

To test the normality of data, we employed the Kolmogorov-Smirnov (MASSEY, 1951) test, since our data presents several identical values. The null hypothesis that the data follow a

Figure 19 – Segmentation results for two benign VCFs (L3 and L5) and one healthy vertebral body (L4). Ground-truth in red and semi-automatic segmentation in cyan. P represents the Precision and S represents Sensitivity.



(a) Original Image. (b) GT and Annotations. (c) BGrowth.

(d) GrowCut. (e) IFT. (f) LazySnapping.

Source: Elaborated by the author.

normal distribution was rejected for every case. Therefore, we used the Wilcoxon (WILCOXON; KATTI; WILCOX, 1970) test to analyze if there were significant differences. We considered the null hypothesis as: "the median from data *a* is not greater than the median from data *b*". We considered the alternative hypothesis as: "the median from data *a* is greater than the median from data *b*". Here, data *a* can be, for example, the Precision from BGrowth and data *b* can be, for example, the Precision from GrowCut.

Table 3 reports the results for the Wilcoxon test considering every group. Note that, BGrowth presented significantly better results, most of the times, than any other method on each measure. Although BGrowth did no presented greater Precision or Accuracy than GrowCut and GrabCut in some cases, BGrowth always presented better F-Measure. This indicates that, in general, BGrowth presents a better balance between Precision and Sensitivity, and, at the same time, presents a greater F-Measure result. In some cases, BGrowth did not present better

Accuracy or Jaccard than the other methods. For this reason, we conducted another statistical test, in which we asses the opposite case. We test, in Table 4, if GrowCut, and GrabCut presented better results than BGrowth in the cases in which BGrowth did not presented greater results in Table 3.

Table 3 – Hypothesis testing for every group: p-value results at the 5% significance level. The null hypothesis is that BGrowth did not present greater median than a given competitor. ✔ indicates rejection of the null hypothesis (BGrowth presented better results), and ✗ indicates failure to reject the null hypothesis (BGrowth did not present greater results).

	BGrowth ×	Accuracy	Precision	Sensitivity	F-Measure	Jaccard
All	GrowCut	0.01 (✔)	0.07 (✗)	0.06 (✗)	0.01 (✔)	0.01 (✔)
	IFT	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	LazySnapping	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	GrabCut	0.00 (✔)	1.00 (✗)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Chan-Vese	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Otsu	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
Healthy	GrowCut	0.01 (✔)	0.08 (✗)	0.07 (✗)	0.00 (✔)	0.00 (✔)
	IFT	0.00 (✔)	0.01 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	LazySnapping	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	GrabCut	0.00 (✔)	1.00 (✗)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Chan-Vese	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Otsu	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
Benign	GrowCut	0.11 (✗)	0.36 (✗)	0.11 (✗)	0.00 (✔)	0.16 (✗)
	IFT	0.00 (✔)	0.00 (✔)	0.02 (✔)	0.00 (✔)	0.00 (✔)
	LazySnapping	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	GrabCut	0.00 (✔)	0.99 (✗)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Chan-Vese	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Otsu	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
Malignant	GrowCut	0.10 (✗)	0.21 (✗)	0.38 (✗)	0.00 (✔)	0.13 (✗)
	IFT	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	LazySnapping	0.00 (✔)	0.02 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	GrabCut	0.00 (✔)	0.85 (✗)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Chan-Vese	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)
	Otsu	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)	0.00 (✔)

Source: Research data.

In Table 4, GrowCut did not present significantly better/greater results than BGrowth, which indicates that both methods presented similar results in those cases. On the other hand, GrabCut presented significantly better Accuracy than BGrowth in the majority of the groups. However, BGrowth presented significantly better Sensitivity and F-Measure than BGrowth (Table 3). Therefore, we can conclude that, overall, BGrowth presented greater or similar results than the competitors.

Table 4 – Hypothesis testing for every group: p-value results at the 5% significance level. The null hypothesis is that BGrowth presented greater median than a given competitor. ✔ indicates rejection of the null hypothesis (BGrowth did not present greater results), and ✗ indicates failure to reject the null hypothesis (BGrowth presented better results) .

	BGrowth ×	Accuracy	Precision	Sensitivity	F-Measure	Jaccard
All	GrowCut	–	0.93 (✗)	0.94 (✗)	–	–
	GrabCut	–	0.00 (✔)	–	–	–
Healthy	GrowCut	–	0.92 (✗)	0.93 (✗)	–	–
	GrabCut	–	0.00 (✔)	–	–	–
Benign	GrowCut	0.89 (✗)	0.64 (✗)	0.89 (✗)	–	0.84 (✗)
	GrabCut	–	0.01 (✔)	–	–	–
Malignant	GrowCut	0.90 (✗)	0.79 (✗)	0.63 (✗)	–	0.87 (✗)
	GrabCut	–	0.15 (✗)	–	–	–

Source: Research data.

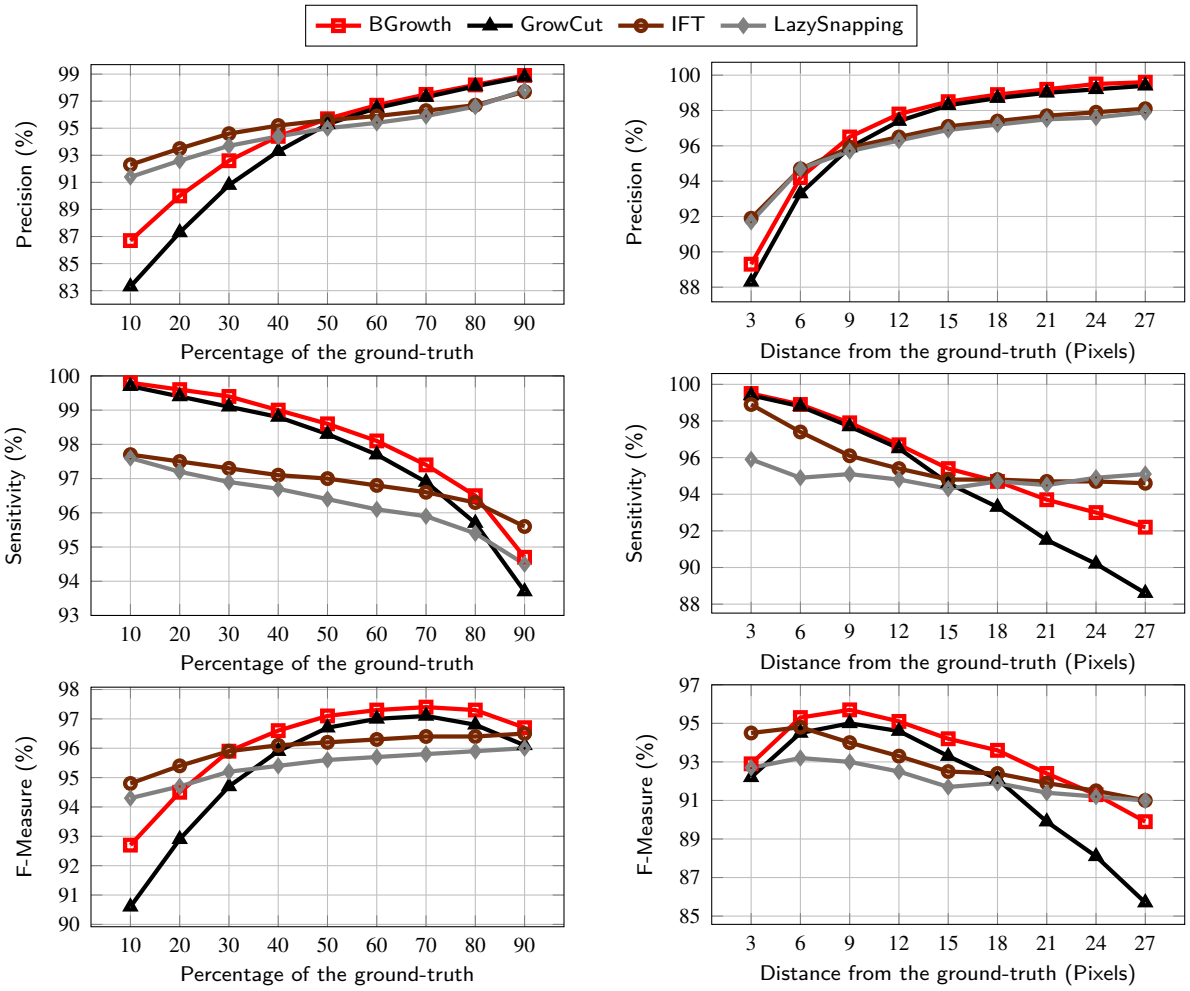
3.3.3 Annotation Variation

In Figure 20 we report the Precision, Sensitivity, and F-Measure for the inside/outside annotation variation, as we detailed in Section 3.2.2.3. We considered group All for the four segmentation methods that achieved the best results in previous experiments. We discuss the results in two parts as follows.

Interior annotation variation (Figure 20a): from 10% to 40%, IFT and LazySnapping presented the highest Precision and the lowest Sensitivity. This indicates that both methods do not grow much the inside region in this range of inside annotation/seeds. From 50% to 80%, BGrowth presented increasingly highest Precision (95% and up), while maintaining the highest Sensitivity, consequently, presenting the highest F-Measure as well. GrowCut presented results closer to BGrowth. At 90%, every method on each measure lower the results. However, 90% of inside annotation (Figure 17c) rarely happens in a real case scenario, as it is practically the ground-truth itself, which takes too much time to annotate. Overall, each method presented a trade-off between Precision and Sensitivity: the more inside annotation, the greater the Precision, while the lower the inside annotation, the greater the Sensitivity. As a consequence, analyzing the F-Measure results, every method presented a better balance between Precision and Sensitivity when given 40% to 80% of inside annotation. Therefore, in this range of inside annotation, BGrowth surpassed all the other methods.

Exterior annotation variation (Figure 20b): once again, every method increases the Precision and decreases the Sensitivity as the distance from the ground-truth increases. BGrowth and GrowCut presented the best Precision from 9 to 27 pixels, while presetting a lower Sensitivity from 21 to 27 pixels. Analyzing the F-Measure, BGrowth presented the best

Figure 20 – Results of the segmentation assessment over annotation variation at the interior/exterior of each vertebral body.



(a) Interior variation.

(b) Exterior variation.

Source: Elaborated by the author.

results from 6 to 21 pixels. Considering a real case scenario: (a) a pixel distance smaller than 6 rarely happens, as it is almost the ground-truth itself. (b) a pixel distance greater than 21 will mostly appear in some parts of the vertebral body (considering a sloppy or rough annotation, rectangle-like), for example, in region with concavity or convexity.

3.4 Final Considerations

The semi-automatic segmentation of Vertebral Compression Fractures (VCFs) is a challenging task. In general, VCFs present several regions with heterogeneous intensities within the same vertebral body. To cope with this challenge, we proposed an efficient and accurate semi-automatic segmentation method called BGrowth, which balances the weights of the regions in expansion. We used simple annotations and achieved 96.1% average Accuracy with low

processing time.

Concerning the objectives of this Doctorate research, in this chapter we approached the **Research Problem 1:** Segmenting fractured or deformed vertebral bodies. Based on the results presented throughout the chapter, we answer the initial formulated research questions as follows:

a) *How can we segment fractured vertebral bodies from MRI exams with accurate boundary delineation with an easier and faster manual interaction than the state-of-the-art approaches?* **Answer:** BGrowth can better delineate borders even at regions that present small intensities transitions. This is specially meaningful in the VCF scenario, due to the non-homogeneous gray intensities within the same vertebral body. BGrowth presents acceptable results even with rough or "sloppy" annotations and significantly outperformed other well-known methods from the literature, presenting the best balance between Precision and Sensitivity. This indicates that our method does not over-segments or under-segments fractured or healthy vertebral bodies as much as the competitors. Thus, our proposed BGrowth can be used to segment fractured vertebral bodies with accurate boundary delineation, demanding less effort of the specialist on stating seed points, making the manual annotation process faster and, at the same time, yielding precise results.

In the next Chapter, we cover the use of BGrowth on a volumetric scenario, in which we investigate the reduction on the number of annotated slices. Therefore, we step forward into the **Research Problem 2**: Reducing manual interaction on the segmentation of lumbar vertebral bodies in MRI exams.

CHAPTER

4

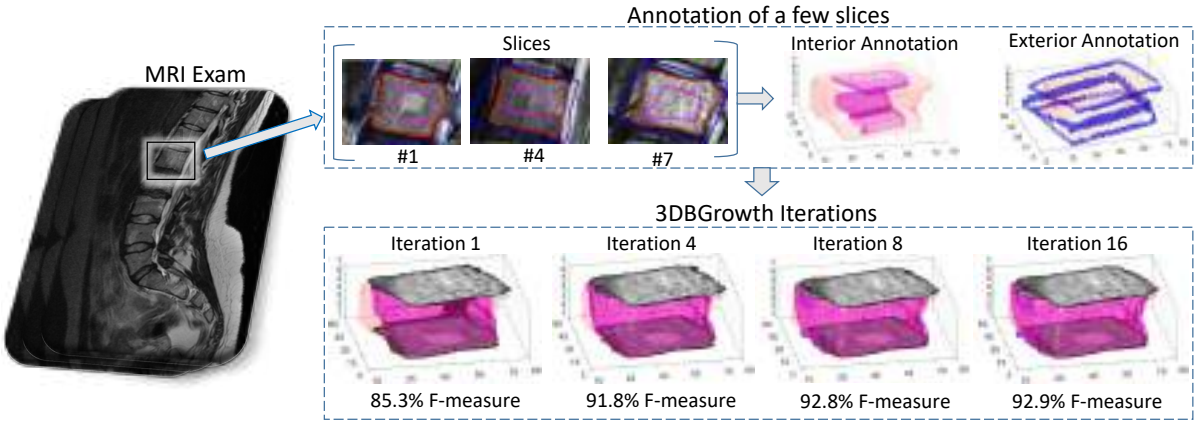
VOLUMETRIC BALANCED GROWTH USING THE SLOPE COEFFICIENT

As presented in Chapter 3, BGrowth method was proposed for the semi-automatic segmentation of crushed vertebral bodies in single slices. BGrowth surpassed several methods from the literature, including GrowCut, presenting better delineated borders on regions with small intensities transitions. For this reason, in Section 4.1, we show how to extend BGrowth to deal with the reconstruction of volumetric exams (3D), introducing the 3DBGrowth method. Moreover, also in Section 4.1, we introduce the Negative Slope Coefficient (NSC), in which we extrapolate the specialists' annotation up to a fixed limit without losing performance/accuracy, aiming at reducing the total time spent on manual annotation, as we explore next. We show the experimental results in Section 4.3, using a representative database with 17 volumetric MRI exams. 3DBGrowth significantly outperformed the state-of-the-art methods and, on average, only 37% of the total slices with vertebral body content must be annotated without losing performance/accuracy. Moreover, 3DBGrowth works well with imprecise seed points, which reduces the time spent on manual annotation by the specialist. Finally, Section 4.4 concludes this chapter.

4.1 Slope Coefficient and 3DBGrowth method

The usual approach of annotating or stating seeds for segmenting medical images can be cumbersome for large 3D exams. Thus, we focus on minimizing the human effort to segment and reconstruct 3D exams built on 2D slices. As illustrated in Figure 21, depending on the MRI exam, not all slices have to be manually annotated by the user to process the 3D reconstruction. If the exams present a small spacing between slices (considering annotations on the sagittal plane), several similar slices do not need to be annotated. This can be assessed by analyzing the negative slope coefficient – NSC (VANNESCHI *et al.*, 2004), detailed in Section 4.1.1. The

Figure 21 – Examples of slices annotation for a single vertebral body (Exam AKa2, L5) and 3DBGrowth's iterations. Ground-truth, interior and exterior annotation in red, magenta and blue, respectively.



Source: Adapted from Ramos *et al.* (2019a).

NSC can provide the best trade-off between annotation time and quantitative measures. Besides, by using a segmentation approach that does not require detailed interior/exterior annotation, such as 3DBGrowth (Section 4.1.2), the total time spent on annotation is greatly diminished. In addition, 3DBGrowth generally requires a simple rectangle-like annotation for the segmentation of individual vertebral bodies. For example, only 3 out of 7 slices were annotated in Figure 21. In average, each slice took 6.5 seconds for annotation and 3DBGrowth took 0.64 seconds to process all 7 slices with only 16 iterations. Summing up, the whole process took $3 \times 6.5 + 0.65 = 20.65$ seconds and achieved a result close to the Ground-Truth.

4.1.1 The Negative Slope Coefficient (NSC)

The NSC can be calculated using the slope between two points, say $\{x_1, y_1\}$ and $\{x_2, y_2\}$, which provides the rate of change between the two points as follows:

$$NSC = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \,. \tag{4.1}$$

When the slope between two values of annotation time gets close to a straight horizontal line, there is no gain in annotation time, in other words, the closer the slope gets to 0, the lower the annotation time gain.

4.1.2 Volumetric BGrowth (3DBGrowth)

In our proposed 3DBGrowth method, detailed in Algorithm 2, we initially consider the segmentation of foreground and background in gray-scale images. That is, considering a digital image I and its annotations/labels as a matrix L, both with dimension $M \times N \times Z$, representing the number of rows, columns, and slices, respectively. Each entry in L has value -1 (background), 0 (unlabeled) or 1 (foreground).

Algorithm 2: 3DBGrowth method. Source codes available in Section A.2.

Input: Image I and labels matrix L.
Output: Segmented binary image L == 1.

1 **for** $\forall(i,j,z)$ **do** `// Step 1: Set initial weights`
2 **if** $L(i,j,z) \neq 0$ **then**
3 $W(i,j,z) \leftarrow 1.0$
4 **for** $\forall(i,j,z)$ **and** $L(i,j,z) \neq 0$ **do** `// Step 2: For each labeled voxel`
5 **for** $\forall(i_n,j_n,z_n)$ **do** `// and its 26-Neighbors (`i_n`, `j_n`, `z_n`)`
6 $s \leftarrow W(i,j,z) \times \left[1 - |I(i,j,z) - I(i_n,j_n,z_n)| \;/\; \max_{\forall i,j,z} I(i,j,z)\right]$
7 **if** $s > W(i_n,j_n,z_n)$ **then**
8 $W(i_n,j_n,z_n) \leftarrow 0.5 \times W(i_n,j_n,z_n) + 0.5 \times s$
9 $L(i_n,j_n,z_n) \leftarrow L(i,j,z)$

Initially, each entry in a weight matrix W (line 1), with the same dimensions as I and L, is set to 1.0 for seeds points and 0 otherwise (lines 2 and 3). Then, for each labeled voxel (i,j,z) (line 4) and each one of its 26 neighbors (i_n,j_n,z_n) (line 5), a strength factor s is calculated (line 6). Here, the absolute intensity difference is normalized by the maximum intensity in the image and subtracted by 1. Finally, s is multiplied by the current weight $W(i,j,z)$, which produces values within $[0,1]$. If the strength s is greater than the neighbor's strength $W(i_n,j_n,z_n)$ (line 7), then the neighbor's strength is averaged with the new strength s (line 8) and its label receives the label of the voxel (i,j,z) (line 9).

The process in line 4 repeats until the algorithm converges or for a fixed number of iterations defined by the user.

4.2 Materials and methods

In this section, we present the materials and methods used in the experimental design. First, in Section 4.2.1, we describe a volumetric database, which is composed of exams coming from seven different hospitals, presenting a variety of scanning parameters as well as a few pathologies. Then, in Section 4.2.2, we present the parameters settings used in the experimental design, which presents the computational set-up as well as how the manual annotations were performed. Next, we detail the state-of-the-art methods considered in the analysis in Section 4.2.3. Finally, in Section 4.2.4, we present the segmentation assessment to validate the experimental results.

4.2.1 Patient database

We used a database reported by Zukić *et al.* (2014) comprised of 26 exams, in which 22 exams are from pathological patients, and four are from healthy ones. Zukić *et al.* (2014) made

Table 5 – Details of the exams used to compose the dataset reported by Zukić *et al.* (2014).

MRI Exam	MRI Seq.	Resolution (voxels)	Resolution (mm^3)	Path.	#VB	Segmented by
DzZ_T1	TI TSE	$512 \times 512 \times 12$	$0.68 \times 0.68 \times 4.4$	None	9	Experienced User
DzZ_T2	T2 TSE	$640 \times 640 \times 12$	$0.55 \times 0.55 \times 4.4$	None	9	Experienced User
AKa2	T2 frFSE	$512 \times 512 \times 15$	$0.70 \times 0.70 \times 4.0$	+	9	Experienced User
AKa3	T1 FSE	$512 \times 512 \times 15$	$0.70 \times 0.70 \times 4.0$	+	9	Experienced User
AKa4	TIRM	$512 \times 512 \times 15$	$0.70 \times 0.70 \times 4.0$	+	9	Experienced User
AKs5	T2 frFSE	$512 \times 512 \times 15$	$0.70 \times 0.70 \times 4.0$	+	9	Experienced User
AKs6	T1 FSE	$512 \times 512 \times 15$	$0.70 \times 0.70 \times 4.0$	+	9	Experienced User
AKs7	TIRM	$512 \times 512 \times 15$	$0.70 \times 0.70 \times 4.0$	+	9	Experienced User
AKs8	T1 FSE	$512 \times 512 \times 15$	$0.70 \times 0.70 \times 4.0$	+	9	Experienced User
S01	T2 SE	$640 \times 640 \times 15$	$0.47 \times 0.47 \times 3.8$	SL	7	Neurosurgeon
S02	T2 SE	$640 \times 640 \times 15$	$0.47 \times 0.47 \times 3.8$	SL	8	Neurosurgeon
F02	T2 SE	$768 \times 768 \times 18$	$0.50 \times 0.50 \times 3.8$	VF,SL	9	Neurosurgeon
St1	T2 SE	$704 \times 704 \times 20$	$0.50 \times 0.50 \times 3.8$	St	8	Neurosurgeon
F04	T2 TSE	$448 \times 448 \times 23$	$1.12 \times 1.12 \times 3.0$	VF+	17	Experienced User
AKs3	T2 frFSE	$512 \times 512 \times 25$	$0.70 \times 0.70 \times 4.0$	+	10	Experienced User
F03	T2 TSE	$320 \times 320 \times 25$	$1.19 \times 1.19 \times 3.3$	VF	7	Neurosurgeon
C002	T2 TSE	$448 \times 448 \times 31$	$1.12 \times 1.12 \times 3.3$	Sco	13	Neurosurgeon

Source: Adapted from Zukić *et al.* (2014).

Note – Seq.: Sequence. #VB: Total of segmented vertebral bodies in the exam. Path.: Pathologies, such as scoliosis (Sco), vertebral fracture (VF), spondylolisthesis (SL), and + represents other pathologies not diagnosed from segmentation.

17 anonymized lumbar MRI exams publicly available among the 26 exams. The images came from different hospitals and scanning machines. The MRI exams vary from the sacrum (S1) to the mid-thoracic (T6-T12) with several health conditions, such as scoliosis, vertebral fracture, and spondylolisthesis.

The MRI exams have 3.24 ± 0.36 mm of slice thickness and 3.87 ± 0.36 mm of spacing between slices. The reference segmentations were produced by manually tracing the vertebral body edges in the primary acquisition plane (23 sagittal, two axial, one coronal). Neurosurgeons and three experienced users under expert supervision made the manual tracing in three to six min per vertebra. The exams present varied MRI sequences, anisotropy factors, and at least one of the three identifiable pathologies: scoliosis, vertebral fracture, and spondylolisthesis. Summing up, Zukić *et al.* (2014) provided 160 segmented vertebral bodies. Since this database came from seven different hospitals and the exams were taken by nine scanning machines (high anisotropy of voxel), in Table 5 we report the details of each one of the 17 exams used to compose the database. No pre or post-processing technique were applied to assure the same conditions for all segmentation methods. All exams were normalized into 256 gray levels (8 bits/pixel) using Equation 2.2.

4.2.2 Parameters settings

The maximum number of iterations was set to 50 for GrowCut and 3DBGrowth.

4.2.2.1 Computational set-up

The experiments were performed on a 2.40GHz Intel(R) Core(TM) i7 CPU and 8GB RAM machine, using Matlab(R) version 2018a.

4.2.2.2 Manual annotations

The initial interior and exterior annotation were performed in a "sloppy" way, i.e., no detailed boundary for accentuated curves were drawn. In general, the annotation looks like a rectangle for the background and a simple line for the foreground. The annotations were performed on each slice on every exam and, to diminish computational processing, each exam were cropped using the convex hull of the exterior annotation. In average, the manual annotation of every slice in a single exam took 11 minutes per exam, which includes, in average, 9 vertebral bodies per exam and an average of 12 slices per vertebral body.

We performed annotations in two ways. First, we annotated the whole exam, as illustrated in Figure 22 from (a) to (c). Then we annotated individual vertebral bodies separately, as illustrated in Figure 22 from (d) to (e).

4.2.3 Other similar methods

In order to evaluate the performance of 3DBGrowth in a volumetric scenario, we compared it with the GrowCut approach, which has been widely used for the task of vertebral bodies segmentation (EGGER; NIMSKY; CHEN, 2017). Since FastGrowCut is an approximation of the original GrowCut, presenting a lower Accuracy (ZHU *et al.*, 2014), we consider only GrowCut in the experiments.

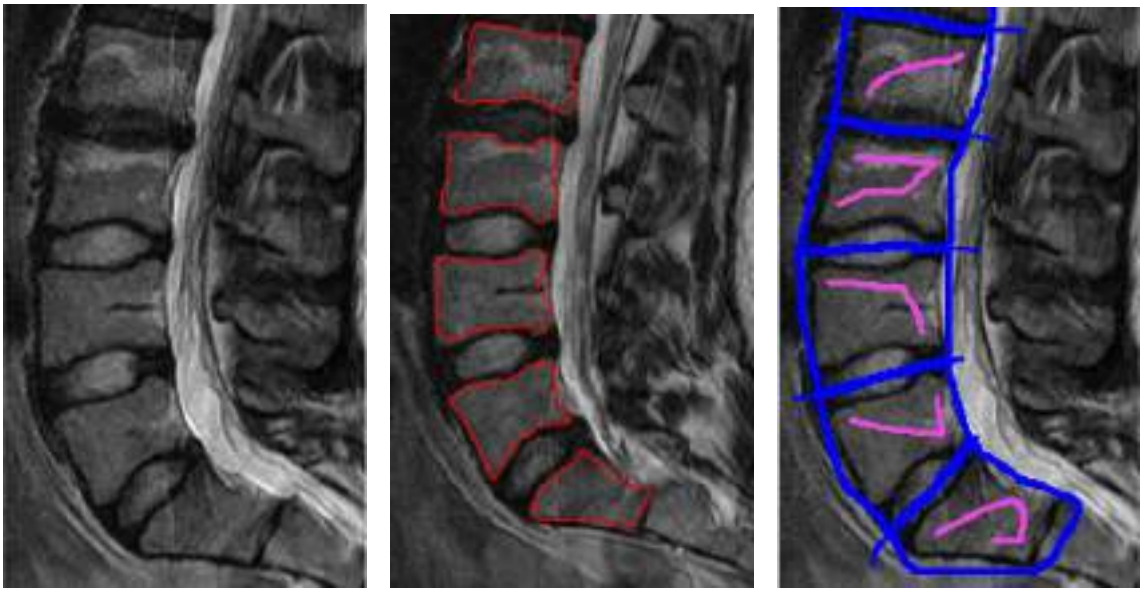
4.2.4 Segmentation assessment

To validate 3DBGrowth's segmentation results, we employed the measures of F-Measure (Equation 2.11), Jaccard (Equation 2.10), Average Hausdorff (Equation 2.14), and Run-Time.

4.3 Experiments, results and discussion

We present our experimental design in four main parts, as illustrated in Figure 23 and detailed as follows. First, in Section 4.3.1, we assessed the performance of each segmentation method using the whole exam. Then, in Section 4.3.2, we validate the performance of 3DBGrowth and GrowCut when not all slices are annotated. Next, in Section 4.3.3, we segmented the vertebral

Figure 22 – Example of sloppy annotation for a few vertebral bodies in one slice. From (a) to (c), whole exam annotation (Aka2, eighth slice). From (d) to (e), annotation for a single vertebral body (St1, L2, tenth slice). Ground-truth in red, interior and exterior annotations in magenta and blue, respectively.



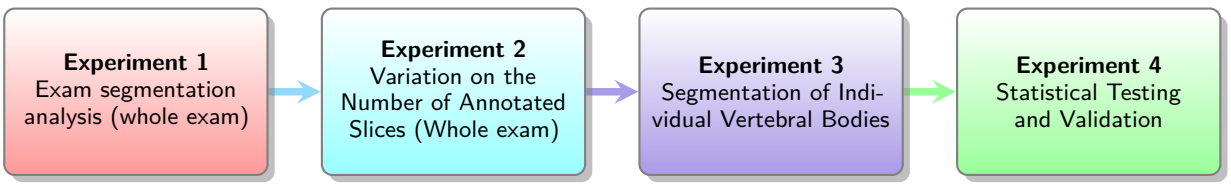
(a) Original. (b) Ground-Truth. (c) Annotation.

(d) Original. (e) Ground-Truth. (f) Annotation.

Source: Adapted from Ramos *et al.* (2019a).

bodies one-by-one for every segmentation method considered in the experiment. In this study, we considered the annotation of individual vertebral bodies instead of the whole exam, as performed in the two previous experiments. Finally, in Section 4.3.4, we applied statistical testing to detect any significant difference between the results of the segmentation techniques.

Figure 23 – Experimental design to validate 3DBGrowth.



Source: Elaborated by the author.

4.3.1 Exam segmentation analysis

Table 6 shows the average F-Measure, Jaccard, and Run-Time in seconds for each one of the 17 exams in the database (whole exam annotation). 3DBGrowth presented on average 81% for F-Measure and 68% for Jaccard while GrowCut presented 76% and 61%, respectively. Thus, 3DBGrowth presented higher F-Measure and Jaccard percentages than GrowCut for all exams, achieving up to 5% and 7% of F-Measure and Jaccard gain, respectively. Moreover, considering F-Measure and Jaccard, 3DBGrowth's standard deviation is slightly lower. Analyzing the Run-Time, very often, 3DBGrowth presented a lower average processing time than GrowCut.

Table 6 – Segmentation assessment between 3DBGrowth and GrowCut, considering annotations on every slice of each volumetric exam.

	F-Measure (%)			Jaccard (%)			Run-Time (s)	
Exam	3DBGrowth	GrowCut	Gain	3DBGrowth	GrowCut	Gain	3DBGrowth	GrowCut
DzZ_T1	**85**	80	4.86	**74**	67	7.0	**18**	21
DzZ_T2	**82**	77	4.20	**69**	63	5.8	**27**	31
AKa2	**82**	77	5.17	**69**	62	7.1	**27**	**27**
AKa3	**78**	73	4.78	**64**	58	6.2	**27**	29
AKa4	**80**	73	7.10	**67**	58	9.4	**26**	27
AKs5	**84**	78	6.54	**73**	63	9.2	**23**	24
AKs6	**84**	79	5.44	**73**	65	7.8	**21**	24
AKs7	**80**	73	7.6	**67**	57	9.9	**21**	24
AKs8	**81**	78	3.39	**68**	64	4.7	**18**	21
S01	**85**	82	2.91	**74**	70	4.3	**44**	50
S02	**83**	78	4.97	**70**	63	6.9	**26**	32
F02	**78**	74	3.63	**64**	59	4.7	**48**	55
St1	**83**	80	2.73	**71**	67	3.9	**61**	67
F04	**78**	75	3.42	**64**	60	4.5	**13**	14
AKs3	**80**	73	6.42	**66**	58	8.4	40	**37**
F03	**80**	77	3.57	**67**	62	4.8	**08**	09
C002	**71**	65	5.85	**55**	48	6.7	16	**14**
Average	**81**	76	4.9	**68**	61	6.6	**27**	30
Std. Dev.	**3.4**	3.9	1.5	**4.7**	5.0	1.9	**13.7**	15.2

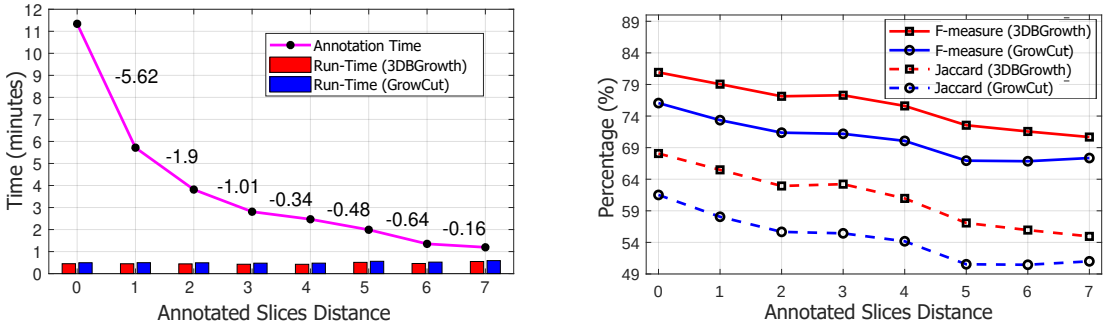
Source: Adapted from Ramos *et al.* (2019a).

Note – Best results highlighted in **bold**.

4.3.2 Variation on the number of annotated slices

We considered the annotations of the whole exam, however, leaving a few slices without annotation. To do so, we set a slice distance, which manages the number of non-annotated

Figure 24 – Run-Time and segmentation assessment between 3DBGrowth and GrowCut over variations on the number of slices manually annotated. NSC values highlighted over the magenta line in (a).



(a) Annotation time and Run-Time results.

(b) F-Measure and Jaccard results.

Source: Ramos *et al.* (2019a).

slices between two annotated slices. For example, a slice distance of 1 implicates that only one slice was left without annotation between two annotated slices. The slice distance started at 0, increased by 1, up to 7.

Figure 24 reports the results for the variation on the number of annotated slices. As the slice distance increases, the average annotation time decreases and the Run-Time keeps almost steady for both methods. Also, F-Measure and Jaccard drops slowly for both methods. However, 3DBGrowth presented best results than GrowCut for both measures. Considering the Negative Slope Coefficient (NSC), detailed in Section 4.1.1, highlighted over the magenta line in Figure 24a, by using a threshold of -1, the best slice distance would be 3, which presents the best trade-off between annotation time and F-Measure/Jaccard in Figure 24b.

4.3.3 Segmentation of individual vertebral bodies

To speed-up the annotation process, we have considered a slice distance of three for this experiment. Table 7 reports the segmentation results for individual vertebral bodies. GrowCut and 3DBGrowth presented equal mean Run-Time and 3DBGrowth presented better mean F-Measure, Jaccard and Average Hausdorff than GrowCut.

Analyzing the average number of annotated slices per vertebra in Table 8, in average, only 37% of the total slices with vertebral body content were annotated. Thus, the manual annotation process was improved, taking, in average, 36 seconds to annotate each vertebral body.

To further investigate the results presented in Table 7, we conducted a statistical test, as detailed next.

Table 7 – Comparison between 3DBGrowth and GrowCut for the F-Measure, Jaccard, Average Hausdorff in voxels and Run-Time in seconds.

		F-Measure (%)		Jaccard (%)		Hausdorff (vox.)		Run-Time (s)	
	Vertebrae	3DBGrowth	GrowCut	3DBGrowth	GrowCut	3DBGrowth	GrowCut	3DBGrowth	GrowCut
Thoracic	T6	**88**	87	**79**	78	**3.16**	4.00	**0.15**	0.16
	T7	**85**	83	**73**	69	**78.8**	79.1	7.95	**6.77**
	T8	**86**	85	**76**	72	79.3	**79.2**	**7.55**	8.37
	T9	**81**	80	50	**52**	80.0	**79.9**	**7.83**	8.38
	T10	**87**	85	**80**	76	**26.2**	26.9	2.95	**2.83**
	T11	**86**	84	**77**	73	**6.09**	6.99	0.91	**0.88**
	T12	**89**	86	**79**	76	**5.63**	6.88	**1.30**	1.36
Lumbar	L1	**89**	87	**78**	76	**6.56**	7.31	1.57	**1.56**
	L2	**88**	86	**79**	76	**6.07**	7.75	**1.52**	1.57
	L3	**86**	85	**75**	72	**6.40**	7.49	**1.68**	1.83
	L4	**88**	86	**76**	74	**7.16**	7.65	**1.77**	1.88
	L5	**87**	85	**76**	74	**7.04**	8.42	**2.33**	2.39
Sacral S1		**88**	86	**79**	76	**6.18**	7.57	**1.74**	1.88
Average		**87**	85	**77**	74	**7.24**	7.72	1.52	1.52
Std. Dev.		**0.07**	0.06	**0.08**	0.08	**4.85**	5.00	1.27	1.27

Source: Adapted from Ramos *et al.* (2019a).

Note – Best results highlighted in **bold**.

4.3.4 Statistical testing

Considering that the resulting values of each measure had several similar values, we employed the Kolmogorov-Smirnov (MASSEY, 1951) and the null hypothesis that the data follows a normal distribution was rejected for all measures. Then, we applied the Wilcoxon (WILCOXON; KATTI; WILCOX, 1970) test, in which 3DBGrowth presented significantly better F-Measure, Jaccard and Average Hausdorff than GrowCut. For the Run-Time, there was no significant difference, which implicates that both methods presented comparable processing time.

4.4 Final Considerations

In this chapter we have tackled the challenging task of semi-automatic segmentation of vertebral bodies in a volumetric scenario. To obtain a proper 3D reconstruction of the vertebral body, one has to pay attention on allowing a fast and accurate segmentation of slices. To cope with this challenge, we have used the negative slope coefficient in order to reduce the total number of annotated slices. We extrapolated the specialists' annotations from a slice to its neighbors up to a given limit without losing accuracy. We have proposed 3DBGrowth method, which

Table 8 – Comparison of the number of annotated slices, considering a slice distance of three.

Vertebrae		Slices annotated	out of	Annotation time (seconds)
Thoracic	T6	$3.0 \pm .00$	$7.0 \pm .00$	$28.7 \pm .00$
	T7	$3.0 \pm .00$	$7.0 \pm .00$	$32.5 \pm .00$
	T8	$3.0 \pm .00$	$7.0 \pm .00$	34.6 ± 11.6
	T9	$3.0 \pm .00$	$7.0 \pm .00$	30.0 ± 6.2
	T10	$2.5 \pm .52$	6.7 ± 3.2	25.9 ± 7.1
	T11	$3.1 \pm .53$	8.5 ± 2.6	30.2 ± 5.8
	T12	$3.6 \pm .50$	9.6 ± 1.9	34.5 ± 8.1
Lumbar	L1	$3.9 \pm .78$	10.2 ± 2.2	36.8 ± 9.2
	L2	$4.2 \pm .75$	10.9 ± 2.3	38.6 ± 10.1
	L3	$4.3 \pm .86$	11.6 ± 2.1	40.0 ± 9.5
	L4	$4.5 \pm .62$	12.5 ± 2.8	39.3 ± 6.3
	L5	$4.5 \pm .72$	12.5 ± 3.0	39.8 ± 7.8
Sacral S1		$4.1 \pm .70$	10.9 ± 3.3	35.8 ± 6.3
Mean		$4.1 \pm .84$	10.9 ± 2.9	35.9 ± 8.8
		37% annotated		

Source: Adapted from Ramos *et al.* (2019a).

Note – The total number of slices considers the number of slices with vertebral body content on each ROI.

significantly outperformed GrowCut and kept comparable running time in the experimental results. Moreover, on average, only 37% of the slices with vertebral body content had to be annotated, consequently, making the manual annotation process faster. We also highlight that 3DBGrowth presented the best results even with simple/sloppy seed points, which demands less effort on the annotation process. As a final remark, concerning the objectives of this Doctorate research, in this chapter, we consider our contribution as a **first step** in the **Research Problem 2:** Reducing manual interaction on the segmentation of lumbar vertebral bodies in MRI exams. In the next chapters, we explore the next steps towards addressing **Research Problem 2**.

CHAPTER

5

CLEVER SEGMENTATION – CLEVERSEG

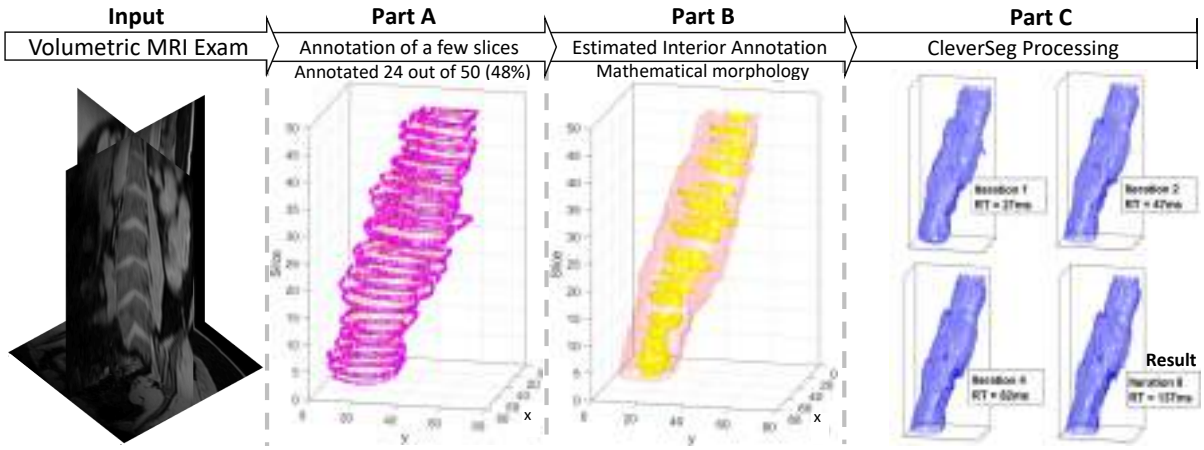
In this Chapter, in Section 5.1, we introduce a new segmentation approach, named CleverSeg), and we take advantage of the Mean Absolute Error (MAE) to select slices to be annotated. MAE does not depend on the physical spacing between slices to identify the most representative slices to be annotated. Moreover, we automatically estimate the interior annotation so that only the outside annotation needs to be placed, what reduces the time spent on manual annotations. Next, in Section 5.2 we detail the materials and methods used to validate our approach. After that, in Section 5.3, the experimental results show that our proposed approach is able to speed up the manual annotation process in up to 50% by annotating only a few representative slices, without loss of accuracy. By annotating only the outside region, the process can be further speed up by another 50%, reducing the total time to only 25% of the previously required. Thus, the total time spent on manual annotation is reduced by up to 75%, and, since human interaction is greatly diminished, allows a more productive and less tiresome activity. Finally, Section 5.4 concludes this chapter.

5.1 CleverSeg: The Proposed Method

Placing (annotating) seeds appropriately in MRI data is a crucial initial step to produce accurate paraspinal muscles segmentation. Nevertheless, due to the 3D nature of MRI data and the complex structure of the human spine, it becomes a very difficult and tiresome task. In our approach, we focus on minimizing the specialist's effort to segment and reconstruct MRI exams built on 2D slices. To this end, we contribute in three main aspects, as depicted in Figure 25 and detailed as follows.

Part A - Annotation of a few slices. We employ the MAE to verify the slices that look alike and, therefore, are not required to be manually annotated. The error is calculated slice vs. slice and it is not dependent on the physical spacing between slices.

Figure 25 – Proposed pipeline: ground-truth in red, exterior annotation in magenta, estimated interior annotation in yellow and CleverSeg segmentation result in blue. RT represents the Run-Time.



Source: Adapted from Ramos *et al.* (2019b).

Part B - Automatic estimation of the interior annotation. Given the initial outside annotation of each slice performed in step A, the inside annotation is automatically estimated using mathematical morphology. As a consequence, the time spent on manual annotation is greatly reduced.

Part C - Fast and effective CleverSeg for semi-automatic segmentation. Our proposed CleverSeg method works in a faster and smart way, using only a few iterations. CleverSeg efficiently propagates the annotated slices in Parts A and B into non-annotated slices. Therefore, the processing time is reduced while keeping high accuracy.

We explore the three main aforementioned contributions next.

5.1.1 Annotations of a few slices

Considering a volumetric exam $I = \{I_1, I_2, .., I_i, .., I_n\}$, in which I_i represents the i-th slice in the exam, similar sequential slices can be avoided from the manual annotation process. To do so, we choose the most representative slices to be annotated by using MAE. Given two distinct slices from a volumetric MRI exam I, say I_1 and I_2, MAE measures the difference between the slices as follows:

$$MAE(I_1, I_2) = \frac{1}{r \times c} \sum_{x=1}^{r} \sum_{y=1}^{c} |I_1(x,y) - I_2(x,y)| \,, \tag{5.1}$$

in which both slices (matrices I_1 and I_2) have the same size $r \times c$. Note that the closer $MAE(I_1, I_2)$ is to zero, the more similar the images are (SILVA *et al.*, 2019; WANG; LU, 2018).

The most representative slices may be selected in a bottom-up manner in three steps, as we detail next.

Step 1: Define I_1 to be manually annotated and set I_1 as the initial slice for comparison with the next slice I_2.

Step 2: If the error $MAE(I_1, I_2)$ is smaller than a threshold η, then I_1 is compared with the next slice I_3. This process repeats until the i-th slice I_i, such that $MAE(I_1, I_i) \geq \eta$. Then, define I_i as the initial slice for comparison and go back to **Step 1**.

Step 3: Repeat **Steps 1 and 2** until reaching the final slice I_n. The last slice is always set to be manually annotated.

Note that, the first (I_1) and last (I_n) slices are always set to be manually annotated. The number of non-annotated slices in between I_1 and I_n may vary according to the threshold η. The value of the threshold η can be tuned manually.

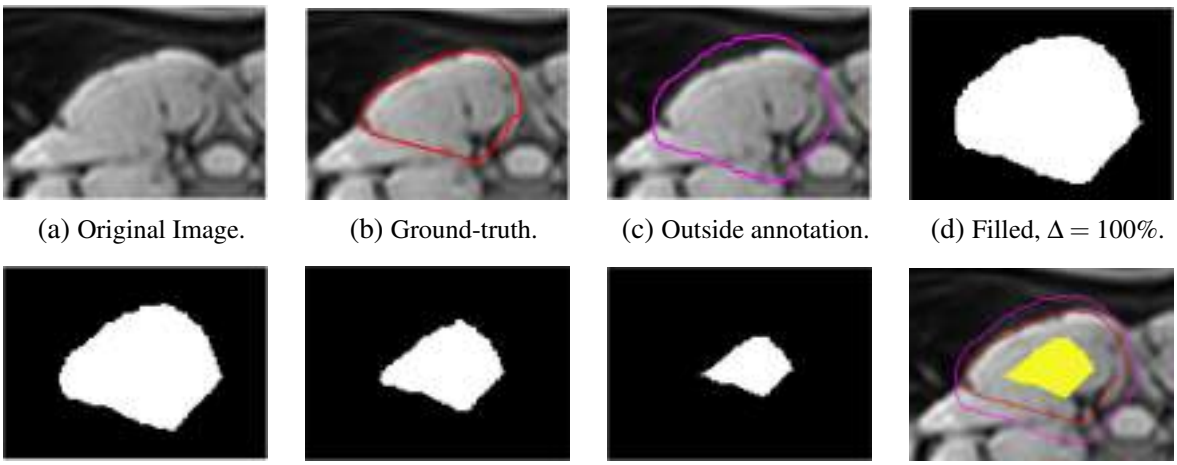
For the purposes of analysis, we define a Percentage of Annotated Slices (*PAS*), which is the number of annotated slices (*#annotatedSlices*) divided by the total number of slices in the exam with muscle content (*#slices*):

$$PAS = \frac{\#annotatedSlices}{\#slices}$$

5.1.2 Estimation of the interior annotation

Given the outside annotation of the i-th slice (I_i) as exemplified in magenta in Figure 26c, we estimate the inside annotation using morphological operations (GONZALEZ; WOODS, 2008) in four steps:

Figure 26 – Illustration of estimating interior annotations. Sample slice with ground-truth in red,exterior annotation in magenta, and estimated interior annotation in yellow ($\Delta = 20\%$).



(a) Original Image. (b) Ground-truth. (c) Outside annotation. (d) Filled, $\Delta = 100\%$.

(e) 3 Erosions, $\Delta = 70\%$. (f) 6 Erosions, $\Delta = 34\%$. (g) 8 Erosions, $\Delta = 16\%$. (h) Estimated annotation.

Source: Adapted from Ramos *et al.* (2019b).

Step I: Represent the manually annotated outside region of I_i as a 2D binary mask k_i. Considering that the region in k_0 is always a closed boundary, fill this boundary with ones (white pixels), as shown in Figure 26d.

Step II: Set k_0 as the initial interior annotation and t_0 as the total number of white pixels in k_0 and apply an erosion operation over k_0 using a 5×5 square structuring element, which results into a new mask k_1. As a consequence, the number of white pixels in k_1 is reduced to t_1. The 5×5 square structuring element allows a smooth erosion without losing the main shape of the region.

Step III: Apply the same erosion over k_1, which results in k_2 with t_2 white pixels. Repeat the erosion process until the i-th iteration, resulting in the mask k_i with t_i white pixels. The erosion process stops when the percentage of remaining white pixels in the eroded mask (k_i), associated to the initial mask k_0, is lower than a threshold Δ:

$$\Delta > \frac{t_i}{t_0} \tag{5.2}$$

Step IV: Finally, use the k_i mask as the inside annotation. The entire erosion process is depicted in Figure 26e to Figure 26g, in which Δ is set to 20%, yielding the estimated inside annotation shown in yellow in Figure 26h.

We further investigated this interior annotation estimation by presenting Fast Inside Estimation (FINE), in Appendix A, a study case with VCFs exams on automatic inside annotation estimation. FINE automatically finds the suitable thresholds so that erosions and skeletonization can be used to determine the inside annotation without using any fixed or empirically defined limits;

5.1.3 Fast and effective CleverSeg for semi-automatic segmentation

Initially, we segment gray scale volumes into foreground and background. However, CleverSeg works for more than two regions, and the difference between voxels intensities can be easily adapted for multi-channel (color) images.

Considering a volume I and a matrix L of corresponding annotations/seeds, both with dimensions $M \times N \times Z$, representing the number of rows, columns, and slices, respectively. The total number of voxels in I is represented by n. The entries of L may have values 0 (unlabeled), 1 (background or outside annotation) or 2 (foreground or inside annotation).

As Algorithm 3 shows, we divide CleverSeg method in three main steps, explored as follows.

Step 1: A weighted matrix W is initialized, so that every entry $\forall w_i \in W$ is filled with ones for corresponding labeled entries $l_i \in L$ (inside/outside annotation) and zeros otherwise (lines

1 to 3):

$$w_i = \begin{cases} 1.0 & \text{if } l_i \neq 0, \\ 0.0 & \text{otherwise.} \end{cases} \tag{5.3}$$

The maximum voxel intensity is calculated/represented as mp (line 4).

Step 2: We analyze every voxel $\forall(i,j,z) \in I$ and each of its 26 neighbors $\forall(i_n, j_n, z_n)$ (line 5) as follows. The absolute difference between the voxel intensity ($I(i,j,z)$) and its neighbor's ($I(i_n, j_n, z_n)$) is calculated and subtracted by the maximum voxel intensity (mp). The result is represented as h (line 7). A strength s is calculated line 8) normalizing h by mp and multiplying with the current voxel strength $W(i,j,z)$.

Step 3: If the difference between s and the neighbor's current strength $W(i_n, j_n, z_n)$ (line 9) is greater than a threshold (θ), then, the strength $W(i_n, j_n, z_n)$ is averaged with the new strength s (line 10) and its label $L(i_n, j_n, z_n)$ updates (line 11).

Algorithm 3: CleverSeg method.

Input: Volumetric exam I and annotation/labels matrix L.
Output: Segmented image (grown regions in L).

```
1  for ∀(i,j,z) do                                   // Step 1: Set initial weights
2      if L(i,j,z) ≠ 0 then
3          W(i,j,z) ← 1.0
4  mp ← max v_i, ∀v_i ∈ V, i = {1,2,3,...,n}
5  for ∀(i,j,z) ∈ I and L(i,j,z) ≠ 0 do          // Step 2: For each labeled voxel
6      for ∀(i_n,j_n,z_n) do                      // and its 26-Neighbors (i_n, j_n, z_n)
7          h ← mp − |I(i,j,z) − I(i_n,j_n,z_n)|
8          s ← W(i,j,z) × h/mp
9          if (s − W(i_n,j_n,z_n)) > θ then       // Step 3: manage averaged values
10             W(i_n,j_n,z_n) ← 0.5 × W(i_n,j_n,z_n) + 0.5 × s
11             L(i_n,j_n,z_n) ← L(i,j,z)
```

Steps 1 and 2 repeat for a maximum number of iterations or until the algorithm converges. The threshold θ manages averaged values in order to avoid iterations that do not contribute to a better segmentation. For example, it can be used to avoid balancing (averaging) values that only change the third or fourth decimal place.

5.2 Materials and methods

In this section, we present the materials and methods used in the experimental design. First, in Section 5.2.1, we describe a meaningful volumetric database of lumbar muscles, which presents a larger number of slices for each muscle compared to the usual vertebral bodies, therefore, it allows a more precise analysis of total time spent on manual annotation. The

database is composed of four groups of muscles: erector spinae left and right and psoas left and right muscles. In total, there are 216 muscles with reference manual segmentations. Then, in Section 5.2.2, we detail the parameters settings along with the computational set-up as well as how the manual annotations were performed. Next, in Section 5.2.3, we report the state-of-the-art methods considered in the analysis. Finally, in Section 5.2.4, we detail the measures considered to validate the experimental results.

5.2.1 Patient database

Burian *et al.* (2019) presented the *MyoSegmenTUM spine* database, which contains 54 MRI exams. Each exam contains reference manual segmentations of four lumbar muscle groups and lumbar vertebral bodies. The muscles include the erector spinae left and right as well as psoas left and right. Summing up, there are $54 \times 4 = 216$ segmented muscles.

The 54 Caucasians volunteers were 51.6 ± 16.7 years old (15 males and 39 females). The volunteers underwent MRI on a 3T system (Ingenia, Philips Healthcare, Best, Netherlands) using a whole-body coil, built-in 12-channel posterior coil, and a 16-channel anterior coil. Subjects were positioned head-first in a supine position. The segmentations were performed by a board-certified radiologist using the image viewer software MITK (Medical Imaging Interaction Toolkit, Heidelberg, Germany). The manual segmentation took 50 min for each of the paravertebral muscles for each subject. The axial proton density fat fraction (PDFF) maps were used to separately segment the erector spine and the psoas muscle on both sides from the cranial part of L2 to the caudal part of L5.

The exams have an average resolution of $334 \times 334 \times 67.4 \pm 135 \times 135 \times 5.02$ voxels and a spatial resolution of $1 \times 1 \times 3.6 \pm 0 \times 0 \times 0.5$ mm.

5.2.2 Parameters settings

The maximum number of iterations was set to 500 for all algorithms considered in the analysis. However, in general, the algorithms take less than 500 iterations to converge. For CleverSeg, the threshold θ, which controls the approximate "roundness" of the averaged weights during the balancing of region expansion, was set to 1% so that to avoid averaging values on the third decimal place. In order to allow a simple or sloppy-like annotation, the Δ threshold was set to a small value, 20%. Thus, imprecise external annotation do not compromise the estimated internal annotation. For the η threshold, we considered an initial value of 0, increasing by 1.5, up until 10.5, which sums up to 8 thresholds. To assure the same and best conditions for all segmentation methods, no pre or post-processing techniques were applied and the exams were normalized into 256 gray levels (8 bits/pixel) using Equation 2.2. The volumes were not normalized to isotropic resolution to avoid adding noise to the image and manual segmentations.

5.2.2.1 Computational set-up

Every experiment used a 2.40GHz Intel(R) Core(TM) i7 CPU and 8GB RAM machine, using Matlab(R) version 2018a.

5.2.2.2 Manual annotations

Outside annotations were performed in a sloppy-like manner. Interior annotations were estimated using our proposed approach (Section 5.1.2). Figure 26c shows an example of outside annotation as well as the estimated interior annotation.

5.2.3 State-of-the-art segmentation methods

In order to evaluate CleverSeg for the segmentation of spinal muscles, we compared it with 3DBGrowth (RAMOS *et al.*, 2019a) and GrowCut (VEZHNEVETS; KONOUCHINE, 2005). Since Fast GrowCut presents a lower Accuracy than GrowCut (ZHU *et al.*, 2014), we did not consider Fast GrowCut in the analysis.

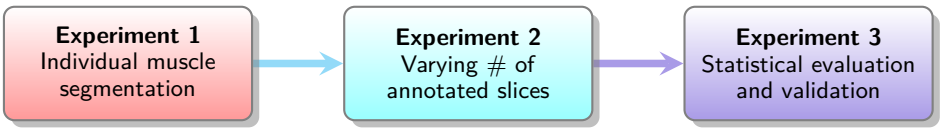
5.2.4 Segmentation assessment

To validate CleverSeg's segmentation results, we employed the following measures: F-Measure (Equation 2.11), Jaccard (Equation 2.10), Average Hausdorff (Equation 2.14), Precision (Equation 2.8), and Run-Time.

5.3 Experiments, results and discussion

In our experimental design, we analyze three main parts, as depicted in Figure 27 and detailed as follows. First, in Section 5.3.1, we assess the segmentation of each muscle. Then,

Figure 27 – Experimental design to validate CleverSeg.



Source: Elaborated by the author.

in Section 5.3.2, we vary the number of annotated slices, based on the error η between slices. Finally, in Section 5.3.3, we statistically evaluate and validate the results.

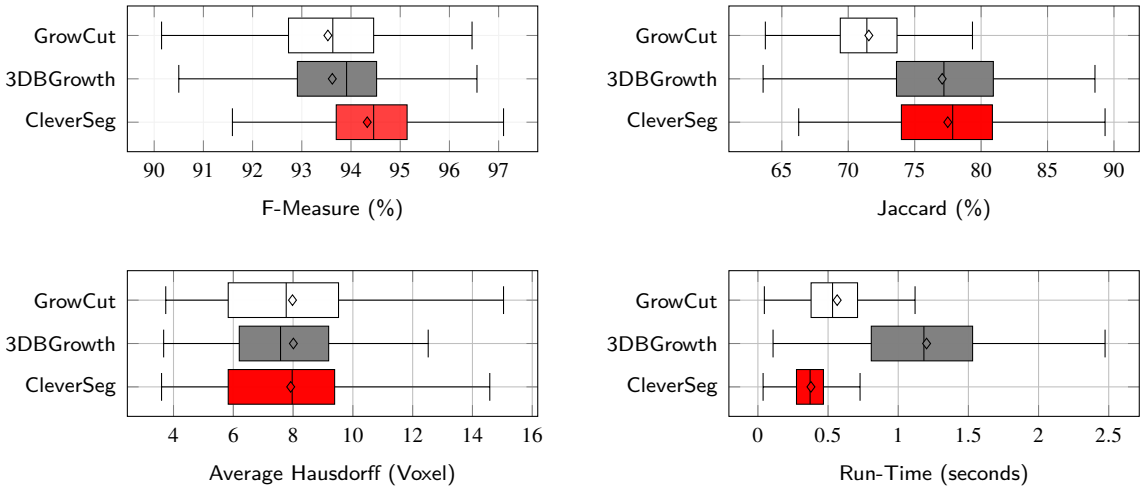
5.3.1 Individual muscle segmentation

We performed the segmentation of each muscle considering the manual annotation of every slice in each MRI exam. Figure 28 reports the results, in which CleverSeg presented better

F-Measure, Jaccard and Run-Time than GrowCut, while keeping a similar Average Hausdorff. Compared to 3DBGrowth, CleverSeg presented better F-Measure, Jaccard and Run-Time, while keeping comparable F-Measure and Average Hausdorff.

CleverSeg presented the fastest Run-Time (377ms), while 3DBGrowth took 1202ms and GrowCut 565ms. The number of iterations were 32 ± 6.5, 99 ± 22.6 and 49 ± 13.2 for CleverSeg, 3DBGrowth and GrowCut, respectively. We highlight that, in Chapter 4 (RAMOS *et al.*, 2019a), 3DBGrowth was tested with 18 slices (in average) and a maximum of 50 iterations. In our experimental database of muscles, in average, there are 67 slices for each exam.

Figure 28 – Comparison between the segmentation approaches.

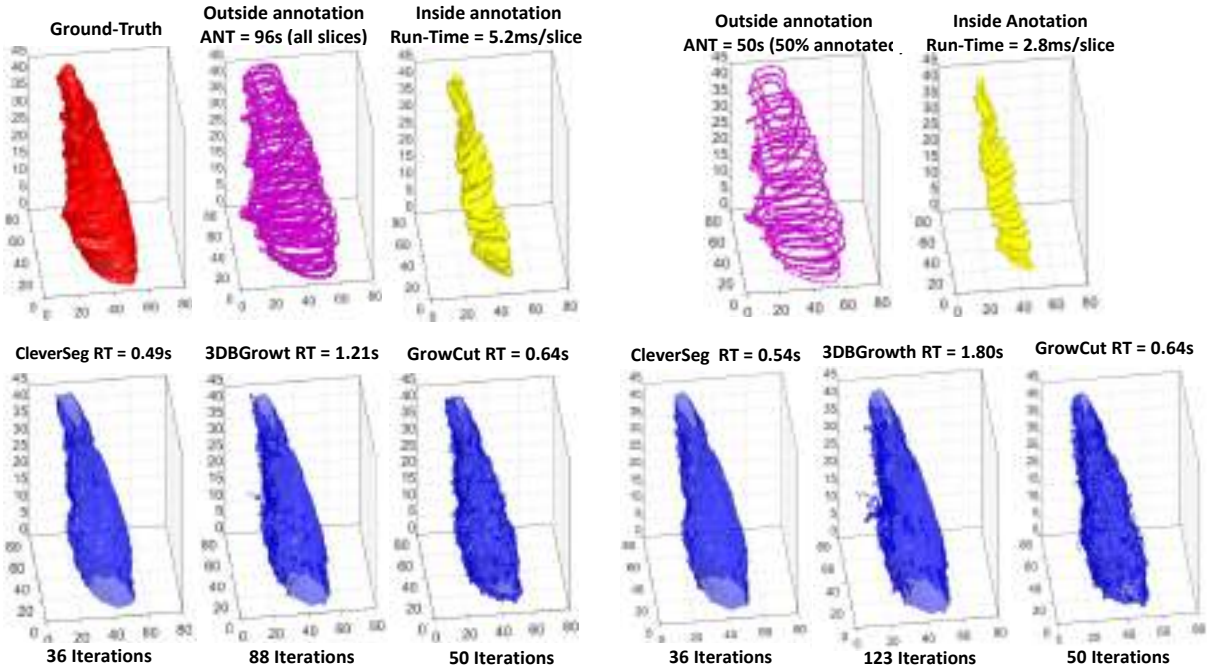


Source: Adapted from Ramos *et al.* (2019b).

In general, 3DBGrowth requires more iterations to converge as the number of slices increases, consequently, increasing the processing time. On the other hand, CleverSeg presented results comparable or better than the competitors, while managing a faster Run-Time. To better illustrate this, Figure 29a shows the segmentation results for a single muscle. Note that, CleverSeg presented the fastest Run-Time and a comparable or better F-Measure than the competitors. For this example, the interior annotation took 96s (ANT) while the estimation of the interior annotation was considerable faster (5.2ms). In average, the interior estimation for all muscles took 4.5 ± 3.6 ms. If the inside annotation was to be manually given, perhaps the time spent on manual annotation (ANT) would be doubled.

Analyzing the segmentation results in Figure 29b, GrowCut presented spiculated borders with a lower F-Measure, while CleverSeg and 3DBGrowth presented smooth borders and equal F-Measure. However, CleverSeg had the lowest number of iterations, and presented the fastest Run-Time. To further validate CleverSeg and, at the same time, reduce the annotation time (ANT), on the next experiment we vary the number of slices with exterior annotation for each muscle.

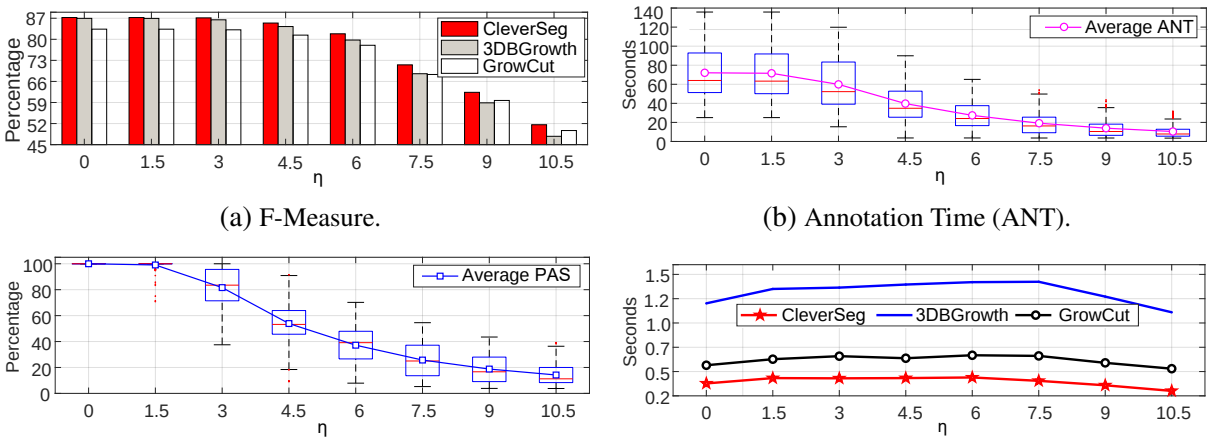
Figure 29 – Segmentation results for a single muscle: exam 52, psoas left.



(a) All slices annotated ($PAS = 100\%$). (b) $\eta = 4.5$ ($PAS = 50\%$).

Source: Adapted from Ramos *et al.* (2019b).

Figure 30 – Results for varying the mean absolute error (η) threshold.



(a) F-Measure. (b) Annotation Time (ANT).

(c) Percentage of Annotated Slices (PAS). (d) Run-Time.

Source: Adapted from Ramos *et al.* (2019b).

5.3.2 Varying the number of annotated slices

For this experiment, we use the error η between slices as a threshold in order to find which slices should be annotated. We summarize the average results for all muscles in Figure 30. The F-Measure starts dropping at $\eta = 4.5$ in Figure 30a, while ANT rapidly drops at $\eta = 1.5$ in Figure 30b along with the *PAS* slices in Figure 30c. For the Run-Time in Figure 30d, CleverSeg was the fastest method for all thresholds.

Table 9 – Wilcoxon's test results at the 5% significance level: ✔ means that CleverSeg was significantly better and ✗ means that no significant difference was observed.

		η							
CleverSeg against	Measure	0.0	1.5	3.0	4.5	6.0	7.5	9.0	10.5
3DBGrowth	F-Measure	✗	✗	✗	✔	✔	✔	✔	✔
	Jaccard	✗	✗	✗	✔	✔	✔	✔	✔
	Average Hausdorff	✗	✗	✔	✔	✔	✔	✔	✔
	Run-Time	✔	✔	✔	✔	✔	✔	✔	✔
GrowCut	F-Measure	✔	✔	✔	✔	✔	✔	✔	✔
	Jaccard	✔	✔	✔	✔	✔	✔	✔	✔
	Average Hausdorff	✗	✗	✗	✗	✗	✗	✗	✗
	Run-Time	✔	✔	✔	✔	✔	✔	✔	✔

Source: Adapted from Ramos *et al.* (2019b).

According to the results reported in Figure 30, $\eta = 4.5$ presented the best trade-off between F-Measure and ANT. For this threshold, the *PAS* drops to approximately 50% and the ANT drops from 72 to 40 seconds (almost 2x faster), losing just a tiny bit of F-Measure (from 87%, 87% and 83% to 85%, 84% and 81% for CleverSeg, 3DBGrowth and GrowCut, respectively). Figure 29b illustrates this, in which, compared to Figure 29a, ANT drops almost to a half, while F-Measure drops 1% for GrowCut. To further validate the results presented herein, in the next section we perform statistical testing.

5.3.3 Statistical evaluation and validation

As the data for all measures presented several similar values, the Kolmogorov-Smirnov (MASSEY, 1951) test was applied at the 5% significance level. The null hypothesis was rejected for all measures, which indicates the data do not follow a normal distribution. Therefore, the Wilcoxon (WILCOXON; KATTI; WILCOX, 1970) test was employed at the 5% significance level and Table 9 reports the results. Note that, CleverSeg presented significantly better running time (Run-Time) than 3DBGrowth and GrowCut. Compared to 3DBGrowth, CleverSeg presented better results for F-Measure, Jaccard and Average Hausdorff from $\eta = 4.5$ to 10.5. In general, CleverSeg presented comparable or significantly better results than the competitors, while always achieving a faster processing time. For η below 3.0, there was no significant difference for most measures due to the high number of slices that are annotate considering this threshold, making both methods comparable results. For the Average Hausdorff, there was no significant difference between CleverSeg and GrowCut. This happened because the Average Hausdorff will present an average between the segmentation distances, therefore not counting the fact that CleverSeg presented several spiculated border.

5.4 Final considerations

The semi-automatic segmentation of muscles in larger volumetric MRI exams is a challenging task. In general, annotating every single slice (inside and outside of the object of interest) in the exam takes too much time. Therefore, allowing a fast and accurate segmentation of slices is crucial in order to obtain a proper 3D reconstruction of the muscle. To ease the manual annotation process, we used the mean absolute error to annotate only the most relevant slices. We also estimated the inside annotation based on the outside annotation, avoiding manual inside annotations. The experimental results showed that only 50% of the slices required outside annotations (on average). Besides, the time spent on overall annotations is 50% faster by using only the outside annotation and quickly estimating the interior annotation with our approach. As a consequence, the time spent on manual annotation is greatly minimized, allowing a more productive and less tiresome activity. We highlight that CleverSeg presented better or similar results than 3DBGrowth and GrowCut while managing a statistically significant lower Run-Time.

Concerning the objectives of this Doctorate research, in this chapter, we consider our contribution as a **second step** in the **Research Problem 2:** Reducing manual interaction on the segmentation of lumbar vertebral bodies in MRI exams. In the next chapters, we explore the next step towards addressing **Research Problem 2**.

CHAPTER

6

FAST CLEVER SEGMENTATION – FASTCLEVERSEG

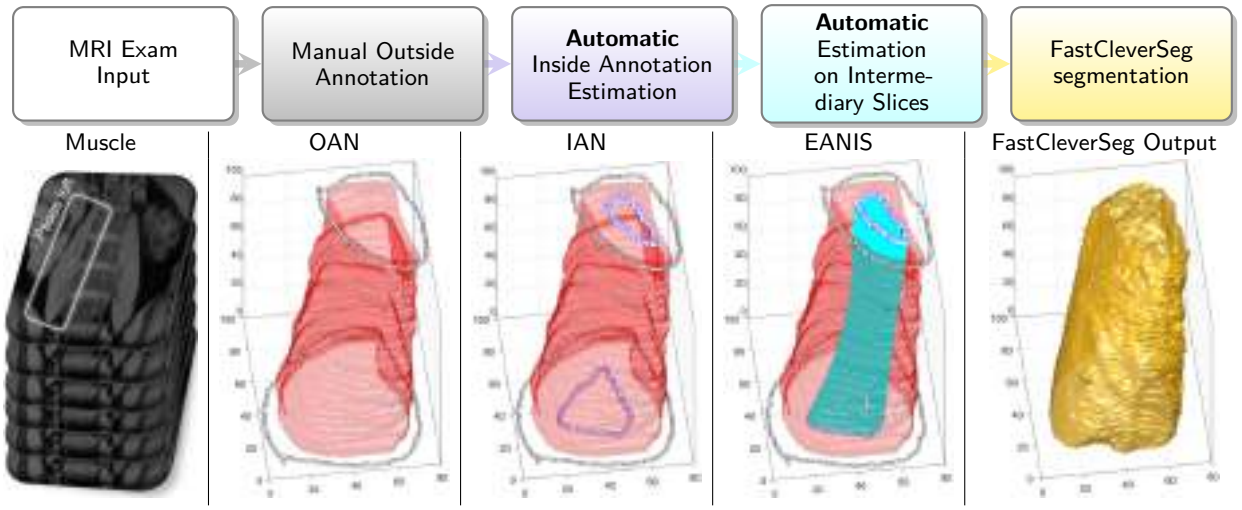
In this Chapter, we present a fast and accurate approach for volumetric segmentation of spine anatomical structures. First, in Section 6.1, we present our proposed approach in two main parts: (1) Estimating ANnotation on Intermediary Slices (EANIS) by requiring the annotation of only two/three slices (3% of the slices) in large MRI exams (about 100 slices), EANIS automatically estimates the annotation of intermediary slices. Thus, the manual annotation speeds up while assisting the segmentation techniques to yield more accurate results; (2) Fast Clever Segmentation (FastCleverSeg) a fast and accurate semiautomatic segmentation approach, which uses a clever balancing of the weights during the expansion of the labels (manual annotation), so that each voxel is visited only once if its label does not change. Next, in Section 6.2 we detail the materials and methods used to validate our approach. After that, in Section 6.3, the experimental results show that, in average, EANIS performed in 6 ± 3 milliseconds per structure, while the manual annotation process took only 8.1 ± 2.2 seconds. Moreover, FastCleverSeg presented an average F-Measure of 94% with the fastest Run-Time: 25 ± 30 milliseconds per structure. Finally, in Section 6.4 we conclude this chapter.

6.1 FastCleverSeg: The Proposed Method

Our approach focus on minimizing the specialist's effort to segment and reconstruct MRI exams built on 2D slices. To this end, we contribute in these two main aspects, as the pipeline shows in Figure 31, and we detail as follows.

Annotation of only two to three slices (Section 6.1.1). Given the Outside ANnotation (OAN) of the slices, we use the approach proposed in (RAMOS *et al.*, 2019b), which uses mathematical morphology to estimate the Inside ANnotation (IAN) based on given OAN of each annotated slice, as we explored in Section 5.1.2. With the estimated IAN, we

Figure 31 – Proposed segmentation pipeline for spine anatomical structures on MRI: Illustration for paravertebral muscle. In red: ground-truth, in black: exterior annotation, in cyan: estimated interior annotation on intermediary slices (EANIS), and in yellow: fCleverSeg segmentation result.



Source: Adapted from Ramos *et al.* (2021).

Estimate the ANnotation on Intermediary Slices (EANIS) by taking advantage of the quick Bresenham's method.

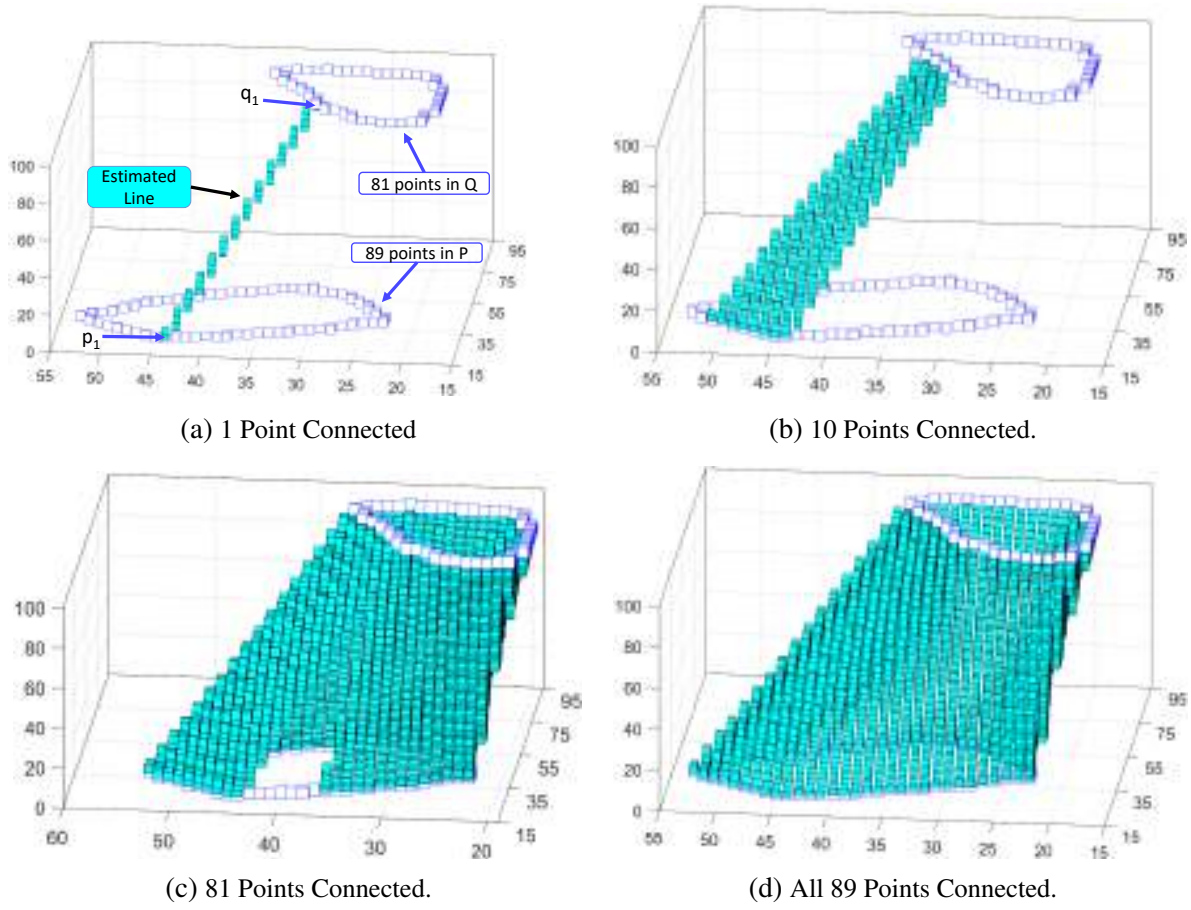
Fast and effective segmentation (Section 6.1.2). We propose the FastCleverSeg method, which presents a faster Run-Time (RT), keeping better or comparable accuracy than the best semi-automatic competitors. FastCleverSeg codes are privately available[1]. However, in the final version of this manuscript, FastCleverSeg codes will be openly available.

6.1.1 Annotation on intermediary slices – EANIS

First, let us define a set of points $P = \{p_1, p_2, ..., p_n\}$ representing the points of the contour of the inside annotation (IAN) in the bottom slice, in which $p_i = \{x_i, y_i, z_i\}$ represents a single point in a 3-dimensional space and n is the number of points. Likewise, we define a set of points $Q = \{q_1, q_2, ..., q_n\}$ for the contour of the IAN on the top slice. Initially, the sets of points P and Q are sorted in ascending way, i.e. points near to the origin $\{x = 0, y = 0\}$ are put at the beginning of each list. We trace a straight line between each pair of points in P and Q, $\{p_1, q_1\}$, as illustrated in Figure 32a. We use Bresenham (1965) line tracing algorithm to determine an approximately straight line between each pair of points. Briefly, the Bresenham's approach is an incremental algorithm that uses only integer addition, subtraction and bit shifting, which are computationally faster to process than traditional computation demanding multiplication and division of values.

[1] <https://drive.google.com/drive/folders/1FlLBQkSYuc6jb29pHUEMgKA2t859uY8O?usp=sharing>

Figure 32 – Bresenham's line tracing algorithm: Example of iterations in the process of estimating the inside annotation on intermediary slices (EANIS).



(a) 1 Point Connected

(b) 10 Points Connected.

(c) 81 Points Connected.

(d) All 89 Points Connected.

Source: Adapted from Ramos *et al.* (2021).

In general, there is a difference in the number of points in P and Q, as exemplified in Figure 32, in which $|P| = 89$ and $|Q| = 81$. To cope with this, we consider each set of points as a circular list. When reaching the last point in the smaller list (q_{81} in the example), the next points to be connected are the first one (on the smaller list, q_1) and the i^{th} point on (the larger list, p_{82}). The process stops when all points on the larger list are connected, e.g. when reaching point p_{89}.

For the external annotation on the intermediary slices, we considered a straight line on the borders of the cropped ROI, which composes a rectangle-like outside annotation at the endmost part of the ROI.

6.1.2 Fast Clever Segmentation – FastCleverSeg

In our proposed FastCleverSeg method, for simplicity's sake, we consider gray-scale volumes and a binary segmentation scenario, i.e. foreground and background segmentation. However, our method can be trivially adapted to multi-channel images or different gray-scales as

well as to segment several regions.

As shown in Algorithm 4, we divide our approach in two steps: initialization and core processing. In the **initialization**, we consider M,N, and Z as the number of rows, columns, and slices, respectively, for the four following matrices:

I: volumetric MRI exam in gray-scale with voxels intensities in $[1,2,...,256]$.

L: labels matrix comprising manual annotations, in which the entries with values 0, 1 and 2 are unlabeled voxels, background/outside annotation and foreground/inside annotation, respectively.

W: weighted matrix, in which, initially, every voxel $\forall\ (x,y,z) \in W$ (line 1) is filled with ones for corresponding labeled entries in L and zeros otherwise (lines 2 and 3):

$$W(i,j,z) = \begin{cases} 0.0 & \text{if} \quad L(i,j,z) = 0, \\ 1.0 & \text{if} \quad L(i,j,z) \neq 0. \end{cases} \tag{6.1}$$

B: boolean matrix that keeps track of the visited status of each voxel, in which **true** represents the visited status and **false** represents the not visited status. Initially, every single voxel is set to not visited or **false** (line 4).

Given matrices I,L,W, and B, FastCleverSeg may proceed to its **core process**, which is detailed in three steps as follows.

Step 1. For every voxel $\forall(i,j,z)$, check if the corresponding entry in the label matrix is annotated, $L(i,j,z) \neq 0$, and was not visited/processed yet, $B(i,j,z) =$ **false**, line 5. If the voxel does not meet the condition, proceed to the next voxel. Otherwise, mark the current voxel (i,j,z) as visited, $B(i,j,z) =$ **true**, (line 6) and go to the next step.

Step 3 For each one of the 26 neighbors $(\{i_n,j_n,z_n\}, n \in [1,2,...,26])$ of the current voxel $\{i,j,z\}$ (line 7), calculate the absolute difference between the current voxel intensity I(i,j,z) and the neighbor's voxel intensity $I(i_n,j_n,z_n)$ normalized by the maximum voxel intensity of the volume and then subtracted by one (line 8):

$$nvd = 1 - \frac{|I(i,j,z) - I(i_n,j_n,z_n)|}{\max\limits_{\forall(i,j,z)\in V} I(i,j,z)} \ .$$

Then, a strength s is computed by multiplying the weight of the current voxel $W(i,j,z)$ with the *normalized voxel difference* (nvd, line 9):

$$s = W(i,j,z) \times nvd \ .$$

Algorithm 4: FastCleverSeg method. Source codes available in Section A.3.

Input: Volumetric exam I and annotation/labels matrix L.
Output: Segmented image (grown regions in L).

```
// Initialization
```
1 **for** $\forall(i,j,z)$ **do**
2 **if** $L(i,j,z) \neq 0$ **then**
3 $W(i,j,z) \leftarrow 1.0$
4 $B(i,j,z) \leftarrow$ **false**

```
// Core process
```
5 **for** $\forall(i,j,z) \in I$ **and** $L(i,j,z) \neq 0$ **and** $B(i,j,z) =$ **false do** `// Step 1: For each labeled voxel not visited`
6 $B(x,y,z) \leftarrow$ **true**
7 **for** $\forall(i_n,j_n,z_n)$ **do** `// Step 2: For each one of the 26-Neighbors` (i_n,j_n,z_n)
8 $nvd \leftarrow 1 - \frac{|I(i,j,z) - I(i_n,j_n,z_n)|}{\max\limits_{\forall(i,j,z)\in I} I(i,j,z)}$
9 $s \leftarrow W(i,j,z) \times nvd$
10 $df \leftarrow s - W(i_n,j_n,z_n)$
11 **if** $df > \theta$ **then** `// Step 3: manage averaged values`
12 $W(i_n,j_n,z_n) \leftarrow [\, W(i,j,z) + W(i_n,j_n,z_n) + s \,]/3$
13 $L(i_n,j_n,z_n) \leftarrow L(i,j,z)$
14 $B(i_n,j_n,z_n) \leftarrow$ **false**

After that, calculate the difference between s and the neighbor's current strength $W(i_n,j_n,z_n)$ as df (line 10):

$$df = s - W(i_n,j_n,z_n) \ .$$

Step 3. If df is greater than a threshold θ (line 11), then, **first**, the current voxel's strength $W(i,j,z)$, the neighbor's current strength $W(i_n,j_n,z_n)$ and the new strength s are averaged and set as the new strength of the neighbor's voxel (line 12):

$$W(i_n,j_n,z_n) = \frac{s + W(i,j,z) + W(i_n,j_n,z_n)}{3} \ .$$

Thus, we introduce a more accurate balancing between the weights. **Secondly**, update the neighbor's label $L(i_n,\ j_n,\ z_n)$ with the current voxel's label $L(i,j,z)$ (line 13):

$$L(i_n,j_n,z_n) = L(i,j,z) \ .$$

Finally, the neighbor's voxel visiting status is set to not visited (line 14), $B(i_n,j_n,z_n) =$ **false**, ensuring that the neighbor voxel will be visited at least one more time, whether it was visited or not.

This process (line 5) continues until there is no label change for all voxels.

The θ threshold avoids balancing (averaging) values that only change the third or fourth decimal place. We explore the θ threshold Section 6.3.5. The visiting status ensures that each voxel is visited only once, unless its label changes. Therefore, the processing time can be significantly improved, since each voxel is visited just a few times, depending on whether its strength changed or not. At the same time, we efficiently propagate the strengths as the region grows by using a more accurate balancing between the weights.

6.2 Materials and methods

In this section, we present the materials and methods used in the experimental design. First, in Section 6.2.1, we describe the volumetric databases used in the experiments. We considered five distinct databases divided into three groups: paravertebral muscles, intervertebral discs, and vertebral bodies. Then, in Section 6.2.2, we detail the parameters settings along with the computational set-up as well as how the manual annotations were performed. Next, in Section 6.2.3, we report the competitors considered in the analysis. Finally, in Section 6.2.4, we detail the measures considered to validate the experimental results.

6.2.1 Patient database

We have used five MRI databases, with respective reference manual segmentations, which we divided into three groups: paravertebral muscles, intervertebral discs, and vertebral bodies. First, we consider a public database of paravertebral muscles (BURIAN *et al.*, 2019), which we detailed in Section 5.2.1. Then, we present an intervertebral disc database Menezes-Reis *et al.* (2016). Finally, we assembled three databases of vertebral bodies, which are publicly available (ZUKIĆ *et al.*, 2014; BURIAN *et al.*, 2019; CHU *et al.*, 2015b; CHU *et al.*, 2015a) Table 10 reports the overall details for the aforementioned databases. The ground-truth segmentations were performed or supervised by a physician.

6.2.2 Parameters settings

In order to evaluate our proposed method for the segmentation task, we compared it with CleverSeg (RAMOS *et al.*, 2019b) 3DBGrowth,(RAMOS *et al.*, 2019a) Fast GrowCut (ZHU *et al.*, 2014) and GrowCut (VEZHNEVETS; KONOUCHINE, 2005). For every method, no maximum number of iterations was set, which implicates that every algorithm runs until convergence. We compare the results with and without our proposed estimated annotation on intermediary slices (EANIS) as well.

For FastCleverSeg and CleverSeg, we set the threshold θ to 1%. The remaining parameters of all methods are set to default settings to achieve the best results of each method and to avoid loss of generalizability. No pre or post-processing techniques were applied to assure the same conditions for all segmentation methods.

Table 10 – Databases scan parameters (prior to the normalization).

Database	MRI Sequences.	TR/TE (s)	Resolution (Voxel)	Resolution (mm^3)
Paravertebral Muscles Burian *et al.* (2019)	T1 TFE	6.4/1.1	334×334×67 ±135×135×5.0	1.0×1.0×3.6 ±0×0×5
Intervertebral Discs Menezes-Reis *et al.* (2016)	T2 SE	3.9/1.6	512×512×16 ±0×0×0	0.4×0.4×4.4 ±0×0×0
Vertebral Bodies Zukić *et al.* (2014)	T1 F/TSE, T2 T/SE, T2 fr/FSE, TIRM	–/–	542×542×18 ±106×106×5	0.7×0.7×3.8 ±0.2×0.2×0.3
Vertebral Bodies Burian *et al.* (2019)	T1 FFE	11/1.4	222×222×23 ±7×7×12	1.0×1.0×3.9 ±0×0 ×0.2
Vertebral Bodies Chu *et al.* (2015b)	T2 TSE	5.2/0.1	305×305×39 ±0×0×0	1.2×1.2×2.0 ±0×0×0

Source: Adapted from Ramos *et al.* (2021).

Note – TE: Echo Time, TR: Repetition Time, SE: Spin-echo, TSE: Turbo Spin-echo, –: Information not reported in the paper neither present on the files's metadata.

To assure the best conditions to all segmentation algorithms, since the majority of the original exams have 65.536 gray levels (16 bits/pixel), we normalized the gray-scale of the exams into 256 intensities (8 bits/pixel) using Equation 2.2. All MRIs were normalized to the isotropic spatial resolution ($1 \times 1 \times 1$ mm^3), employing a trilinear interpolation.

6.2.2.1 Computational set-up

Every experiment used a 2.40GHz Intel(R) Core(TM) i7 CPU and 8GB RAM machine, using Matlab(R) version 2018a.

6.2.2.2 Manual annotations

For every segmentation task, two or three representative slices present Outside Annotation (OAN) as follows:

Two OAN. For the paravertebral muscle database, we have considered only the manual outside annotation of the top and bottom slices due to the approximately regular shape of the muscle (not concave or convex in the plane perpendicular to the annotations plane).

Three OAN. For the intervertebral discs and vertebral bodies, we have considered only the top, middle, and bottom slices due to their non-regular shape (presents concavity or convexity).

For each slice with OAN, the Inside Annotation (IAN) was automatically calculated based on the OAN. Therefore, the number of annotated slices required relies on the shape of

the ROI. In our case, only two or three annotated slices were required, as aforementioned. The annotations on intermediary slices were estimated using EANIS (Section 6.1.1).

6.2.3 Competitors

In order to evaluate our proposed method for the segmentation task, we compared it with CleverSeg (RAMOS *et al.*, 2019b), 3DBGrowth (RAMOS *et al.*, 2019a), FastGrowCut (ZHU *et al.*, 2014), and GrowCut (VEZHNEVETS; KONOUCHINE, 2005), which are state-of-the-art methods for semi-supervised segmentation.

6.2.4 Comparison measures

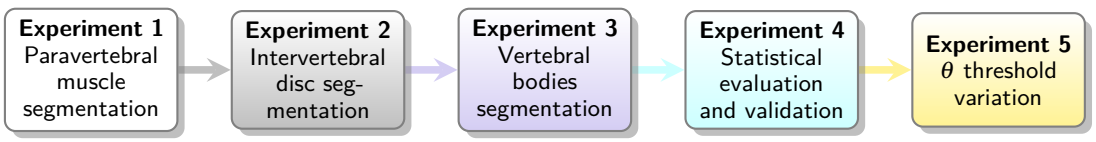
In order to compare the resulting segmentation yielded by each segmentation method, we considered the F-Measure (Equation 2.11), Average Hausdorff (Equation 2.14), AUC (Equation 2.13), and Run-Time.

6.3 Experiments, results and discussion

For all experiments, the results with EANIS implicates that only the outside manual annotation on the most representative slices were given and the inside annotation was automatically estimated on those slices. The results with EANIS implicates that the inside and outside annotation were estimated using our approach on the intermediary slices.

In our experimental design, we analyze four main parts, depicted in Figure 33, and detailed as follows. Next, we segment only intervertebral discs. First, we analyze the segmentation

Figure 33 – Experimental design to validate FastCleverSeg.



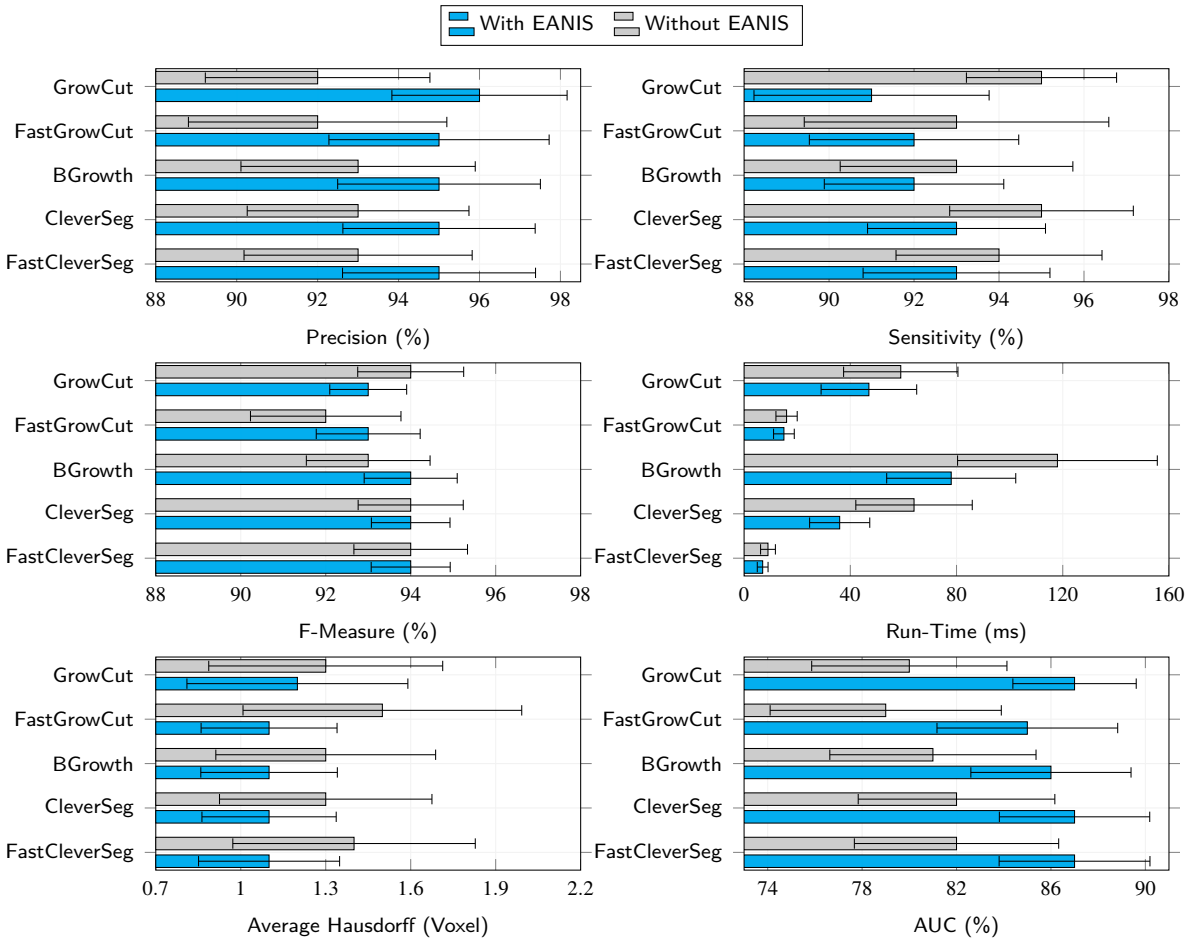
Source: Elaborated by the author.

of every paravertebral muscle. After that, we analyze the segmentation of every vertebral body. Then, we analyze the overall results by joining the results of the three parts aforementioned. We also analyze the data distribution of the results and perform validations using statistical testing. Finally, we evaluate the effect of changing the θ threshold on FastCleverSeg and CleverSeg methods.

6.3.1 Paravertebral muscle

Figure 34 shows the results (average values) for the paravertebral muscles database. Note that, in general, EANIS improved the results for all segmentation methods considering every measure. The Sensitivity was slightly lower with EANIS for GrowCut. On the other hand, the Precision and AUC are higher with EANIS for GrowCut. For the Average Hausdorff, AUC, and F-Measure, FastCleverSeg and CleverSeg presented very close results (smallest value, which is better). Analyzing the Run-Time, FastCleverSeg presented the fastest Run-Time while BGrowth was the slowest one.

Figure 34 – MRI database segmentation results for the paravertebral muscles with and without EANIS.

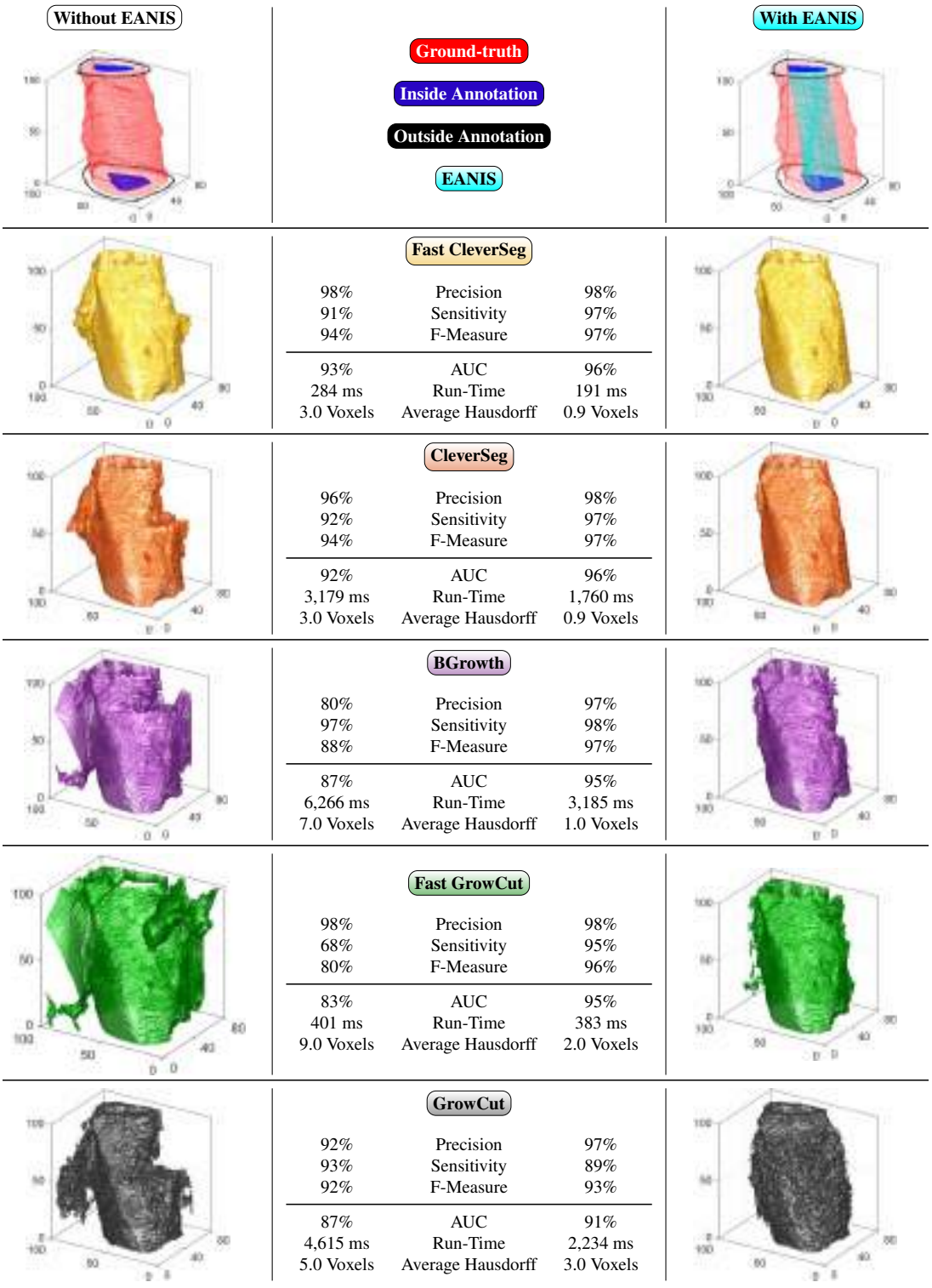


Source: Adapted from Ramos *et al.* (2021).

To further illustrate the results, Figure 35 shows the segmentation for a single muscle with and without EANIS. In general, all methods over-segments (lower Precision) the muscle without EANIS due to the absence of annotations on the intermediary slices. On the other hand, with EANIS (in cyan), all methods produce a segmentation closer to the ground-truth with a lower Average Hausdorff and Run-Time while presenting a higher F-Measure, AUC, Precision, and Sensitivity. With and without EANIS, FastCleverSeg and CleverSeg presented the best results. FastGrowCut and GrowCut presented spiculated borders even with EANIS, while the

remaining methods presented smoother borders.

Figure 35 – Segmentation results for a single paravertebral muscle with and without EANIS.

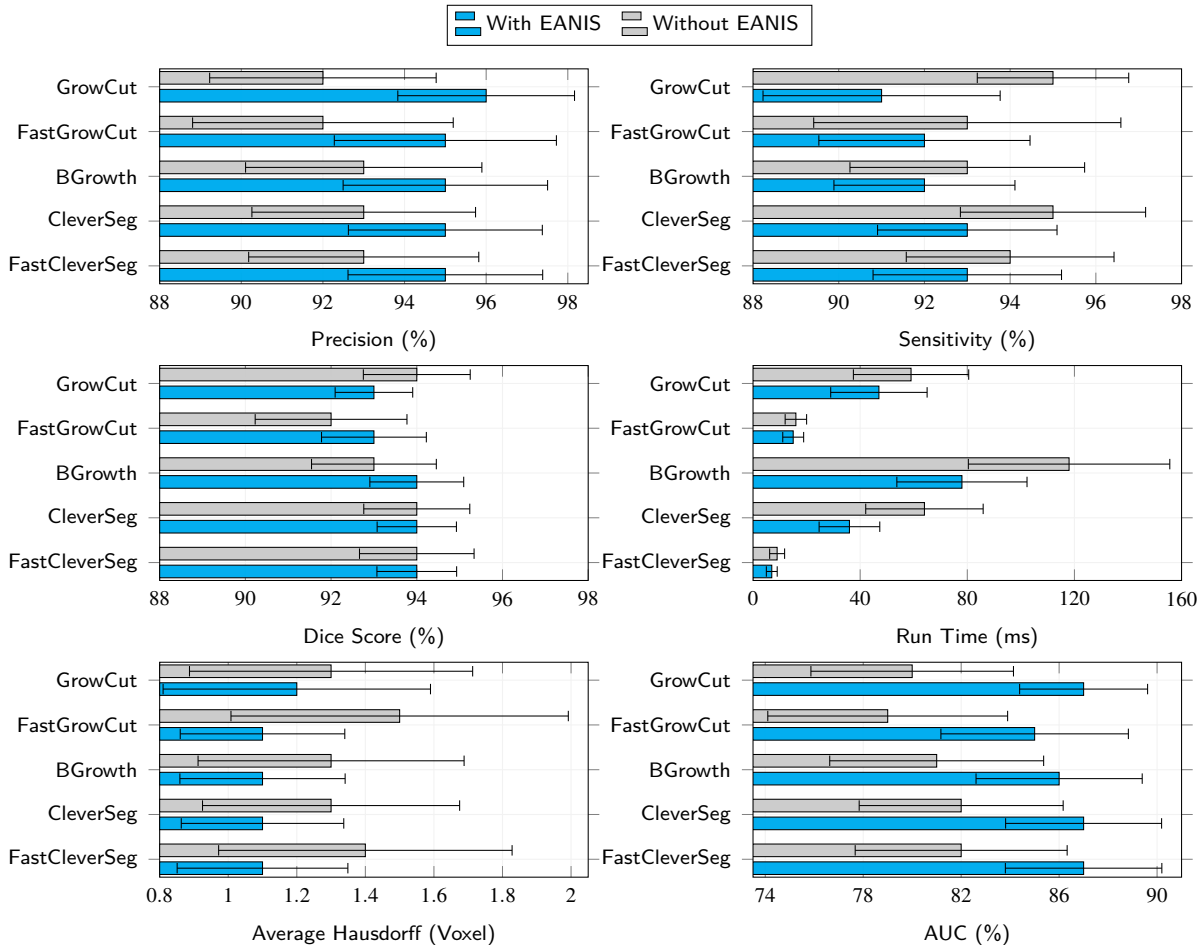


Source: Adapted from Ramos *et al.* (2021).

6.3.2 Intervertebral disc

Figure 36 summarizes the average segmentation results with and without EANIS for the intevertebral discs database. Note that, every segmentation method presented a higher Sensitivity without EANIS. However, the Precision and AUC were higher with EANIS, while the F-Measure was similar for most of the methods. Additionally, note that with a lower error margin one can see that the results with EANIS were more concise. This indicates that, without EANIS, most methods under-segments (lower Precision) the intervertebral discs. Analyzing the Run-Time, in general, all methods performed faster with EANIS.

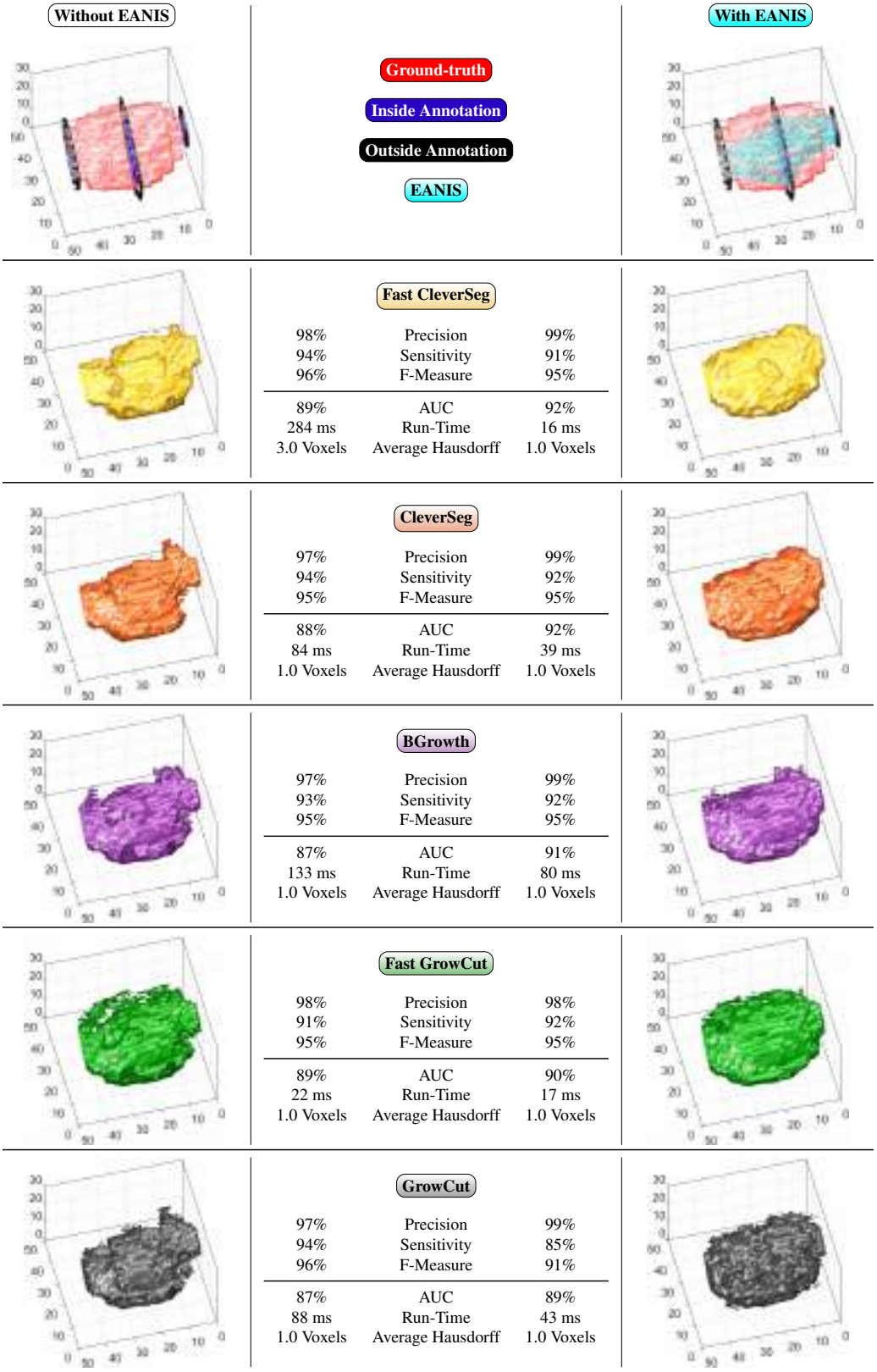
Figure 36 – Segmentation results for the intervertebral discs MRI database with and without EANIS.



Source: Adapted from Ramos *et al.* (2021).

To better illustrate the results, Figure 37 shows the segmentation for a single intervertebral disc. With EANIS, for every segmentation method, the Precision was higher and, therefore, the results were closer to the ground-truth (even visually) with a higher AUC. As a last remark, every segmentation method had a faster Run-Time with EANIS: FastCleverSeg was the fastest one while BGrowth was the slowest one.

Figure 37 – Segmentation results for a single intervertebral disc with and without EANIS.

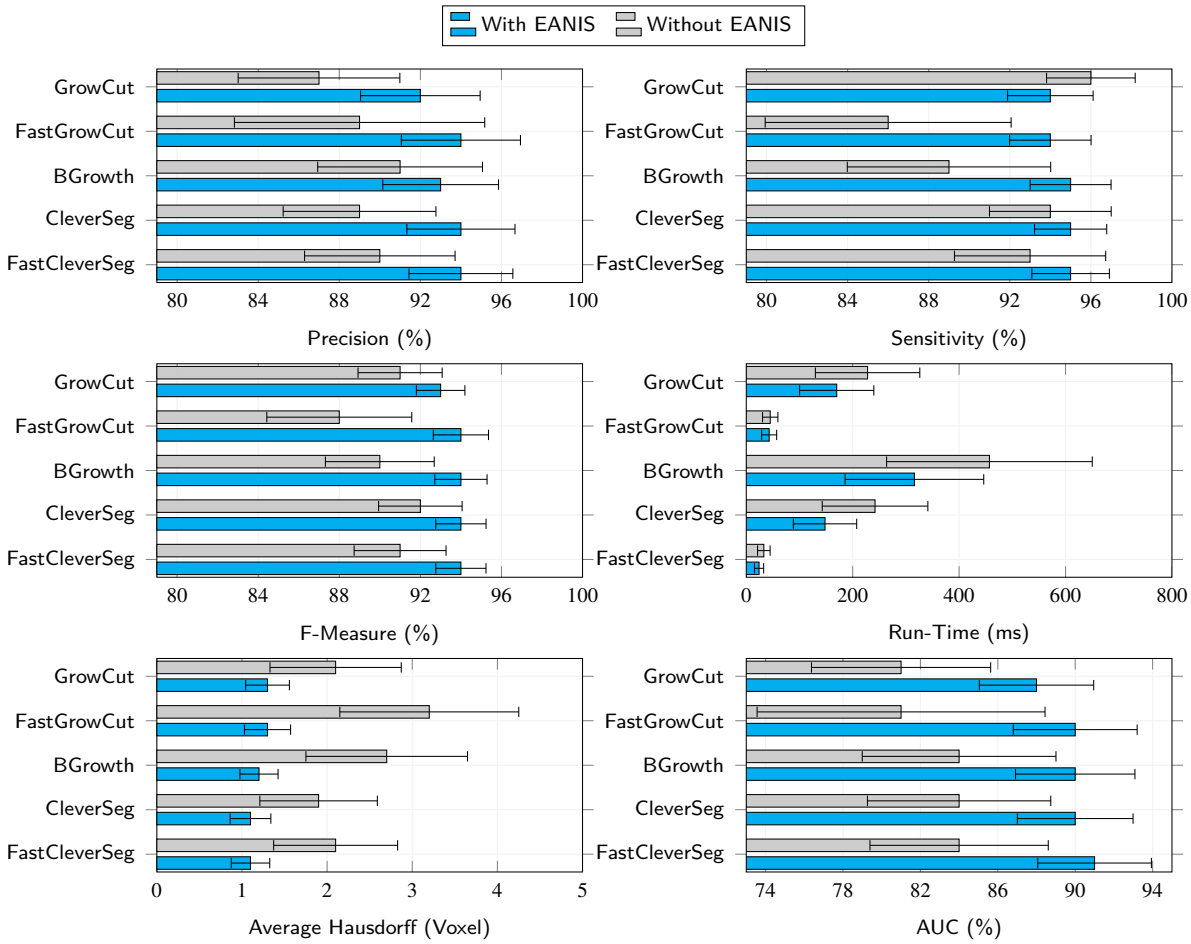


Source: Adapted from Ramos *et al.* (2021).

6.3.3 Vertebral body

The vertebral bodies database is made up of three vertebral bodies databases, as we described in Section 6.2.1. Figure 38 reports the average segmentation results for the assembled vertebral bodies database. Note that, for most measures, EANIS improved the results for every method and, at the same time, presented a lower standard deviation, which implicates that, with EANIS, the segmentation methods presented more concise results. For the GrowCut method, although there was no Sensitivity improvement, the Precision along with the F-Measure and AUC presented better results with EANIS. To the greater extent, FastCleverSeg and CleverSeg presented close results and FastCleverSeg presented the fastest Run-Time (RT) with and without EANIS.

Figure 38 – Segmentation results for the vertebral body MRI database with and without EANIS.

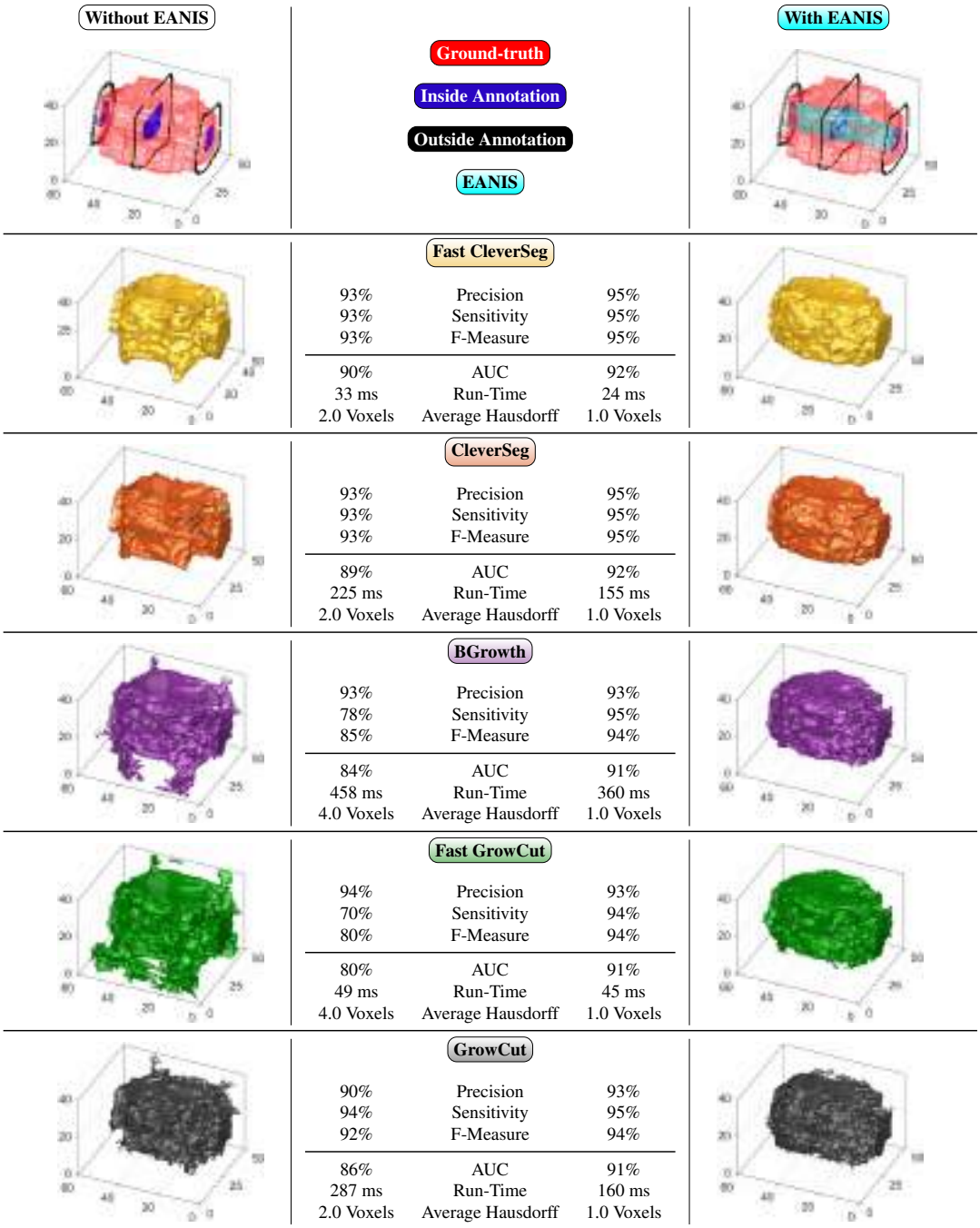


Source: Adapted from Ramos *et al.* (2021).

Figure 39 shows the segmentation results for a single vertebral body. In general, all methods over-segment in a few parts, while under-segment in the others without EANIS. This issue is minimized with EANIS, so that all methods presented a higher Precision and Sensitivity, consequently, a higher F-Measure and AUC. With and without EANIS, FastCleverSeg and

CleverSeg presented the best results with smoother borders. FastCleverSeg had the fastest Run-Time.

Figure 39 – Comparison of segmentation results for a single vertebral body with and without EANIS.



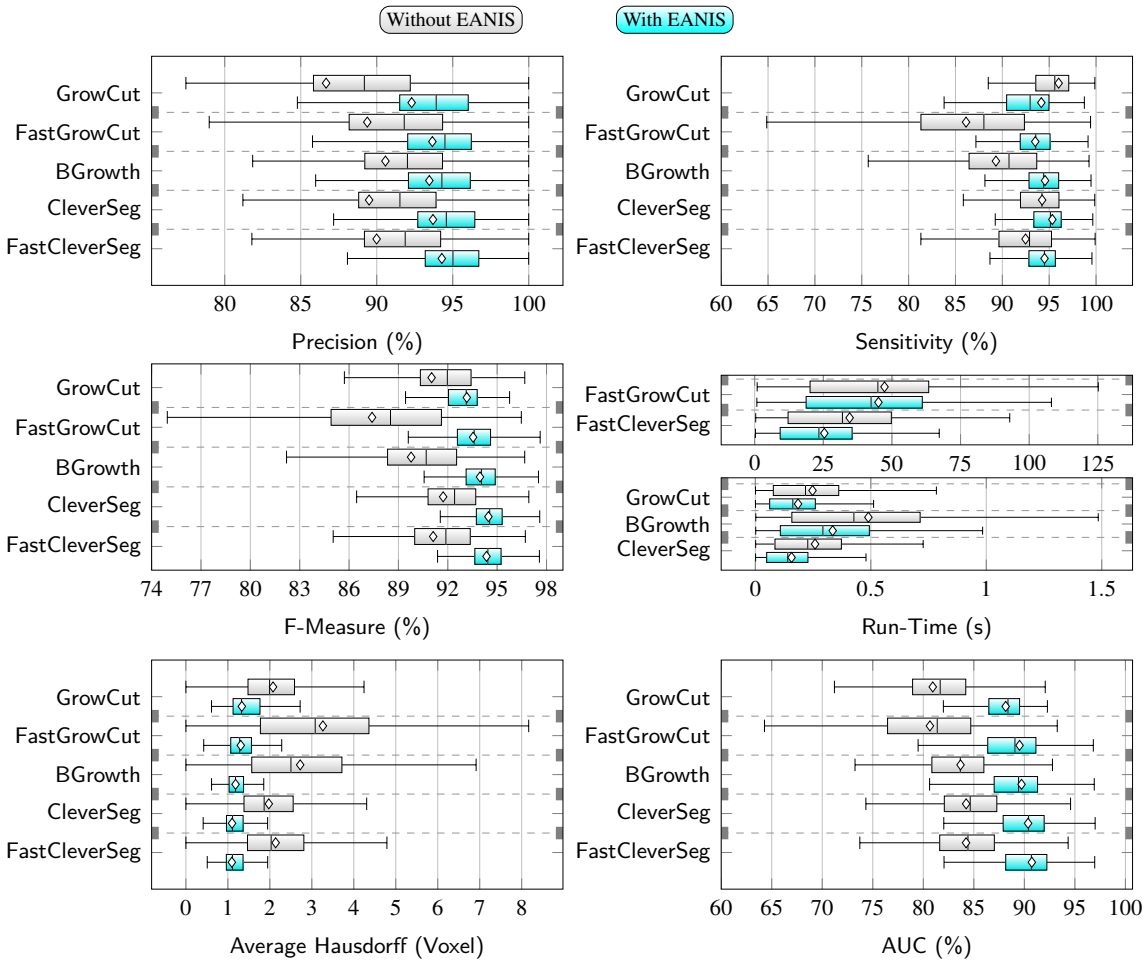
Source: Adapted from Ramos *et al.* (2021).

6.3.4 Statistical evaluation and validation

In order to analyze the segmentation of all databases considered in the analysis, Figure 40 shows the data distribution of all segmentation results. Note that, there were no Sensitivity

improvements with EANIS for GrowCut. However, for all the remaining measures, GrowCut presented better results with EANIS. For the remaining segmentation methods, the results with EANIS were better for every measure. In general, FastCleverSeg and CleverSeg presented the best results for all measures. However, for the Run-Time, FastCleverSeg was the fastest one with and without EANIS, surpassing the fastest competitor (FastGrowCut).

Figure 40 – Overall data distribution for all MRI databases considered in this work. Comparison with and without EANIS.



Source: Adapted from Ramos *et al.* (2021).

To further validate the results, we conducted statistical testing over the data presented in Figure 40, as we discuss next.

6.3.4.1 Statistical testing

First, we tested the normality of the data by applying the Kolmogorov-Smirnov test at the 1% significance level. The null hypothesis was rejected for all cases and all measures, which indicates the data do not follow a normal distribution. In this case, the Wilcoxon test is the most indicated for the significance analysis.

We employed the Wilcoxon test at the 1% significance level, and we assess the results in two main parts. First, in Table 11, we analyze if there is a significant difference with EANIS against without EANIS for all methods considering every measure.

As reported in Table 11, with EANIS, all methods yielded significantly better results on every measure. Then, in Table 12, we verify if there is significant difference for FastCleverSeg

Table 11 – Wilcoxon test p-value results (at the 1% significance level) for each method with EANIS against itself without EANIS: ✔means that a significant difference is observed.

	Measure					
Method	Hausdorff	AUC	Run-Time	F-Measure	Precision	Sensitivity
FastCleverSeg	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)
FastGrowCut	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)
CleverSeg	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)
BGrowth	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)
GrowCut	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)

Source: Adapted from Ramos *et al.* (2021).

against the competitors' methods with and without EANIS for all methods considering every measure as well. As Table 12 reports, considering no EANIS, FastCleverSeg presented similar results to:

- FastGrowCut: for the AUC and Precision measures. However, for Sensitivity and F-Measure it was significantly better.
- CleverSeg: for the Precision. However, the Sensitivity was significantly better and, consequently, the F-Measure was significantly better as well.
- BGrowth: for the Precision, however, the Sensitivity and F-Measure was significantly better.
- GrowCut: for the Average Hausdorff and F-Measure.

Considering the results with EANIS, FastCleverSeg presented similar results with FastGrowCut for the Hausdorff and F-Measure. There were similar results for the CleverSeg method and FastCleverSeg for the Sensitivity. However, the Precision and F-Measure was significantly better for FastCleverSeg. Predominantly, FastCleverSeg with EANIS presented significantly better results than the remaining methods. FastCleverSeg was significantly faster than every method with and without EANIS.

Table 12 – Wilcoxon test p-value results (at the 1% significance level) considering annotations with and without EANIS: ✔ means significantly better results for FastCleverSeg and ✗means that no significant difference is observed.

Measure	Hausdorff		AUC		Run-Time	
FastCleverSeg against	Without EANIS	With EANIS	Without EANIS	With EANIS	Without EANIS	With EANIS
FastGrowCut	0.000 (✔)	0.885 (✗)	0.106 (✗)	0.031 (✔)	0.000 (✔)	0.000 (✔)
CleverSeg	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)
BGrowth	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)
GrowCut	0.088 (✗)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)
Measure	F-Measure		Precision		Sensitivity	
FastCleverSeg against	Without EANIS	With EANIS	Without EANIS	With EANIS	Without EANIS	With EANIS
FastGrowCut	0.000 (✔)	0.068 (✗)	0.082 (✗)	0.002 (✔)	0.000 (✔)	0.000 (✔)
CleverSeg	0.000 (✔)	0.000 (✔)	0.498 (✗)	0.000 (✔)	0.000 (✔)	0.138 (✗)
BGrowth	0.000 (✔)	0.000 (✔)	0.104 (✗)	0.000 (✔)	0.000 (✔)	0.000 (✔)
GrowCut	0.059 (✗)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)	0.000 (✔)

Source: Adapted from Ramos *et al.* (2021).

6.3.5 θ threshold variation

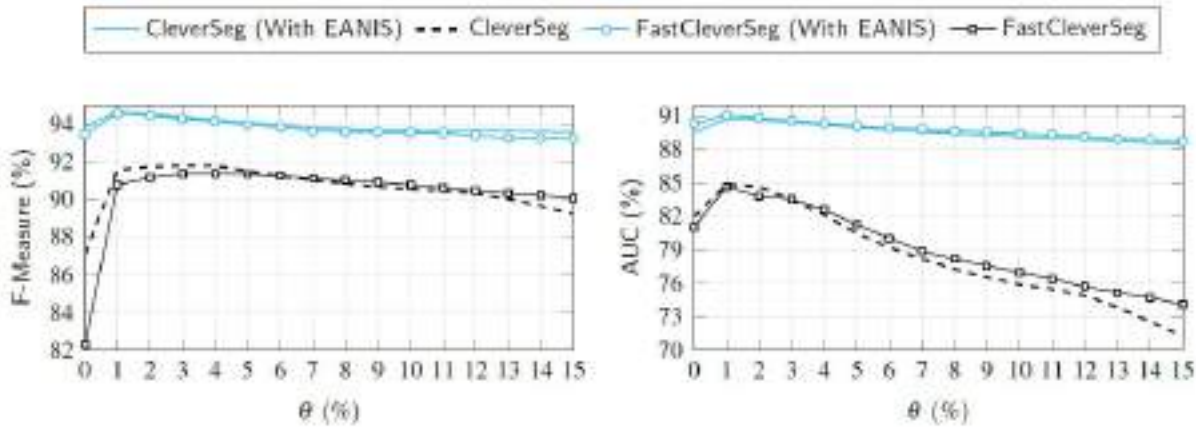
In this experiment, we analyze the effect of varying the θ threshold. We started at 0%, increased by 1%, up until 15%, which sums up to 16 thresholds values. Figure 41 shows the results for the F-Measure and AUC in both scenarios: with and without EANIS.

Note that, from 1% to 15%, the measures decreases as θ increases, and both methods present a decrease in performance for all measures below 1%. Both methods present the best results at 1% with and without EANIS. Conclusively, 1% is the most suitable value for the θ threshold for FastCleverSeg and CleverSeg. For this reason, we employed this threshold value in the BGrowth algorithm in Chapter 3.

6.4 Final considerations

The semi-automatic segmentation of human structures, such as paravertebral muscle, intervertebral discs and vertebral body in large volumetric MRI exams is a challenging task. In general, manually annotating every slice of an exam is a time-consuming and tiresome task for the specialist. On the grounds of that, allowing a fast and accurate segmentation of slices is crucial in order to obtain a proper 3D reconstruction of the segmented structure. To cope with this, we proposed the estimation of annotation on intermediary slices (EANIS) along with the automatic estimation of the inside annotation, so that only two or three representative slices had to be annotated and the whole volumetric exam could be efficiently segmented using our

Figure 41 – Effect of changing the θ threshold: Comparison for several θ threshold values for FastCleverSeg and CleverSeg methods.



Source: Adapted from Ramos *et al.* (2021).

proposed FastCleverSeg method. Furthermore, the inside annotation of each slice is automatically estimated based on the outside annotation, which speeds up the manual step.

Concerning the objectives of this Doctorate research, in this chapter we approached the **Research Problem 2:** Reducing manual interaction on the segmentation of lumbar vertebral bodies in MRI exams. Based on the results presented throughout the chapter, we answer the initial formulated research question as follows:

a) How can we reduce the total time and effort spent on manual annotations of vertebral anatomical structures in MRI exams and maintain segmentation accuracy?

Answer: We firmly believe that FastCleverSeg and EANIS may aid physicians in producing more ground-truths in a faster manner, speeding up the laborious task of manually segmenting regions of interest. By annotating only two or three representative slices, the whole volumetric exam could be efficiently segmented using our proposed FastCleverSeg method. The experimental results showed that our proposed segmentation technique is the fastest one and, at the same time, presented results similar to or better than the competitors, which are the state-of-the-art regarding semi-automatic segmentation methods. Thus, physicians, as well as scientists, may benefit from this, since deep-learning approaches can be applied with even more confidence assuming a larger number of samples.

In the next chapter, we employed our segmentation method into an application scenario in which we cover the texture analysis of vertebral bodies in MRIs. Thus, we step forward into the **Research Problem 3**: Assessing bone fragility fractures using volumetric MRI and radiomic texture features.

CHAPTER

7

ANALYSIS OF BONE FRAGILITY FRACTURES USING TEXTURE FEATURES

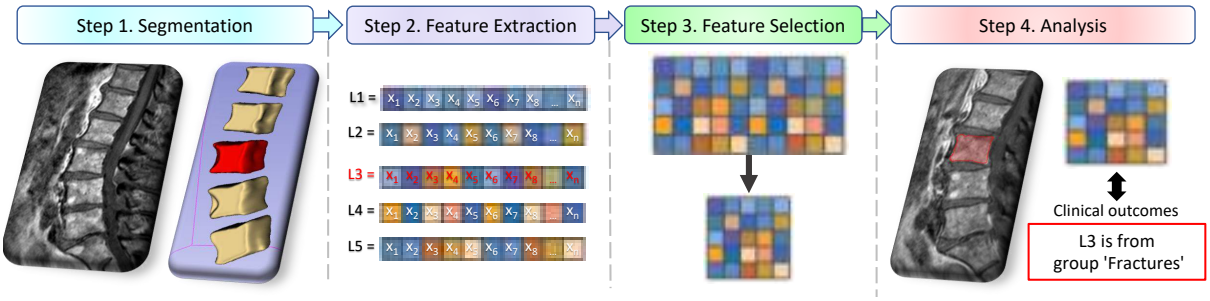
In this Chapter, we use our segmentation method in an application scenario, in which we present an extensive empirical study aimed at assessing fragility fractures secondary to osteoporosis. First, in Section 7.1, we delve into the background and motivation. We developed a radiomic-based approach called BonE Analysis Using Texture (BEAUT) to assess fragility fractures. To evaluate BEAUT, we considered the materials and methods detailed in Section 7.2. Next, in Section 7.3, we present the experimental results, on a meaningful database composed of 47 T2-weighted exams. BEAUT achieved an Accuracy of 92% and 97% AUC with feature selection to discriminate between patients with bone mass loss, spotting those at risk of fragility fractures. Finally, Section 7.4 concludes this Chapter.

7.1 Proposed Method: BonE Analysis Using Texture – BEAUT

To aid in the assessment of lumbar fractures in MRIs, we propose BEAUT, a novel method which takes advantage of the radiomics approach. We divide BEAUT in four steps, as shown in Figure 42 and detailed as follows. First, BEAUT gets a lumbar MRI as input, and segment the vertebral bodies accordingly using CleverSeg. Next, we extract features of texture patterns from each vertebral body. After that, we analyze and select the most relevant features. Finally, we apply a classification model to spot a fragility fracture. For this, BEAUT builds a classification model on previous images to analyze fractures in a specific patient.

BEAUT combines classifiers aiming at achieving the best possible results. We considered two techniques, AdaBoost (Adaptive Boosting) (FREUND; SCHAPIRE, 1996) and LogitBoost (FRIEDMAN; HASTIE; TIBSHIRANI, 2000) to perform classification using all or just selected features, respectively.

Figure 42 – BEAUT pipeline: radiomics analysis for bone mass loss and prediction of fragility fractures secondary to osteoporosis.



Source: Adapted from Ramos *et al.* (2021).

7.2 Materials and methods

In this section, we present the materials and methods used in the experimental design. First, in Section 7.2.1, we describe the volumetric databases used in the experiments. We used a representative and balanced database comprising 47 MRIs exams. Then, in Section 7.2.2, we detail the segmentation and feature extraction. We used CleverSeg to segment the volumetric exams and extracted 1,674 texture features using PyRadiomics. Next, in Section 7.2.3, we explored the analysis of features, in which we selected 434 features using the Chi-square feature ranking. After that, in Section 7.2.4, we detail the classifiers used to compare and validate our approach. We considered representative and well-established classifiers, such as SVM, k-NN, Naïve Bayes, Discriminant Analysis (Disc. Analysis), and Decision Tree. Then, in Section 7.2.5, we detail the parameters settings along with the computational set-up. Finally, Section 7.2.6, we explore the measures considered to validate the experimental results.

7.2.1 Patient Database

The database is composed of 47 DEXA and volumetric spine MRI T2-weighted FSE sequences[1]. Every participants agreed to participate in the study and provided written informed consent. The database was acquired at the Ribeirão Preto Medical School (FMRP-USP[2]) in Ribeirão Preto, SP – Brazil, under the Ethics Research Committee approval (47580315.5.0000.5440) following the Declaration of Helsinki.

All the MRI exams where taken by a 1.5-Tesla system (Philips Achieva, Philips Medical Systems, Best, the Netherlands) with echo time (TE) of 120 ms, 4.4 mm gap, 19 echo-train length, 3900 ms repetition time (TR), 3.0 mm slice thickness, and 2 min 16 s scan duration. The individuals in this study were at least 50 years old and had a Body Mass Index (BMI) between 18 and 25 kg/m^2. None of the patients used medications that affect bone metabolism or

[1] Database acquired in partnership with another doctoral research from M.D. Jamilly G. Maciel under the supervision of Professor M.D. Marcello H. Nogueira-Barbosa (FAPESP Process 2018/04266-9).

[2] <https://www.fmrp.usp.br/en/>

presented reported histories of smoking and/or alcoholism in the last 10 years. Moreover, they did not present any structural abnormality in the lower spine that could affect MRI or DEXA measurements.

As shown in Table 13, the database presents 12 osteopenic patients without fractures and 12 osteopenic with fragility fracture, 12 osteoporotic patients without fractures and 11 osteoporotic with fragility fracture. We organized the database into two categories, 'Fractures' and 'No Fractures', In total, 97 vertebral bodies presented fragility fracture, fitting into the 'Fractures' group. The 'No Fractures' group includes 97 vertebral bodies that do not present fragility fracture. Note that we filtered out vertebral bodies that were fractured and could bias the analysis. We highlight that vertebral bodies from the same patient are not on both groups neither present osteopenia and osteoporosis at the same time.

Table 13 – Database overview: group 'No fractures' represents cases with fractures while group 'Fractures' represents the opposite case.

Description	Age	BMD (g/cm^2)	T-score	#exams	Group	#selected
Osteopenia	69 ± 5.5	0.85 ± 0.05	-1.9 ± 0.4	12	'No Fractures'	97
Osteoporosis	70 ± 6.5	0.73 ± 0.05	-2.9 ± 0.6	12	$24 \times 5 = 120$	VBs
Osteopenia	75 ± 5.8	0.84 ± 0.07	-1.7 ± 0.3	12	'Fractures'	97
Osteoporosis	71 ± 5.5	0.67 ± 0.06	-3.4 ± 0.4	11	$23 \times 5 = 115$	VBs
			Total	47	235	194

Source: Adapted from Ramos *et al.* (2021).

BMD is calculated in g/cm2, and then compared against a reference population. The comparison yields a T-score, which shows how much higher or lower the bone density is compared to a healthy 30-year-old[3]. The lower the score, the weaker the bones are (LEWIECKI; BORGES, 2006), as follows:

T-score of -1.0 or above: normal bone density.

T-score between -1.0 and -2.5: low bone density, or osteopenia.

T-score of -2.5 or lower: osteoporosis.

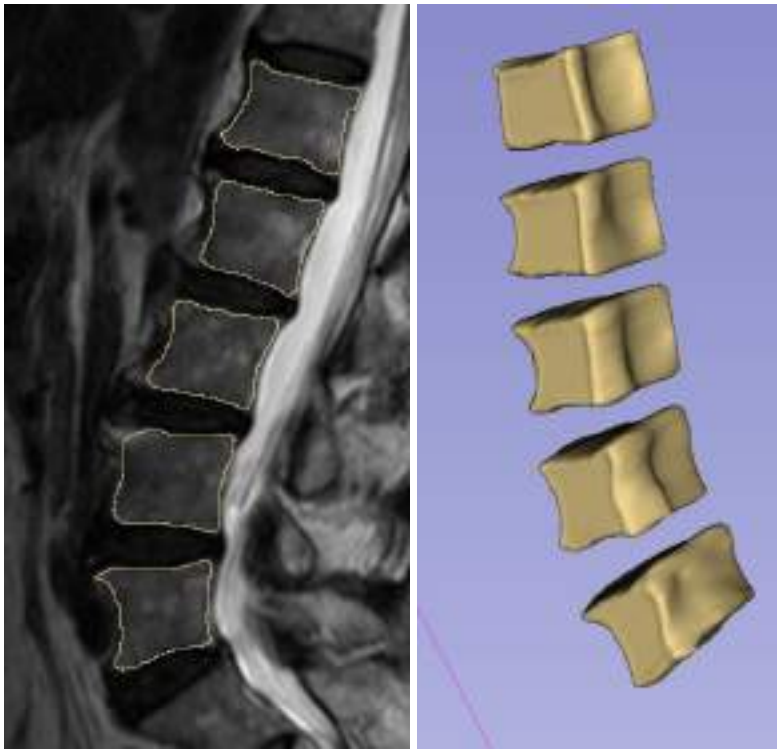
T-score of -2.5 or lower with a fragility fracture: severe osteoporosis.

[3] The age when bones are at their strongest.

7.2.2 Segmentation and feature extraction

We performed a segmentation (Figure 43) of L1-L5 for each T2 exam using Clever Segmentation (RAMOS *et al.*, 2019b) in 3D Slicer[4](KIKINIS; PIEPER; VOSBURGH, 2014). Manual corrections were performed in order to remove over segmentations or to correct under segmentations. The segmentations were performed by an experienced technician under expert supervision and took, in average, 15 minutes per exam.

Figure 43 – Example of segmentations for exam 10.



(a) T2 – MRI exam. (b) T2 – 3D Segmentation.

Source: Elaborated by the author.

In total, 1,674 texture features were extracted from each segmented vertebral body using PyRadiomics (GRIETHUYSEN *et al.*, 2017), as shown in Table 14. That is:

- 93 texture features were extracted from the original exams.
- 837 texture features were extract by adding a few variations using Laplacian of Gaussian (LoG) to the original exams.
- 744 texture features were extract by adding Coiflets Wavelets variations to the original exams.

[4] <https://github.com/JonathanRamos/SlicerCleverSegmentation>

The LoG and Wavelets variations are frequently applied in this Radiomics features extraction. We used LoG sigma values of 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, and 5. For Wavelets, we used default decomposition settings, yielding 8 variations. Each Wavelet variation considered High-pass (H) and Low-pass (L) filters in the following combinations: LLH, LHL, LHH, HLL, HLH, HHL, HHH, and LLL. We re-sampled every exam to isotropic spacing.

Table 14 – Summary of texture features considered in this work.

Name	#features	#variations	
		LoG	Wavelets
Neigh. Gray Tone Difference Matrix (NGTDM)	5	9	8
Gray Level Dependence Matrix (GLDM)	14	9	8
Gray Level Run-Length Matrix (GLRLM)	16	9	8
Gray Level size zone matrix (GLSZM)	16	9	8
First-Order Statistics (FOS)	18	9	8
Gray Level Concurrence Matrix (GLCM)	24	9	8
Quantity	93	54 (837)	48 (744)
Total	93 + 837 + 744 = 1,674		

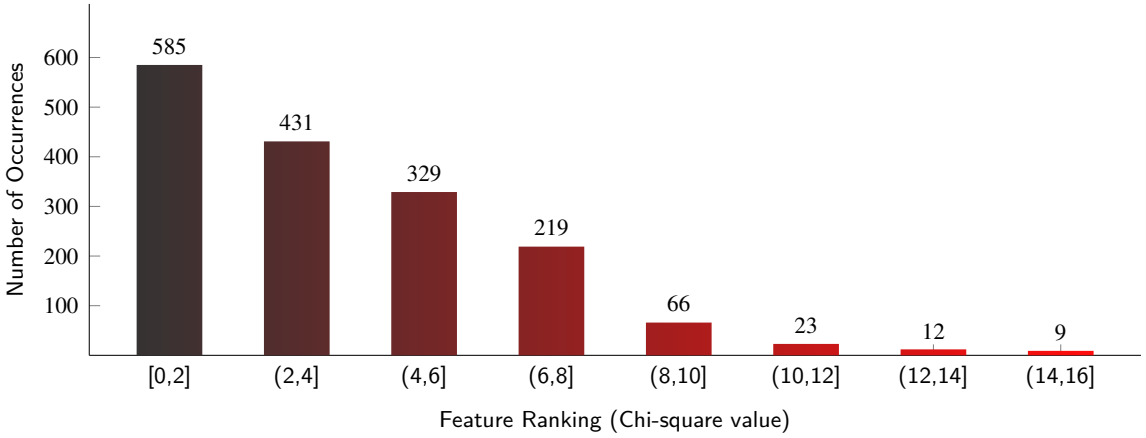
Source: Adapted from Ramos *et al.* (2021).

7.2.3 Feature Selection

We employed the Chi-square univariate feature ranking over the whole database to select the most relevant texture features. In statistics, the Chi-square test checks the independence of two events/variables. The test measures how much two variables deviate from each other. The smaller the Chi-Square value, the less independent the feature is from the label/class. In feature selection, the more dependent on the label/class, the more powerful the feature is to differentiate that label/class. Therefore, the higher the Chi-Square value, the more relevant the feature is (JIN *et al.*, 2006; BOUCKAERT; FRANK, 2004).

The Chi-square test has been used in a few radiomics approaches and yielded promising results (JIN *et al.*, 2006; BOUCKAERT; FRANK, 2004), so that we applied the Chi-square test for feature ranking in our work as well. Due to the total number of features, we show, in Figure 44, the ranking results as an histogram. We empirically selected features with ranking larger than 5.28, summing up to 434 features. Those features comprises 78 unique features across the original VOIs and LoG/Wavelet variations. We will focus on presenting the 10 most frequent features, as follows: 19 GLCM Contrast, 14 NGTDM Gray Level Variance, 14 GLRLM Gray Level Non Uniformity, 11 GLCM Cluster Tendency, 11 GLCM Maximum Probability, 9 FOS Variance, 9 GLCM Joint Entropy, 9 GLCM Imc1, 9 GLCM Correlation, 9 FOS 90Percentile. We made available a complete description of the features selected at <https://github.com/JonathanRamos/beaut>.

Figure 44 – Histogram for Chi-square univariate feature ranking. The farthest to the left (black), the lower the ranking, inversely, the farthest to the right (red), the higher the ranking.



Source: Adapted from Ramos *et al.* (2021).

7.2.4 Classifiers

To validate our proposed approach, BEAUT we made comparison with five representative and well-known literature classifiers. In order to achieve the best results, we trained/tuned the hyper-parameters of all classifiers. The tuning was performed by a Bayesian optimization and acquisition function of expected improvement per second plus with a max of 100 iterations. We detail the algorithms and tuned hyper-parameters as follows.

- Support Vector Machine (SVM): calculates the hyper-plane that best separates the classification groups in the training data. The parameters for all and for selected texture features are respectively: kernel function: *{Gaussian, Gaussian}* and kernel scale: *{5.60, 3.92}*, box constraint: *{168.95, 971.73}*.

- *k*-Nearest Neighbor (*k*-NN): observes the distance among training instances. The parameters for all and selected texture features are respectively: Distance:*{Spearman, City-block}*, number of neighbors:*{1, 1}*, distance weight: *{squared inverse, inverse}*.

- Naïve Bayes: estimates the probability of a feature based on how frequent it appears in the training data.

- Discriminant Analysis (Disc. Analysis): finds a linear combination of features that characterizes or separates the classes. The parameters for all and selected texture features are respectively: discriminant: *{diagonal linear, diagonal linear}*, fill coefficients: *{no, no}*.

- Decision Tree: takes advantage of the “divide-and-conquer” technique, where a problem is divided in sub-problems to derive a decision tree. The parameters for all and selected texture features are respectively: split criterion: *{cross entropy, Gini's diversity index}*, maximum number of decision splits: *{8, 8}*, surrogate: *{no, no}*.

For BEAUT, we set AdaBoost's learning cycles to 496 and a learning rate of 0.12 considering a decision tree learner with 4 maximum number of splits. For LogitBoost, we used 186 learning cycles and a learning rate of 0.90 considering a decision tree learner with at most 3. AdaBoost gets the outputs of a given initial classifier, considered as a "weak learner", and combines the predictions into a weighted sum that represents the final output. In general, using a decision tree as AdaBoost's weak learner tends to present better results (KÉGL, 2013). The AdaBoost method can be divided in two versions, M1 and M2, in which M stands for "Multi-class". For binary classification problems (our case), M1 and M2 are equivalent (FREUND; SCHAPIRE, 1996). LogitBoost takes AdaBoost as a generalized additive model and applies logistic regression as a cost function. Therefore, LogitBoost can be seen as a convex optimization approach aiming at minimizing the logistic loss (FRIEDMAN; HASTIE; TIBSHIRANI, 2000).

To avoid the loss of generalizability, parameters not reported in this section were set to the respective (widely used) default settings of each classifier.

7.2.5 Parameter settings

In our experimental setup, we employed ten repeated 10-fold cross-validation (stratified) for the classification as well as for the statistical testing. Moreover, we trained the hyper-parameters of every classifier considered in the analysis aiming at achieving the best result of each method.

7.2.5.1 Computational setup

Every experiment was performed on a 3.40GHz Intel(R) Core(TM) i7 CPU and 8GB RAM machine, using Matlab(R) version 2020a.

7.2.5.2 Statistical tests

We compared the repeated cross-validation Accuracy results of two classification models. Considering that our sample size is limited and the data follows a normal distribution, we employed a t-test at the 1% and 5% significance level. We considered 10 degrees of freedom for the t-distribution to find the critical region and estimate the p-value. The null hypothesis is: "the set of predictions resulting from classification model 'A' is more accurate than classification model 'B'."

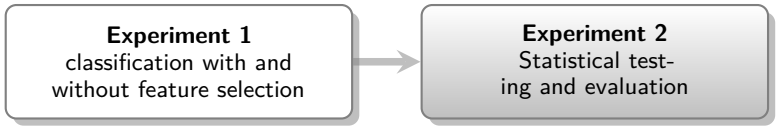
7.2.6 Comparison measures

For comparison, we used well-established measures, such as Accuracy (Equation 2.7), Precision (Equation 2.8), Sensitivity (Equation 2.9), F-Measure (Equation 2.11), and AUC (Equation 2.13).

7.3 Experiments, results and discussion

In our experimental design, we analyze two main parts, as depicted in Figure 45 and described as follows. First, in Section 7.3.1 we report the classification results with and without feature selection. Then, in Section 7.3.2 we present statistical testing, in which we analyze whether there were significant Accuracy differences among the classification models.

Figure 45 – Experimental design to validate BEAUT.



Source: Elaborated by the author.

7.3.1 Classification with and without feature selection

Table 15 reports the classification results with (+) and without (–) feature selection. We discuss the results in two parts as follows.

Without feature selection. BEAUT achieved the best results, presenting the largest values for every measure while *k*-NN reached the second-best results for every measure. SVM exhibited slightly lower results than *k*-NN while the remaining classifiers (Decision Tree, Disc. Analysis, and Naïve Bayes) performed poorly compared to BEAUT and *k*-NN.

To further validate the results, we show in Figure 46a the ROC plot along with the AUC value of each classifier. BEAUT presented an AUC of 92%, giving the highest result. SVM obtained the second-best ROC curve (89%) result and *k*-NN was the third (83%). Naïve Bayes, Discriminant Analysis, and Decision Tree obtained the smaller AUC results, at 61%, 73%, and 76%, respectively.

With selected features. Once again, BEAUT reached the highest results for every measure. Compared to the results using all features, BEAUT was able to improve 8%, in average. The SVM classifier achieved the second-best results for all measures, and improved 3% in average. The performance of the *k*-NN method reduced 5% in average, when compared to the results using all features. Decision Tree, Disc. Analysis, and Naïve Bayes methods presented slightly better results with features selection. Except the *k*-NN classifier, all other methods yielded better overall results when feature selection is applied.

Figure 46b shows a further validation of the results the ROC plot along with the AUC results of each classifier. As we can see, BEAUT presented the highest AUC (97%), SVM presented the second best result (90%). The *k*-NN method presented an AUC of 89%, corresponding to the third-best result. Disc. Analysis, Naïve Bayes, and Decision Tree

presented the lowest AUC, with 74%, 76%, and 80%, respectively. Most methods presented better result with selected features compared to the AUC with all features in Figure 46a. *k*-NN was the only method that presented lower AUC with feature selection.

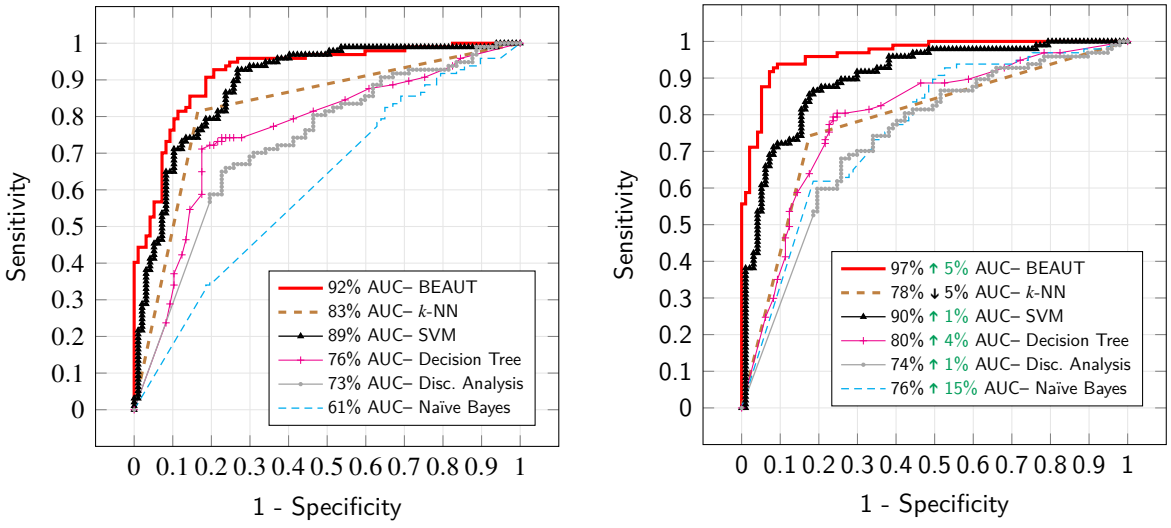
Table 15 – Classification results with (+) and without (–) selected features.

	Measure (%)									
	Accuracy		Precision		Sensitivity		Specificity		F-Measure	
Classifier	–	+	–	+	–	+	–	+	–	+
BEAUT	**85**	**92** ↑ **8**	**84**	**93** ↑ **9**	**86**	**91** ↑ **5**	**84**	**93** ↑ **9**	**85**	**92** ↑ **7**
k-NN	83	78 ↓ 5	82	76 ↓ 6	84	83 ↓ 1	81	74 ↓ 7	83	79 ↓ 4
SVM	80	83 ↑ 3	81	82 ↑ 1	78	84 ↑ 6	81	81 ↓ 0	80	83 ↑ 3
Decision Tree	75	76 ↑ 1	75	76 ↑ 1	76	77 ↑ 1	74	75 ↑ 1	76	77 ↑ 1
Disc. Analysis	70	71 ↑ 1	69	70 ↑ 1	73	73 ↓ 0	67	68 ↑ 1	71	71 ↓ 0
Naïve Bayes	57	68 ↑ 11	55	67 ↑ 12	79	71 ↓ 8	35	65 ↑ 30	65	69 ↑ 4

Source: Adapted from Ramos *et al.* (2021).

Note – Best column values in **bold** and second-best underlined.We represent equal or decreased result with selected feature as ↓ and improved results with selected features as ↑.

Figure 46 – ROC results for classifiers considering all and selected features. We represent AUC increase and decrease in (b), compared to (a), as ↑ and ↓ , respectively.



(a) All features.

(b) Selected Features.

Source: Adapted from Ramos *et al.* (2021).

7.3.2 Statistical testing

We performed the statistical tests in two parts considering the Accuracy of each classifier. First, we compared each classifier using selected features against every classifier using all

Table 16 – Accuracy testing (p-values) of classifiers with selected features against classifiers with all features, at the 5% significance level (p-values lower than 0.05, $p < 0.05$).

		All Features					
	against	BEAUT	k-NN	SVM	Decision Tree	Disc. Analysis	Naïve Bayes
Selected Features	BEAUT	0.017 (✔)	0.031 (✔)	0.006 (✔)	0.001 (✔)	0.001 (✔)	0.001 (✔)
	k-NN	0.985 (✗)	0.856 (✗)	0.618 (✗)	0.021 (✔)	0.005 (✔)	0.006 (✔)
	SVM	0.963 (✗)	0.870 (✗)	0.485 (✗)	0.042 (✔)	0.004 (✔)	0.002 (✔)
	Decision Tree	0.999 (✗)	0.991 (✗)	0.969 (✗)	0.317 (✗)	0.162 (✗)	0.123 (✗)
	Disc. Analysis	0.999 (✗)	0.999 (✗)	0.999 (✗)	0.790 (✗)	0.510 (✗)	0.691 (✗)
	Naïve Bayes	0.9995 (✗)	0.9992 (✗)	0.9983 (✗)	0.7815 (✗)	0.531 (✗)	0.001 (✔)

Source: Adapted from Ramos *et al.* (2021).

Note – ✔ means significantly better results for row classifier and ✗ means that no significant difference is observed for the classifier of that column.

features, as shown in Table 16. Next, in Table 17, we assessed whether BEAUT using selected features presents significantly better Accuracy than the competitors using selected features as well. In both cases, for each p-value, the null hypothesis is that each row approach has greater Accuracy than the technique in the corresponding column.

Analyzing the principal diagonal in Table 16, we can see that only BEAUT and Naïve Bayes using selected features presented a significantly better Accuracy than itself using all features. In the first column with p-values, BEAUT using selected features presented significantly better Accuracy than every classifier using all features, exhibiting low p-values ($p < 0.02$) most of the times. In the second and third columns with p-values, k-NN and SVM using selected features presented significantly better Accuracy than Decision Tree, Discriminant Analysis, and Naïve Bayes using all features. For the remaining cases, no significant differences were observed, presenting, most of the times, large p-values ($p > 0.9$).

Table 17 – Accuracy testing (p-value) for classifiers using selected features, at the 1% significance level ($p < 0.01$).

against	k-NN	SVM	Decision Tree	Disc. Analysis	Naïve Bayes
BEAUT	0.0035 (✔)	0.0044 (✔)	0.0003 (✔)	0.0004 (✔)	0.0004 (✔)

Source: Adapted from Ramos *et al.* (2021).

Note – ✔ means there was significantly better Accuracy for BEAUT.

We can see in Table 17 that BEAUT presented a significantly better Accuracy than every other classifier using selected features at the 1% significance level. Notice that the p-value is quite low ($p < 0.004$) in every case, indicating that BEAUT has a higher discriminating power

than every compared methods.

7.4 Final considerations

In this chapter we presented BEAUT, a new approach aimed at improving the ability to identify patients with fragility fractures secondary to osteoporosis, using the texture analysis on spine MRI of lumbar vertebrae not affected by previous fractures. To achieve better assessment of patients at risk of developing fragility fractures, we investigated the use of texture features extracted from volumetric segmentation of lumbar vertebral bodies. We used radiomics features and well-known classifiers (with trained hyper-parameters) to aid in the assessment of patients with fragility fractures, differentiating them from patients without fragility fractures.

Concerning the objectives of this Doctorate research, in this chapter we approached the **Research Problem 3.** Assessing bone fragility fractures using volumetric MRI and radiomic texture features. Based on the results presented throughout this chapter, we answer the initial formulated research questions as follows:

a) *Can we assess bone fragility secondary to osteoporosis using MRI?*

Answer:: The experimental results on a meaningful dataset with 47 T2-weighted MRI exams showed that, by using BEAUT with texture feature selection, we achieved a better discriminating performance for differentiation of patients at risk of developing lumbar fragility fractures. Our proposed method, BEAUT, achieved a significantly better Accuracy (92%), when compared to competitor classifiers (that achieved at most 89%). Moreover, BEAUT reached 97% AUC, which surpassed all the competitors as well. Thus, our results open the possibility of future use of spine MRI to differentiate patients at risk for developing fragility fractures.

b) *Is it possible to find potential biomarkers to predict bone fragility fractures?*

Answer: We ascertain that quantitative texture features can potentially discriminate individuals with bone mass loss and identify those at risk of fragility fractures. We believe that further studies, with a larger database, can be conducted to identify isolated attributes, or a set of attributes, to differentiate subjects at risk of fragility fractures. Future longitudinal studies are necessary to evaluate if texture features extracted from spine MRI may really predict future fragility fracture incidence.

Our study has limitations that deserve mention:

1. Machine learning was able to differentiate patients with and without osteoporotic fractures, based on features extracted from vertebrae without fractures and with normal appearance

on MRI, but we did not test if the algorithm can predict future development of fractures. Future studies are necessary to address this question.

2. The database we used was relatively small and from the same institution. Thus, we did not evaluate the generalizability of our method with spine MRI performed in other institutions.

CHAPTER

8

CONCLUSIONS

In this Doctorate research we aimed at advancing in the semi-automatic segmentation and analysis of vertebral bodies obtained by MRI, focusing on reducing the manual interaction. Specifically, we presented three main contributions, that we discuss next, along with the research hypothesis and thesis of the Doctorate research. We organized this Chapter in three parts as follows. First, Section 8.1 states our main contributions. Next, Section 8.2 lists the future work we envision. Finally, Section 8.3 summarizes the publications resulted from this PhD research.

8.1 Contributions

As we stated in Chapter 1, our main goal of this Doctorate research was to answer the following question: "*How can we ease manual interaction in semi-automatic segmentation of vertebral bodies in MRI exams and, at the same time, keep or improve the segmentation quality?*" This is a challenging question, due to vertebrae deformations/fractures, diversity of MRI acquisition parameters, high computational cost of segmentation algorithms, as well as how to embed the expert-knowledge acquired over several years of experience without adding observer-bias. From this context, we contributed in three main aspects:

1. BGrowth method for the segmentation of fractured or deformed vertebral bodies in MRI.

   BGrowth method better delineated VCFs boundaries and achieved the best Accuracy while keeping acceptable Run-Time performance. We highlight that we employed a rough/sloppy manual annotation and BGrowth presented the best results.

2. Family of segmentation methods to reduce manual interaction on the segmentation of volumetric structures in spine in MRI.

   We proposed a family of segmentation methods, in which FastCleverSeg represented the approach in this challenging task of reducing manual interaction in the volumetric

segmentation process. FastCleverSeg worked along with EANIS, requiring the annotation of only two or three slices to produce an accurate volumetric segmentation. We showed that, combining FastCleverSeg and EANIS makes the segmentation process faster, regarding processing time and manual annotation, making it easier the production of reference manual segmentations.

3. BEAUT approach to identify patients at risk of bone fragility fractures in MRI using Radiomics.

   By using texture features in a radiomics approach, BEAUT was able to achieve high diagnostic performance for differentiating patients with and without vertebral body fragility fracture. The best results were obtained with feature selection, achieving up to 97% of AUC and 92% of Accuracy.

Now regarding the hypothesis of this Doctorate research:

Hypothesis

The segmentation of vertebral bodies in MRI can be performed computationally faster with easier manual interaction and, at the same time, producing accurate results.

We are able to say that our segmentation method performs faster using less manual interaction and, at the same time, keep comparable or better results than other state-of-the-art methods. Next we point out future developments that can be done, based on the results of the research achieved in this Doctorate work.

8.2 Future Work

As future work we envision the following developments:

Asses inter and intra observer subjective variability: to test subjective variability of details employed in the manual annotation step, a few studies should be conducted to further validate our segmentation methods. We believe such experiment would assess how well our methods can incorporate expert knowledge in the segmentation, specially in the case of VCFs segmentation.

Extending/incorporate FastCleverSeg into 3D Slicer software: So far, we have only made available CleverSeg as a 3D Slicer extension. As future work, we look forward to implementing FastCleverSeg as well as EANIS in 3D Slicer. This will allow a faster annotation process, consequently, a faster overall segmentation in a friendly graphical user interface. Thus, we believe more ground-truth can be produced in a faster manner. Therefore, machine learning algorithms (deep learning approaches) can be trained to segment considering a lot more samples.

Compare our family of segmentation methods with Deep Learning approaches : perform a throughout analysis of our methods against neural networks (Deep Learning) models to the performance of our approaches.

Perform longitudinal studies to validate BEAUT: since our approach was able to differentiate patients with and without osteoporotic fractures, based on features extracted from vertebrae without fractures and with normal appearance on MRI, we did not test if the algorithm can predict future development of fractures. Thus, future studies are necessary to evaluate if texture features extracted from spine MRI may really predict future fragility fracture incidence. To do so, bigger databases, coming from distinct institutions, should be used to evaluate the generalizability of our method with spine MRI performed in other institutions.

8.3 List of Publications

In this section, we divide the publications in three parts, along with the Qualis-CC Capes[1]. First, in Section 8.3.1, we describe the main publications, which resulted directly from this Doctorate research. Next, in Section 8.3.2, we report the submitted works, that are woks under review by the peers. Finally, in Section 8.3.3, we detail the complementary publications, the ones resulting from contributions with GBdI[2] research group. We highlight that GBdI also helped achieving the results reported in this thesis and contributed to the formation of the PhD candidate.

8.3.1 Main publications

We describe the main publications in chronological order as follows.

1. SAC 2019, Qualis-CC Capes A2:

   RAMOS, J. S.; WATANABE, C. Y. V.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M. BGrowth: An efficient approach for the segmentation of vertebral compression fractures in magnetic resonance imaging. **Symposium on Applied Computing (SAC)**, Association for Computing Machinery, New York, NY, USA, p. 220–227, 2019. Available: <https://doi.org/10.1145/3297280.3299728>.

2. CBMS 2019, Qualis-CC Capes A3:

   RAMOS, J. S.; CAZZOLATO, M. T.; FAIÇAL, B. S.; NOGUEIRA-BARBOSA, M. H.; JR., C. T.; TRAINA, A. J. M. 3DBGrowth: volumetric vertebrae segmentation and reconstruction in magnetic resonance imaging. **International Symposium on Computer-**

[1] <https://ppgcc.github.io/discentesPPGCC/pt-BR/qualis/>
[2] <http://www.gbdi.icmc.usp.br/>

Based Medical Systems (CBMS), p. 435–440, June 2019. Available: <https://doi.org/10.1109/CBMS.2019.00091>.

3. SIBGRAPI 2019, Qualis-CC Capes A3:

   RAMOS, J. S.; CAZZOLATO, M. T.; FAIÇAL, B. S.; LINARES, O. A. C.; NOGUEIRA-BARBOSA, M. H.; Traina Jr., C.; TRAINA, A. J. M. Fast and smart segmentation of paraspinal muscles in magnetic resonance imaging with CleverSeg. In: **Conference on Graphics, Patterns and Images (SIBGRAPI)**. Manchester: CPS (Conference Publishing Services), 2019. (SIBGRAPI '19), p. 73–83. ISSN 2377-5416. Available: <https://doi.org/10.1109/SIBGRAPI.2019.00019>.

4. SAC 2020, Qualis-CC Capes A2:

   RAMOS, J. S.; CAZZOLATO, M. T.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M. FINE: Improving time and precision of segmentation techniques for vertebral compression fractures in MRI. **Symposium on Applied Computing (SAC)**, Association for Computing Machinery, New York, NY, USA, p. 198–201, 2020. Available: <https://doi.org/10.1145/3341105.3374100>.

5. CARS 2021, Qualis-CC Capes A2:

   RAMOS, J. S.; MACIEL, J. G.; TRAINA, A. J.; AZEVEDO-MARQUES, P. M. de; NOGUEIRA-BARBOSA, M. Spine MRI texture analysis and prediction of osteoporotic vertebral fracture. **Computer Assisted Radiology and Surgery (CARS)**, p. 1, 2021. Accepted for publication.

6. CBMS 2021, Qualis-CC Capes A3:

   RAMOS, J. S.; CAZZOLATO, M. T.; GOMES, J.; JR., C. T.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M. BEAUT: a radiomic approach to predict potential lumbar fractures in magnetic resonance imaging. **International Symposium on Computer-Based Medical Systems (CBMS)**, p. 1–6, 2021. Accepted for publication.

8.3.2 Work under review

We report the work submitted and under review as follows.

1. JDI 2021, Qualis-CC Capes A2:

   RAMOS, J. S.; CAZZOLATO, M. T.; FAIÇAL, B. S.; LINARES, O. C.; MENEZES-REIS, R.; AZEVEDO-MARQUES, P. M.; NOGUEIRA-BARBOSA, M. H.; JR., C. T.; ; TRAINA, A. J. M. Fast and accurate volumetric segmentation of spine anatomical structures in magnetic resonance imaging using fcleverseg. **PLOS ONE**, p. 1–12, 2021. Under review.

8.3.3 Complementary contributions

We detail the complementary publications, in partnership with GBdI as follows.

1. CBMS 2020, Qualis-CC Capes A3:

   CAZZOLATO, M. T.; RAMOS, J. S.; RODRIGUES, L. S.; SCABORA, L. C.; CHINO, D. Y. T.; JORGE, A. E. S.; AZEVEDO-MARQUES, P. M.; TRAINA, C.; TRAINA, A. J. M. Semi-automatic ulcer segmentation and wound area measurement supporting telemedicine. **International Symposium on Computer-Based Medical Systems (CBMS)**, p. 356–361, 2020. Available: <https:/doi.org/10.1109/CBMS49503.2020.00073>.

2. CBM 2021, Qualis-CC Capes A2:

   CAZZOLATO, M. T.; RAMOS, J. S.; RODRIGUES, L. S.; SCABORA, L. C.; CHINO, D. Y. T.; JORGE, A. E. S.; AZEVEDO-MARQUES, P. M. de; JR., C. T.; TRAINA, A. J. M. The utrack framework for segmenting and measuring dermatological ulcers through telemedicine. **Computers in Biology and Medicine**, p. 1–12, 2021. Available: <https://doi.org/10.1016/j.compbiomed.2021.104489>.

REFERENCES

AHMAD, O. F.; SOARES, A. S.; MAZOMENOS, E.; BRANDAO, P.; VEGA, R.; SEWARD, E.; STOYANOV, D.; CHAND, M.; LOVAT, L. B. Artificial intelligence and computer-aided diagnosis in colonoscopy: current evidence and future directions. **The Lancet Gastroenterology & Hepatology**, v. 4, n. 1, p. 71 – 80, 2019. ISSN 2468-1253. Available: <https://doi.org/10.1016/S2468-1253(18)30282-6>. Citations on pages 1, 23, 24, and 25.

AWAD, M.; KHANNA, R. Support vector machines for classification. In: ____. **Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers**. Berkeley, CA: Apress, 2015. p. 39–66. ISBN 978-1-4302-5990-9. Available: <https://doi.org/10.1007/978-1-4302-5990-9_3>. Citation on page 20.

AZEVEDO-MARQUES, P. M.; SPAGNOLI, H. F.; FRIGHETTO-PEREIRA, L.; MENEZES-REIS, R.; METZNER, G. A.; RANGAYYAN, R. M.; NOGUEIRA-BARBOSA, M. H. Classification of vertebral compression fractures in magnetic resonance images using spectral and fractal analysis. In: **37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2015, Milan, Italy, August 25-29, 2015**. IEEE, 2015. p. 723–726. Available: <https://doi.org/10.1109/EMBC.2015.7318464>. Citations on pages 4 and 5.

BANERJEE, P.; HU, M.; KANNAN, R.; KRISHNASWAMY, S. A semi-automated approach to improve the efficiency of medical imaging segmentation for haptic rendering. **Journal of Digital Imaging (JDI)**, v. 30, n. 4, p. 519–27, Aug 2017. ISSN 1618-727X. Available: <https://doi.org/10.1007/s10278-017-9985-2>. Citation on page 3.

BARBIERI, P. D.; PEDROSA, G. V.; TRAINA, A. J. M.; NOGUEIRA-BARBOSA, M. H. Vertebral body segmentation of spine MR images using Superpixels. In: JUNIOR, C. T.; RODRIGUES, P. P.; KANE, B.; MARQUES, P. M. de A.; TRAINA, A. J. M. (Ed.). **Symposium on Computer-Based Medical Systems (CBMS)**. São Carlos and Ribeirão Preto, BR: Conference Publishing Services (CPS), 2015. p. 44–49. ISSN 1063-7125. Available: <https://doi.org/10.1109/CBMS.2015.11>. Citations on pages 4, 5, 39, and 119.

BICAN, O.; MINAGAR, A.; PRUITT, A. A. The spinal cord: A review of functional neuroanatomy. **Neurologic Clinics**, v. 31, n. 1, p. 1 – 18, 2013. ISSN 0733-8619. Spinal Cord Diseases. Available: <https://doi.org/10.1016/j.ncl.2012.09.009>. Citation on page 27.

BOUCKAERT, R. R.; FRANK, E. Evaluating the replicability of significance tests for comparing learning algorithms. In: DAI, H.; SRIKANT, R.; ZHANG, C. (Ed.). **Advances in Knowledge Discovery and Data Mining**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 3–12. ISBN 978-3-540-24775-3. Available: <https://doi.org/10.1007/978-3-540-24775-3_3>. Citation on page 93.

BRESENHAM, J. E. Algorithm for computer control of a digital plotter. **IBM Systems Journal**, v. 4, n. 1, p. 25–30, 1965. ISSN 0018-8670. Available: <https://doi.org/10.1147/sj.41.0025>. Citation on page 72.

BROOMHEAD, D.; LOWE, D. Multivariable functional interpolation and adaptive networks. **Complex Systems**, v. 2, p. 321–355, 1988. Citation on page 19.

BURIAN, E.; ROHRMEIER, A.; SCHLAEGER, S.; DIECKMEYER, M.; DIEFENBACH, M. N.; SYVÄRI, J.; KLUPP, E.; WEIDLICH, D.; ZIMMER, C.; RUMMENY, E. J.; KARAMPINOS, D. C.; KIRSCHKE, J. S.; BAUM, T. Lumbar muscle and vertebral bodies segmentation of chemical shift encoding-based water-fat MRI: the reference database MyoSegmenTUM spine. **BMC Musculoskeletal Disorders**, v. 20, n. 1, p. 152, Apr 2019. ISSN 1471–2474. Available: <https://doi.org/10.1186/s12891-019-2528-x>. Citations on pages 3, 4, 64, 76, and 77.

CASTI, P.; MENCATTINI, A.; NOGUEIRA-BARBOSA, M. H.; FRIGHETTO-PEREIRA, L.; AZEVEDO-MARQUES, P. M.; MARTINELLI, E.; NATALE, C. D. Cooperative strategy for a dynamic ensemble of classification models in clinical applications: the case of MRI vertebral compression fractures. **International Journal of Computer Assisted Radiology and Surgery**, v. 12, n. 11, p. 1971–1983, Nov 2017. Available: <https://doi.org/10.1007/s11548-017-1625-2>. Citations on pages 4, 5, and 30.

CAZZOLATO, M. T.; RAMOS, J. S.; RODRIGUES, L. S.; SCABORA, L. C.; CHINO, D. Y. T.; JORGE, A. E. S.; AZEVEDO-MARQUES, P. M.; TRAINA, C.; TRAINA, A. J. M. Semi-automatic ulcer segmentation and wound area measurement supporting telemedicine. **International Symposium on Computer-Based Medical Systems (CBMS)**, p. 356–361, 2020. Available: <https:/doi.org/10.1109/CBMS49503.2020.00073>. Citations on pages 8 and 105.

CAZZOLATO, M. T.; RAMOS, J. S.; RODRIGUES, L. S.; SCABORA, L. C.; CHINO, D. Y. T.; JORGE, A. E. S.; AZEVEDO-MARQUES, P. M. de; JR., C. T.; TRAINA, A. J. M. The utrack framework for segmenting and measuring dermatological ulcers through telemedicine. **Computers in Biology and Medicine**, p. 1–12, 2021. Available: <https://doi.org/10.1016/j.compbiomed.2021.104489>. Citations on pages 8 and 105.

CAZZOLATO, M. T.; SCABORA, L. C.; NESSO, M. R.; MILANO-OLIVEIRA, L. F.; COSTA, A. F.; KASTER, D. S.; KOENIGKAM-SANTOS, M.; AZEVEDO-MARQUES, P. M. de; TRAINA, C.; TRAINA, A. J. dp-BREATH: Heat maps and probabilistic classification assisting the analysis of abnormal lung regions. **Computer Methods and Programs in Biomedicine (CMPB)**, v. 173, p. 27–34, 2019. ISSN 0169-2607. Available: <https://doi.org/10.1016/j.cmpb.2019.01.014>. Citation on page 3.

CHAN, T. F.; VESE, L. A. Active contours without edges. **IEEE Transactions on Image Processing**, IEEE Press, Piscataway, NJ, USA, v. 10, n. 2, p. 266–277, Feb. 2001. ISSN 1057-7149. Available: <https://doi.org/10.1109/83.902291>. Citations on pages 39 and 40.

CHANG, G.; BOONE, S.; MARTEL, D.; RAJAPAKSE, C. S.; HALLYBURTON, R. S.; VALKO, M.; HONIG, S.; REGATTE, R. R. MRI assessment of bone structure and microarchitecture. **Journal of Magnetic Resonance Imaging**, Wiley, v. 46, n. 2, p. 323–337, Feb. 2017. Available: <https://doi.org/10.1002/jmri.25647>. Citation on page 6.

CHU, C.; BELAVÝ, D. L.; ARMBRECHT, G.; BANSMANN, M.; FELSENBERG, D.; ZHENG, G. **Annotated T2-weighted MR images of the Lower Spine**. 2015. Available: <https://doi.org/10.5281/zenodo.22304>. Citations on pages 4 and 76.

CHU, C.; BELAVÝ, D. L.; ARMBRECHT, G.; BANSMANN, M.; FELSENBERG, D.; ZHENG, G. Fully automatic localization and segmentation of 3D vertebral bodies from CT/MR images via a learning-based method. **PLOS ONE**, Public Library of Science, v. 10, n. 11, p. 1–22, 11 2015. Available: <https://doi.org/10.1371/journal.pone.0143327>. Citations on pages 4, 76, and 77.

CONDE, A.; DOMÍNGUEZ, J. Scaling the chord and hellinger distances in the range [0,1]: An option to consider. **Journal of Asia-Pacific Biodiversity**, v. 11, n. 1, p. 161 – 166, 2018. ISSN 2287-884X. Available: <https://doi.org/10.1016/j.japb.2018.01.001>. Citation on page 18.

CUÉNOD, C. A.; LAREDO, J. D.; CHEVRET, S.; HAMZE, B.; NAOURI, J. F.; CHAPAUX, X.; BONDEVILLE, J. M.; TUBIANA, J. M. Acute vertebral collapse due to osteoporosis or malignancy: appearance on unenhanced and gadolinium-enhanced MR images. **Radiology**, v. 199, n. 2, p. 541–549, 1996. Available: <https://doi.org/10.1148/radiology.199.2.8668809>. Citation on page 29.

DURON, L.; BALVAY, D.; PERRE, S. V.; BOUCHOUICHA, A.; SAVATOVSKY, J.; SADIK, J.-C.; THOMASSIN-NAGGARA, I.; FOURNIER, L.; LECLER, A. Gray-level discretization impacts reproducible mri radiomics texture features. **PloS one**, Public Library of Science, v. 14, n. 3, p. 1–14, Mar 2019. ISSN 1932-6203. Available: <https://doi.org/10.1371/journal.pone.0213459>. Citations on pages 1, 24, and 25.

EGGER, J.; NIMSKY, C.; CHEN, X. Vertebral body segmentation with GrowCut: Initial experience, workflow and practical application. **SAGE Open Med**, v. 5, p. 1–10, 2017. Available: <https://doi.org/10.1177/2050312117740984>. Citations on pages 40 and 53.

ENGSTROM, C. M.; FRIPP, J.; JURCAK, V.; WALKER, D. G.; SALVADO, O.; CROZIER, S. Segmentation of the quadratus lumborum muscle using statistical shape modeling. **Jounal of Magnetic Resonance Imaging (JMRI)**, v. 33, n. 6, p. 1422–1429, 2011. Available: <https://doi.org/10.1002/jmri.22188>. Citation on page 5.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. Inteligência artificial: Uma abordagem de aprendizado de máquina. **Livros Técnicos e Científicos**, 2011. Citation on page 18.

FALCÃO, A. X.; STOLFI, J.; LOTUFO, R. de A. The image foresting transform: Theory, algorithms, and applications. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, USA, v. 26, n. 1, p. 19–29, Jan. 2004. ISSN 0162-8828. Available: <https://doi.org/10.1109/TPAMI.2004.1261076>. Citations on pages 39 and 40.

FATOYE, F.; GEBRYE, T.; ODEYEMI, I. Real-world incidence and prevalence of low back pain using routinely collected data. **Rheumatology International**, v. 39, n. 4, p. 619–626, Apr 2019. ISSN 1437-160X. Available: <https://doi.org/10.1007/s00296-019-04273-0>. Citation on page 2.

FREBURGER, J. K.; HOLMES, G. M.; AGANS, R. P.; JACKMAN, A. M.; DARTER, J. D.; WALLACE, A. S.; CASTEL, L. D.; KALSBEEK, W. D.; CAREY, T. S. The rising prevalence of chronic low back pain. **Archives of internal medicine**, v. 169, n. 3, p. 251–258, Feb 2009. ISSN 1538-3679. Available: <https://doi.org/10.1001/archinternmed.2008.543>. Citation on page 2.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: **Thirteenth International Conference on Machine Learning**. San Francisco: Morgan Kaufmann, 1996. p. 148–156. Available: <https://doi.org/10.5555/3091696.3091715>. Citations on pages 20, 89, and 95.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). **Ann. Statist.**, The Institute of Mathematical Statistics, v. 28, n. 2, p. 337–407, 04 2000. Available: <https://doi.org/10.1214/aos/1016218223>. Citations on pages 20, 89, and 95.

FRIGHETTO-PEREIRA, L.; MENEZES-REIS, R.; METZNER, G. A.; RANGAYYAN, R. M.; NOGUEIRA-BARBOSA, M. H.; AZEVEDO-MARQUES, P. M. Classification of vertebral compression fractures in magnetic resonance images using shape analysis. In: **2015 E-Health and Bioengineering Conference (EHB)**. [s.n.], 2015. p. 1–4. Available: <https://doi.org/10.1109/EHB.2015.7391550>. Citations on pages 4 and 5.

FRIGHETTO-PEREIRA, L.; MENEZES-REIS, R.; METZNER, G. A.; RANGAYYAN, R. M.; MARQUES, P. M. de A.; NOGUEIRA-BARBOSA, M. H. Semiautomatic classification of benign versus malignant vertebral compression fractures using texture and gray-level features in magnetic resonance images. In: JR., C. T.; RODRIGUES, P. P.; KANE, B.; MARQUES, P. M. de A.; TRAINA, A. J. M. (Ed.). **28th IEEE International Symposium on Computer-Based Medical Systems, CBMS 2015, Sao Carlos, Brazil, June 22-25, 2015**. IEEE Computer Society, 2015. p. 88–92. Available: <https://doi.org/10.1109/CBMS.2015.37>. Citations on pages 4 and 5.

FRIGHETTO-PEREIRA, L.; METZNER, G. A.; MARQUES, P. M. de A.; NOGUEIRA-BARBOSA, M. H.; OLOUMI, F.; RANGAYYAN, R. M. Recognition of vertebral compression fractures in magnetic resonance images using statistics of height and width. In: **2016 IEEE International Symposium on Medical Measurements and Applications, MeMeA 2016, Benevento, Italy, May 15-18, 2016**. IEEE, 2016. p. 1–5. Available: <https://doi.org/10.1109/MeMeA.2016.7533731>. Citations on pages 4 and 5.

FRIGHETTO-PEREIRA, L.; RANGAYYAN, R. M.; METZNER, G. A.; MARQUES, P. M. de A.; NOGUEIRA-BARBOSA, M. H. Shape, texture and statistical features for classification of benign and malignant vertebral compression fractures in magnetic resonance images. **Computers in Biology and Medicine**, v. 73, p. 147–156, 2016. Available: <https://doi.org/10.1016/j.compbiomed.2016.04.006>. Citation on page 29.

GAONKAR, B.; XIA, Y.; VILLAROMAN, D. S.; KO, A.; ATTIAH, M.; BECKETT, J. S.; MACYSZYN, L. Multi-parameter ensemble learning for automated vertebral body segmentation in heterogeneously acquired clinical MR images. **Journal of Translational Engineering in Health and Medicine (JTEHM)**, v. 5, p. 1–12, 2017. ISSN 2168-2372. Available: <https://doi.org/10.1109/JTEHM.2017.2717982>. Citations on pages 2 and 5.

GEMAN, S.; GEMAN, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PAMI-6, n. 6, p. 721–741, Nov 1984. ISSN 0162-8828. Available: <https://doi.org/10.1109/TPAMI.1984.4767596>. Citation on page 40.

GILLIES, R. J.; KINAHAN, P. E.; HRICAK, H. Radiomics: Images are more than pictures, they are data. **Radiology**, v. 278, n. 2, p. 563–577, 2016. PMID: 26579733. Available: <https://doi.org/10.1148/radiol.2015151169>. Citations on pages 14, 25, and 26.

GODINHO, T. M.; VIANA-FERREIRA, C.; SILVA, L. A. B.; COSTA, C. A routing mechanism for cloud outsourcing of medical imaging repositories. **IEEE Journal of Biomedical and Health Informatics**, v. 20, n. 1, p. 367–375, Jan 2016. ISSN 2168-2194. Available: <https://doi.org/10.1109/JBHI.2014.2361633>. Citation on page 1.

GONZALEZ, R. **Digital Image Processing**. Upper Saddle River, NJ: Pearson Education, 2009. ISBN 9788131726952. Citations on pages 13, 14, and 15.

GONZALEZ, R.; WOODS, R. **Digital Image Processing**. Upper Saddle River, NJ: Pearson Education, 2011. ISBN 9780133002324. Citation on page 12.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Citation on page 61.

GRIETHUYSEN, J. J. M. van; FEDOROV, A.; PARMAR, C.; HOSNY, A.; AUCOIN, N.; NARAYAN, V.; BEETS-TAN, R. G. H.; FILLION-ROBIN, J. C.; PIEPER, S.; AERTS, H. J. W. L. Computational Radiomics System to Decode the Radiographic Phenotype. **Cancer Res**, v. 77, n. 21, p. 104–107, 2017. Citation on page 92.

GROBEL, M. C. B.; TELLES, V. L. C. N. Da comunicação visual pré-histórica ao desenvolvimento da linguagem escrita, e a evolução da autenticidade documentoscópica. **Revista Acadêmica Oswaldo Cruz**, v. 1, p. 1–12, 2014. ISSN 2357-81873. Available: <http://revista.oswaldocruz.br/Edicao_01/Artigos>. Citation on page 1.

HAMMERNIK, K.; EBNER, T.; STERN, D.; URSCHLER, M.; POCK, T. Vertebrae segmentation in 3D CT images based on a variational framework. In: YAO, J.; GLOCKER, B.; KLINDER, T.; LI, S. (Ed.). **Recent Advances in Computational Methods and Clinical Applications for Spine Imaging**. Cham: Springer International Publishing, 2015. p. 227–233. ISBN 978-3-319-14148-0. Available: <http://doi.org/10.1007/978-3-319-14148-0_20>. Citations on pages 2 and 6.

HILLE, G.; SAALFELD, S.; SEROWY, S.; TöNNIES, K. Vertebral body segmentation in wide range clinical routine spine MRI data. **Computer Methods and Programs in Biomedicine**, v. 155, p. 93–99, 2018. ISSN 0169-2607. Available: <https://doi.org/10.1016/j.cmpb.2017.12.013>. Citation on page 5.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning: With Applications in R**. Springer, New York, NY: Springer Publishing Company, Incorporated, 2014. ISBN 1461471370, 9781461471370. Available: <https://doi.org/10.1007/978-1-4614-7138-7>. Citation on page 19.

JIANG, X.; ZHANG, R.; NIE, S. Image segmentation based on level set method. **Physics Procedia - International Conference on Medical Physics and Biomedical Engineering**, v. 33, p. 840 – 845, 2012. ISSN 1875-3892. Available: <https://doi.org/10.1016/j.phpro.2012.05.143>. Citation on page 40.

JIN, X.; XU, A.; BIE, R.; GUO, P. Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. In: LI, J.; YANG, Q.; TAN, A.-H. (Ed.). **Data Mining for Biomedical Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 106–115. ISBN 978-3-540-33105-6. Available: <https://doi.org/10.1007/11691730_11>. Citation on page 93.

JUNIOR, J. R. F.; KOENIGKAM-SANTOS, M.; CIPRIANO, F. E. G.; FABRO, A. T.; AZEVEDO-MARQUES, P. M. de. Radiomics-based features for pattern recognition of lung cancer histopathology and metastases. **Computer Methods and Programs in Biomedicine**, v. 159, p. 23–30, 2018. ISSN 0169-2607. Available: <https://doi.org/10.1016/j.cmpb.2018.02.015>. Citations on pages 2 and 39.

KAHN, C. E.; CARRINO, J. A.; FLYNN, M. J.; PECK, D. J.; HORII, S. C. Dicom and radiology: Past, present, and future. **Journal of the American College of Radiology**, v. 4, n. 9, p. 652 – 657, 2007. ISSN 1546-1440. Available: <https://doi.org/10.1016/j.jacr.2007.06.004>. Citation on page 23.

KARIMI, D.; SAMEI, G.; KESCH, C.; NIR, G.; SALCUDEAN, S. E. Prostate segmentation in MRI using a convolutional neural network architecture and training strategy based on statistical shape models. **International Journal of Computer Assisted Radiology and Surgery (IJCARS)**, v. 13, n. 8, p. 1211–1219, 2018. Available: <https://doi.org/10.1007/s11548-018-1785-8>. Citation on page 3.

KÉGL, B. **The return of AdaBoost.MH: multi-class Hamming trees**. 2013. Available: <https://arxiv.org/abs/1312.6086>. Citation on page 95.

KIKINIS, R.; PIEPER, S. D.; VOSBURGH, K. G. 3D slicer: a platform for subject-specific image analysis, visualization, and clinical support. In: JOLESZ, F. A. (Ed.). **Intraoperative Imaging and Image-Guided Therapy**. New York, NY: Springer New York, 2014. p. 277–289. ISBN 978-1-4614-7657-3. Available: <https://doi.org/10.1007/978-1-4614-7657-3_19>. Citations on pages 5 and 92.

KOREZ, R.; LIKAR, B.; PERNUŠ, F.; VRTOVEC, T. Model-based segmentation of vertebral bodies from MR images with 3D CNNs. In: OURSELIN, S.; JOSKOWICZ, L.; SABUNCU, M. R.; UNAL, G.; WELLS, W. (Ed.). **Medical Image Computing and Computer-Assisted Intervention (MICCAI)**. Cham: Springer Int. Publishing, 2016. p. 433–441. ISBN 978-3-319-46723-8. Available: <http://doi.org/10.1007/978-3-319-46723-8_50>. Citations on pages 2 and 5.

KUMAR, V.; GU, Y.; BASU, S.; BERGLUND, A.; ESCHRICH, S. A.; SCHABATH, M. B.; FORSTER, K.; AERTS, H. J.; DEKKER, A.; FENSTERMACHER, D.; GOLDGOF, D. B.; HALL, L. O.; LAMBIN, P.; BALAGURUNATHAN, Y.; GATENBY, R. A.; GILLIES, R. J. Radiomics: the process and the challenges. **Magnetic resonance imaging**, v. 30, n. 9, p. 1234–1248, Nov. 2012. ISSN 1873-5894. Available: <https://doi.org/10.1016/j.mri.2012.06.010>. Citations on pages 1, 14, 24, and 25.

LAMBIN, P.; RIOS-VELAZQUEZ, E.; LEIJENAAR, R.; CARVALHO, S.; STIPHOUT, R. G. P. M. van; GRANTON, P.; ZEGERS, C. M. L.; GILLIES, R.; BOELLARD, R.; DEKKER, A.; AERTS, H. J. W. L. Radiomics: Extracting more information from medical images using advanced feature analysis. **European Journal of Cancer**, Elsevier, v. 48, n. 4, p. 441–446, 2017/09/19 2012. ISSN 0959-8049. Available: <https://doi.org/10.1016/j.ejca.2011.11.036>. Citation on page 25.

LAROBINA, M.; MURINO, L. Medical image file formats. **Journal of digital imaging**, Springer US, v. 27, n. 2, p. 200–206, Apr 2014. ISSN 1618-727X. Available: <https://doi.org/10.1007/s10278-013-9657-9>. Citation on page 23.

LEWIECKI, E. M.; BORGES, J. A. L. C. Bone density testing in clinical practice. **Arquivos Brasileiros de Endocrinologia & Metabologia**, scielo, v. 50, p. 586 – 595, 08 2006. ISSN 0004-2730. Available: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27302006000400004&nrm=iso>. Citation on page 91.

LI, Y.; SUN, J.; TANG, C.-K.; SHUM, H.-Y. Lazy Snapping. **ACM Transactions on Graphs**, ACM, New York, NY, USA, v. 23, n. 3, p. 303–308, Aug. 2004. ISSN 0730-0301. Available:

<http://doi-acm-org.ez67.periodicos.capes.gov.br/10.1145/1015706.1015719>. Citations on pages 40 and 119.

LÓPEZ, E.; IBARZ, E.; HERRERA, A.; PUÉRTOLAS, S.; GABARRE, S.; MÁS-VAL, Y.; MATEO-AGUDO, J.; GIL-ALBAROVA, J.; GRACIA, L. A predictive mechanical model for evaluating vertebral fracture probability in lumbar spine under different osteoporotic drug therapies. **Comput. Methods Programs Biomed.**, v. 131, p. 37–50, 2016. Available: <https://doi.org/10.1016/j.cmpb.2016.04.006>. Citation on page 6.

LU, Y.-W.; JIANG, J.-G.; QI, M.-B.; ZHAN, S.; YANG, J. Segmentation method for medical image based on improved GrabCut. **International Journal of Imaging Systems and Technology**, v. 27, n. 4, p. 383–390, 2017. Available: <https://doi.org/10.1002/ima.22242>. Citation on page 39.

MACIEL, J. G.; ARAÚJO, I. M. de; TRAZZI, L. C.; AZEVEDO-MARQUES, P. M. de; SALMON, C. E. G.; PAULA, F. J. A. de; NOGUEIRA-BARBOSA, M. H. Association of bone mineral density with bone texture attributes extracted using routine magnetic resonance imaging. **Clinics**, Fundacao Faculdade de Medicina, v. 75, 2020. Available: <https://doi.org/10.6061/clinics/2020/e1766>. Citation on page 6.

MAROŠEVIĆ, T. The Hausdorff distance between some sets of points. **Mathematical Communications**, v. 23, n. 2, Jan 2018. ISSN 1848-8013. Available: <https://www.mathos.unios.hr/mc/index.php/mc/article/view/2540>. Citation on page 22.

MARTINEZ, J. M.; KOENEN, R.; PEREIRA, F. MPEG-7: the generic multimedia content description standard, part 1. **IEEE MultiMedia**, v. 9, n. 2, p. 78–87, 2002. Available: <https://doi.org/10.1109/93.998074>. Citation on page 17.

MASSEY, F. J. The Kolmogorov-Smirnov test for goodness of fit. **Journal of the American Statistical Association**, American Statistical Association, v. 46, n. 253, p. 68–78, 1951. Available: <https://doi.org/10.2307/2280095>. Citations on pages 42, 57, 68, and 124.

MCCLUNG, M. R. The relationship between bone mineral density and fracture risk. **Current Osteoporosis Reports**, v. 3, n. 2, p. 57–63, Jun 2005. ISSN 1544-2241. Available: <https://doi.org/10.1007/s11914-005-0005-y>. Citation on page 6.

MEHTA, S. D.; SEBRO, R. Computer-aided detection of incidental lumbar spine fractures from routine dual-energy x-ray absorptiometry (DEXA) studies using a support vector machine (SVM) classifier. **J. Digit. Imaging**, v. 33, n. 1, p. 204–210, 2020. Available: <https://doi.org/10.1007/s10278-019-00224-0>. Citation on page 6.

MENEZES-REIS, R.; SALMON, C. E.; BONUGLI, G. P.; MAZOROSKI, D.; TAMASHIRO, M. H.; SAVARESE, L. G.; NOGUEIRA-BARBOSA, M. H. Lumbar intervertebral discs T2 relaxometry and T1ρ relaxometry correlation with age in asymptomatic young adults. **Quant Imaging Med Surg**, v. 6, n. 4, p. 402–412, Aug 2016. Available: <https://doi.org/10.21037/qims.2016.08.01>. Citations on pages 76 and 77.

MY-MS.ORG. **MRI Plane Mathematics**. 2008. [Online; accessed 10-January-2021]. Available: <https://my-ms.org/mri_planes.htm>. Citations on pages 12 and 13.

OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, Jan 1979. ISSN 0018-9472. Available: <https://doi.org/10.1109/TSMC.1979.4310076>. Citations on pages 14, 39, and 40.

PEREIRA, L. F. **Classificação semiautomática de fraturas vertebrais benignas e malignas em imagens de ressonância magnética**. Master's Thesis (Master's Thesis) — Faculdade de Medicina de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2016. Available: <https://teses.usp.br/teses/disponiveis/17/17142/tde-07062017-100712/pt-br.php>. Citation on page 29.

RAMOS, J. S.; CAZZOLATO, M. T.; FAIÇAL, B. S.; LINARES, O. C.; MENEZES-REIS, R.; AZEVEDO-MARQUES, P. M.; NOGUEIRA-BARBOSA, M. H.; JR., C. T.; ; TRAINA, A. J. M. Fast and accurate volumetric segmentation of spine anatomical structures in magnetic resonance imaging using fcleverseg. **PLOS ONE**, p. 1–12, 2021. Under review. Citations on pages 8, 72, 73, 77, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, and 104.

RAMOS, J. S.; CAZZOLATO, M. T.; FAIÇAL, B. S.; NOGUEIRA-BARBOSA, M. H.; JR., C. T.; TRAINA, A. J. M. 3DBGrowth: volumetric vertebrae segmentation and reconstruction in magnetic resonance imaging. **International Symposium on Computer-Based Medical Systems (CBMS)**, p. 435–440, June 2019. Available: <https://doi.org/10.1109/CBMS.2019.00091>. Citations on pages 3, 5, 7, 50, 54, 55, 56, 57, 58, 65, 66, 76, 78, 104, 119, and 125.

RAMOS, J. S.; CAZZOLATO, M. T.; FAIÇAL, B. S.; LINARES, O. A. C.; NOGUEIRA-BARBOSA, M. H.; Traina Jr., C.; TRAINA, A. J. M. Fast and smart segmentation of paraspinal muscles in magnetic resonance imaging with CleverSeg. In: **Conference on Graphics, Patterns and Images (SIBGRAPI)**. Manchester: CPS (Conference Publishing Services), 2019. (SIBGRAPI '19), p. 73–83. ISSN 2377-5416. Available: <https://doi.org/10.1109/SIBGRAPI.2019.00019>. Citations on pages 7, 60, 61, 66, 67, 68, 71, 76, 78, 92, and 104.

RAMOS, J. S.; CAZZOLATO, M. T.; GOMES, J.; JR., C. T.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M. BEAUT: a radiomic approach to predict potential lumbar fractures in magnetic resonance imaging. **International Symposium on Computer-Based Medical Systems (CBMS)**, p. 1–6, 2021. Accepted for publication. Citations on pages 8, 90, 91, 93, 94, 97, 98, and 104.

RAMOS, J. S.; CAZZOLATO, M. T.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M. FINE: Improving time and precision of segmentation techniques for vertebral compression fractures in MRI. **Symposium on Applied Computing (SAC)**, Association for Computing Machinery, New York, NY, USA, p. 198–201, 2020. Available: <https://doi.org/10.1145/3341105.3374100>. Citations on pages 104, 120, 122, 123, 127, 128, and 129.

RAMOS, J. S.; MACIEL, J. G.; TRAINA, A. J.; AZEVEDO-MARQUES, P. M. de; NOGUEIRA-BARBOSA, M. Spine MRI texture analysis and prediction of osteoporotic vertebral fracture. **Computer Assisted Radiology and Surgery (CARS)**, p. 1, 2021. Accepted for publication. Citations on pages 8 and 104.

RAMOS, J. S.; WATANABE, C. Y. V.; NOGUEIRA-BARBOSA, M. H.; TRAINA, A. J. M. BGrowth: An efficient approach for the segmentation of vertebral compression fractures in magnetic resonance imaging. **Symposium on Applied Computing (SAC)**, Association for Computing Machinery, New York, NY, USA, p. 220–227, 2019. Available: <https://doi.org/10.1145/3297280.3299728>. Citations on pages 7, 35, 36, 38, 103, 119, 120, and 121.

RAVAL, K.; ZAFAR, S. Digital watermarking with copyright authentication for image communication. In: **International Conference on Intelligent Systems and Signal Processing (ISSP)**. Gujarat, India: IEEE, 2013. p. 111–116. Available: <https://doi.org/10.1109/ISSP.2013.6526885>. Citation on page 1.

ROTHER, C.; KOLMOGOROV, V.; BLAKE, A. GrabCut: Interactive foreground extraction using iterated graph cuts. **ACM Transactions on Graphs**, ACM, New York, USA, v. 23, n. 3, p. 309–314, Aug. 2004. ISSN 0730-0301. Available: <https://doi.org/10.1145/1186562.1015720>. Citations on pages 40 and 119.

SAMET, H. **Foundations of Multidimensional and Metric Data Structures**. 500 Sansome Street, Suite 400, San Francisco, CA 94111: Elsevier Science, 2006. (Morgan Kaufmann). ISBN 9780123694461. Citation on page 17.

SILVA, W. S.; JASBICK, D. L.; WILSON, R. E.; AZEVEDO-MARQUES, P. M.; TRAINA, A. J. M.; SANTOS, L. F. D.; JORGE, A. E. S.; OLIVEIRA, D. de; BEDO, M. V. N. A two-phase learning approach for the segmentation of dermatological wounds. **Computer-Based Medical Systems (CBMS)**, p. 1–6, June 2019. Available: <https://doi.org/10.1109/CBMS.2019.00076>. Citation on page 60.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, v. 45, n. 4, p. 427 – 437, 2009. ISSN 0306-4573. Available: <https://doi.org/10.1016/j.ipm.2009.03.002>. Citation on page 21.

SPINE, M. B. . **Spinal fractures**. 2018. Available: <https://mayfieldclinic.com/pe-SpineFract.htm>. Accessed: 20/04/2018. Citations on pages 27, 28, and 29.

STEFAN, P.; HABERT, S.; WINKLER, A.; LAZAROVICI, M.; FÜRMETZ, J.; ECK, U.; NAVAB, N. A radiation-free mixed-reality training environment and assessment concept for C-arm-based surgery. **IJCARS**, v. 13, n. 9, p. 1335–1344, Sep 2018. ISSN 1861-6429. Available: <https://doi.org/10.1007/s11548-018-1807-6>. Citation on page 3.

STUTZ, D.; HERMANS, A.; LEIBE, B. Superpixels: an evaluation of the state-of-the-art. **Computer Vision and Image Understanding**, v. 166, p. 1 – 27, 2018. ISSN 1077-3142. Available: <https://doi.org/10.1016/j.cviu.2017.03.007>. Citation on page 15.

TAHA, A. A.; HANBURY, A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. **BMC Medical Imaging**, v. 15, p. 1–29, Aug 2015. Available: <https://doi.org/10.1186/s12880-015-0068-x>. Citations on pages 21 and 22.

TELLACHE, M.; PITHIOUX, M.; CHABRAND, P.; HOCHARD, C. Femoral neck fracture prediction by anisotropic yield criteria. **European Journal of Computational Mechanics**, Taylor & Francis, v. 18, n. 1, p. 33–41, 2009. Available: <https://doi.org/10.3166/ejcm.18.33-41>. Citation on page 6.

TIMMEREN, J. E. van; CESTER, D.; TANADINI-LANG, S.; ALKADHI, H.; BAESSLER, B. Radiomics in medical imaging—"how-to" guide and critical reflection. **Insights into Imaging**, v. 11, n. 1, p. 91, Aug 2020. ISSN 1869-4101. Available: <https://doi.org/10.1186/s13244-020-00887-2>. Citations on pages 1 and 25.

VANNESCHI, L.; CLERGUE, M.; COLLARD, P.; TOMASSINI, M.; VÉREL, S. Fitness clouds and problem hardness in genetic programming. In: SPRINGER. **Genetic and Evolutionary Computation Conference**. 2004. p. 690–701. Available: <https://doi.org/10.1007/978-3-540-24855-2_76>. Citation on page 49.

VEZHNEVETS, V.; KONOUCHINE, V. GrowCut - interactive multi-label N-D image segmentation by cellular automata. **International Conference on Computer Graphics and Vision**

(GraphiCon), v. 1, Nov 2005. Available: <https://www.graphicon.ru/oldgr/en/research/Segment/index.html>. Citations on pages 5, 15, 40, 65, 76, 78, 119, and 125.

VIKHAR, P.; KARDE, P. Exploring differential evolution algorithm for content based image retreival system. **International Conference for Convergence in Technology (I2CT)**, IEEE, Mumbai, India, p. 1217–1219, April 2017. Available: <https://doi.org/10.1109/I2CT.2017.8226320>. Citation on page 23.

WANG, W.; LU, Y. Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, v. 324, p. 012049, mar 2018. Available: <https://doi.org/10.1088/1757-899X/324/1/012049>. Citation on page 60.

WANG, Y. X. J.; SANTIAGO, F. R.; DENG, M.; NOGUEIRA-BARBOSA, M. H. Identifying osteoporotic vertebral endplate and cortex fractures. **Quantitative Imaging Medical Surgery (QIMS)**, v. 7, n. 5, p. 555–591, October 2017. Available: <http://doi.org/10.21037/qims.2017.10.05>. Citation on page 2.

WILCOXON, F.; KATTI, S.; WILCOX, R. Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. **Selected Tables in Math. Statistics**, v. 1, p. 171–259, 1970. Citations on pages 43, 57, 68, and 125.

WU, H.-Z.; ZHANG, X.-F.; HAN, S.-M.; CAO, L.; WEN, J.-X.; WU, W.-J.; GAO, B.-L. Correlation of bone mineral density with MRI T2* values in quantitative analysis of lumbar osteoporosis. **Archives of Osteoporosis**, v. 15, n. 1, p. 18, Feb 2020. ISSN 1862-3514. Available: <https://doi.org/10.1007/s11657-020-0682-2>. Citation on page 6.

WU, S.; NAKAO, M.; MATSUDA, T. Automatic grabcut based lung extraction from endoscopic images with an initial boundary. In: **International Conference on Signal Processing (ICSP)**. [s.n.], 2016. p. 1374–1378. ISSN 2164-5221. Available: <https://doi.org/10.1109/ICSP.2016.7878051>. Citation on page 39.

XU, Y.; LIN, L.; HU, H.; WANG, D.; LIU, Y.; WANG, J.; HAN, X.; CHEN, Y. W. A retrieval system for 3D multi-phase contrast-enhanced CT images of focal liver lesions based on combined bags of visual words and texture words. **International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)**, IEEE, Datong, China, p. 806–810, Oct 2016. Available: <https://doi.org/10.1109/CISP-BMEI.2016.7852820>. Citations on pages 1, 24, and 25.

XUE, Z.; LONG, L. R.; ANTANI, S.; THOMA, G. R. Spine X-ray image retrieval using partial vertebral boundaries. In: **Symposium on Computer-Based Medical Systems (CBMS)**. [s.n.], 2011. p. 1–6. ISSN 1063-7125. Available: <https://doi.org/10.1109/CBMS.2011.5999147>. Citation on page 3.

YONG, Z.; JIAZHENG, Y.; HONGZHE, L.; QING, L. Grabcut image segmentation algorithm based on structure tensor. **The Journal of China Universities of Posts and Telecommunications**, v. 24, n. 2, p. 38 – 47, 2017. ISSN 1005-8885. Available: <https://doi.org/10.1016/S1005-8885(17)60197-3>. Citation on page 39.

ZDILLA, M. J.; HATFIELD, S. A.; MCLEAN, K. A.; CYRUS, L. M.; LASLO, J. M.; LAMBERT, H. W. Circularity, Solidity, Axes of a Best Fit Ellipse, Aspect Ratio, and Roundness of the Foramen Ovale: A Morphometric Analysis With Neurosurgical Considerations.

J Craniofac Surg, v. 27, n. 1, p. 222–228, Jan 2016. Available: <https://doi.org/10.1097/scs.0000000000002285>. Citation on page 122.

ZHU, L.; KOLESOV, I.; GAO, Y.; KIKINIS, R.; TANNENBAUM, A. An effective interactive medical image segmentation method using Fast GrowCut. **Medical Image Computing and Computed-Assisted Intervention (MICCAI)**, v. 17, p. 1–9, 2014. Available: <https://nac.spl.harvard.edu/files/nac/files/zhu-miccai2014.pdf>. Citations on pages 40, 53, 65, 76, and 78.

ZUKIĆ, D.; VLASÁK, A.; EGGER, J.; HOŘÍNEK, D.; NIMSKY, C.; KOLB, A. Robust detection and segmentation for diagnosis of vertebral diseases using routine MR images. **Computer Graphics Forum**, v. 33, n. 6, p. 190–204, 2014. Available: <https://doi.org/10.1111/cgf.12343>. Citations on pages 21, 4, 51, 52, 76, 77, and 119.

APPENDIX

A

FAST INSIDE ANNOTATION – FINE: A STUDY CASE WITH VCF EXAMS

In this Appendix, we present Fast INside Estimation (FINE), which automatically estimates the inside seeds based on the outside annotation. First, we go into the background and motivation in Section A.1. Then, in Section A.2, we present FINE, which automatically finds the suitable thresholds so that erosions and skeletonization can be used to determine the inside annotation without using any fixed or empirically defined limits. After that, in Section A.4, the experimental results on a representative database show that FINE achieved, on average, higher Sensitivity (97%) and F-Measure (96%). Finally, Section A.5 concludes this Appendix.

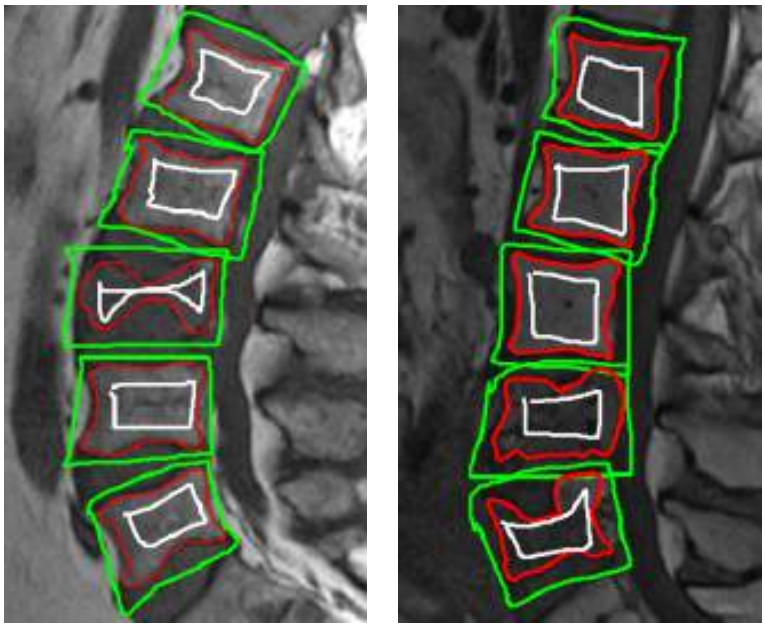
A.1 Background and motivation

As we discussed in Chapter 3, the automatic or semiautomatic segmentation of VCFs is a challenging task. VCFs presents non-homogeneous gray-scale intensities within the same vertebral body (RAMOS *et al.*, 2019), as exemplified example in **??**. An accurate segmentation algorithm plays a significant role and can assist the medical specialist in surgical planning and evaluation of suitable treatments (ZUKIĆ *et al.*, 2014).

We introduced BGrowth in Chapter 3 to cope with the challenging task of VFCs segmentation. BGrowth is a region-growing approach that balances weights during the region expansion. To do so, BGrowth uses sloppy-like or straightforward inside and outside annotation (seed points). BGrowth surpassed every method considered in the analysis: one-seed-point approaches such as VBSeg (BARBIERI *et al.*, 2015) and Otsu 's threshold as well as inside/outside annotation (seeds) approaches, such as GrowCut (VEZHNEVETS; KONOUCHINE, 2005), GrabCut (ROTHER; KOLMOGOROV; BLAKE, 2004) and LazySnapping (LI *et al.*, 2004). In Chapter 4, we extended BGrowth to work with volumetric exams and we surpassed the competitors as well (RAMOS *et al.*, 2019a). However, manually drawing inside and outside annotations is not practical when

dealing with a more significant number of exams.

Figure 47 – Example of sloppy inside (white) and outside (green) annotations (seed points) used by Ramos *et al.* (2019): ground-truth (GT) in red.



Source: Adapted from Ramos *et al.* (2020).

In general, semiautomatic segmentation methods, such as BGrowth, and GrowCut require manual annotations (seeds) on the inside and outside of the object of interest (vertebral bodies in this work). To reduce the total time spent on manual annotation, we propose Fast INside Estimation (FINE), which estimates the inside annotation/seeds based on the shape of the outside annotation. Thus, allowing a more accurate segmentation of the vertebral body, and, at the same time, reducing the manual annotation time. To this end, we employ the erosion and skeletonization of the outside annotation in a reduced manner, as detailed next.

A.2 Proposed method: Fast INside Estimation – FINE

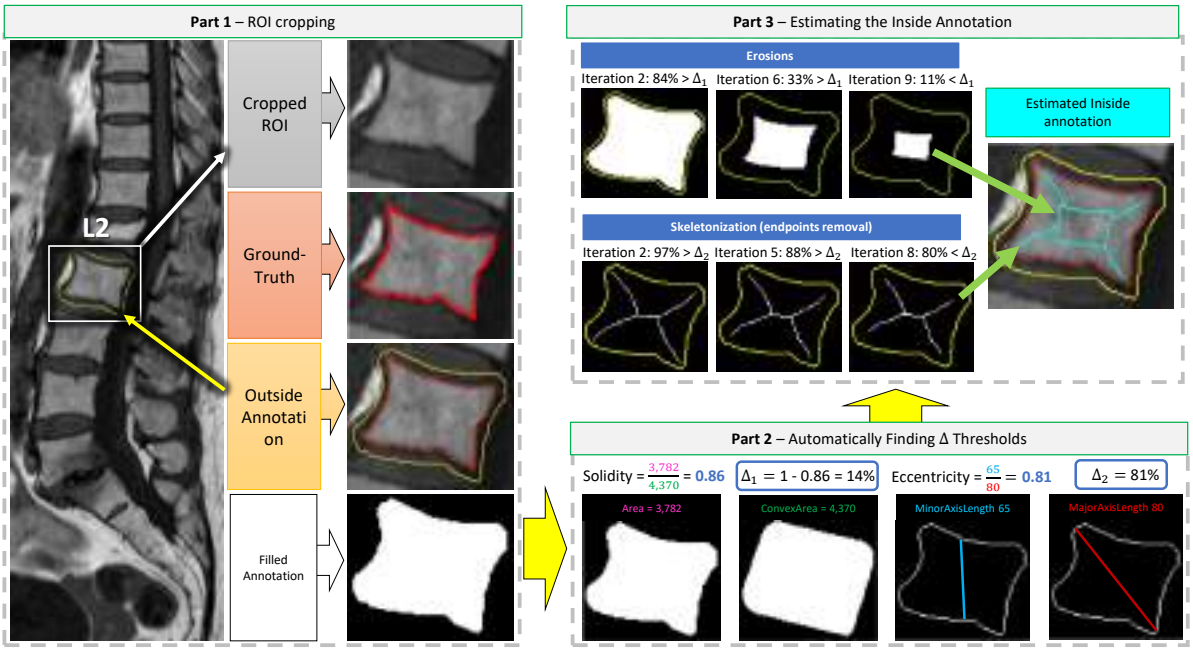
As depicted in Figure 48, we divide FINE into three main parts, as follows.

Part 1: crop the Region of Interest (ROI) based on the outside annotation, which is manually given. We fill the outside annotation and represent it as k, as detailed in Section A.2.1.

Part 2: define the most suitable percentage to reduce the regions. We automatically calculate two shape measures (Δ_1 and Δ_2) from the filled outside annotation, the *solidity* and the *eccentricity*, as defined in Section A.2.2.

Part 3: estimate the inside annotation by joining an eroded region of k and a smaller portion of the skeleton of k. First, we iteratively erode the region k until it is smaller than a threshold

Figure 48 – FINE pipeline: example of ROI and manual outside annotation as well as estimation of the inside annotation.



Source: Adapted from Ramos *et al.* (2019).

value (Δ_1). Then, we iteratively remove endpoints from the skeleton of k, until the number of pixels in the smaller skeleton associated to the initial skeleton is smaller than a threshold (Δ_2). Finally, we join the eroded result with the shrank skeleton result to produce the final estimated inside annotation, as we discuss in Section A.2.3.

A.2.1 Cropping the ROI

Given the outside annotation as a circular list of points in a closed contour, we fill the region (represented as k) using morphological operations. We use the convex hull of k to crop the ROI. However, in general, the semiautomatic segmentation methods use an 8-neighbor schema and does not produce the best results in this area of the image. For this reason, we add a margin error of 2 pixels for every side of the ROI so that each segmentation method can achieve its best result.

A.2.2 Finding Δ values automatically

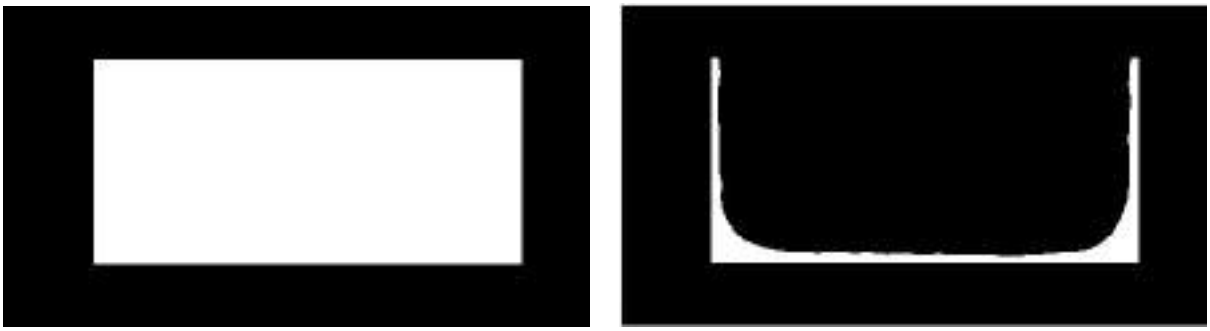
Let us define two shape measures that we will employ in our method, Solidity and Eccentricity, as follows.

Definition 7 (Solidity)**.** The proportion of pixels in the convex hull (convex hull area of k) that are also in the region (area of k):

$$\text{Solidity} = \frac{\text{area of } k}{\text{convex area of } k}\,, \tag{A.1}$$

which describes the extent to which a given shape is convex or concave. The Solidity of a completely convex shape is 1, as exemplified in Figure 49a. The farther the Solidity deviates from 1, the higher is the extent of concavity (ZDILLA *et al.*, 2016), as exemplified in Figure 49b.

Figure 49 – Example of Solidity for distinct shapes (in white).



(a) Completely convex shape: Solidity = 1. (b) Shape with sizeable concavity: Solidity = 0.1.

Source: Adapted from Ramos *et al.* (2020).

Definition 8 (Eccentricity)**.** Is the ratio of the length of the short (minor) axis to the length of the long (major) axis of k (ZDILLA *et al.*, 2016):

$$\text{Eccentricity} = \frac{\text{Major Axis Length of } k}{\text{Minor Axis Length of } k} \tag{A.2}$$

The Eccentricity is also known as the aspect ratio or ellipticity. The Eccentricity of a perfect square is 1, as exemplified in Figure 49a, while the more prolonged the major axis length, the closer to 0 the Eccentricity, as illustrated in Figure 50b.

We define two Δ values (shape measures) to use as percentages to reduce the regions. First, we erode the region k up until a given rate, which we define as Δ_1 and is calculated using the Solidity. We take the reciprocal of the Solidity to define the final Δ_1 value:

$$\Delta_1 = 1 - \text{Solidity} \tag{A.3}$$

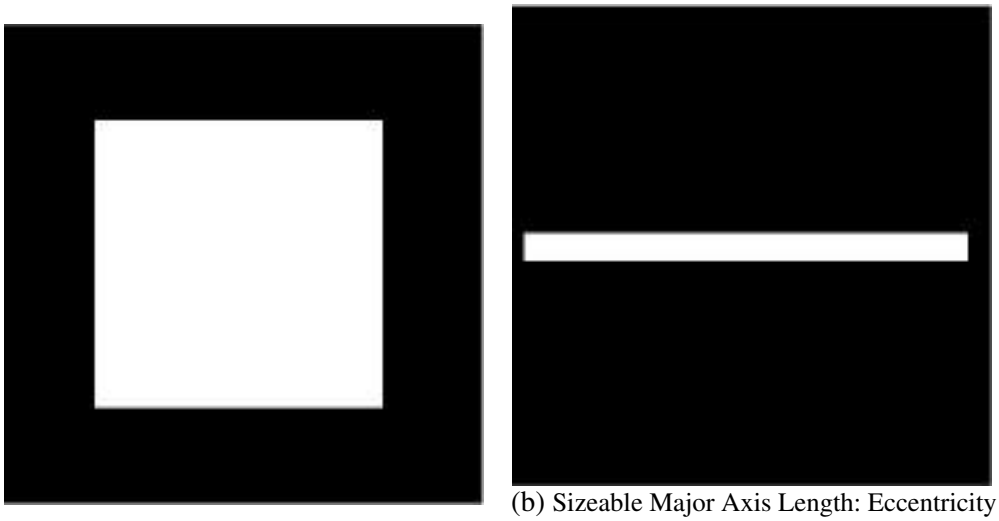
Then, we use the skeleton of k and reduce it up to a given rate as well, which we define as Δ_2, and calculate using the Eccentricity. We define the Eccentricity as the final Δ_2 value:

$$\Delta_2 = \text{Eccentricity} \tag{A.4}$$

A.2.3 Inside annotation estimation

To estimate the inside annotation of the vertebral body based on the given outside annotation (represented as a binary mask k), we join the results of an iterative erosion process (Section A.2.4) along with an iterative reduction of the skeleton (Section A.2.5) of the region k. We can quickly calculate both approaches by using mathematical morphology. FINE automatically compute the threshold rather than empirically defining a static one.

Figure 50 – Example of Eccentricity for distinct shapes (in white).



(a) Perfect Square: Eccentricity = 1.

(b) Sizeable Major Axis Length: Eccentricity = 0.01.

Source: Adapted from Ramos *et al.* (2020).

A.2.4 Sequential erosion

We iteratively apply erosions to the filled outside annotation until the region is smaller than a given threshold (Δ_1). This process has two main steps:

Step 1: set k as the initial interior annotation and t as the total number of white pixels in k and apply an erosion operation over k using a 3×3 square structuring element, which results into a new mask k_1. As a consequence, the number of white pixels in k_1 reduces to t_1. The square structuring element allows a smooth erosion without losing the main shape of the region. In this work, we use a small size (3×3) structuring element, since it allows a more gradating shrinking of the region.

Step 2: apply the same erosion over k_1, which results in k_2 with t_2 white pixels. Repeat the erosion process until the i-th iteration, resulting in the mask k_i with t_i white pixels. The erosion process stops when the percentage of remaining white pixels in the eroded mask (k_i) associated with the initial mask k is lower than a threshold Δ_1:

$$\Delta_1 > \frac{t_i}{t}.$$

After that, we use the k_i mask as part of the inside annotation, as illustrated in Figure 48, Part 3, on the ninth iteration.

A.2.5 Shrinking the skeleton

We represent the skeleton of the initial filled outside annotation as sk. Similarly to the erosion process, we shrink the skeleton iteratively by removing the endpoints of sk:

Step A: set sk as the initial interior annotation and st as the total number of white pixels in sk. Next, remove the endpoints, which are terminal points in the skeleton, and represent this new skeleton as sk_1, which has st_1 white pixels.

Step B: Remove the endpoint of sk_1, which results in the skeleton sk_2. The process continues until the i-th iteration

$$\Delta_2 > \frac{st_i}{st}.$$

Then, we set the reduced skeleton sk_i as part of the inside annotation, as exemplified in Part 3 (the eighth iteration on the skeletonization) of Figure 48.

A.3 Materials and Methods

To validate FINE, we first describe the experimental database in Section A.3.1, which is composed of healthy vertebral bodies, malignant and benign VCFs. Next, in Section A.3.2, we present the parameters settings for each of the semiautomatic segmentation technique considered in our analysis. We also present the computational setup along with the statistical tests used. Then, in Section A.3.3, we present the segmentation methods considered to validate FINE. Finally, in Section A.3.4, we detail the comparison measures employed.

A.3.1 Patient Database

We used the database described in Section 3.2.1, which is composed of 89 healthy vertebral bodies and 102 with VCFs (single slices).

A.3.2 Parameters settings

We used the parameters settings defined by the authors of each method to meet the best conditions of every technique. We compare FINE with the manual inside annotation, and a static threshold $\Delta = 20\%$. The static threshold uses only the erosion process as a simple attempt to estimate the inside annotation. The Δ value was set based on empirical testing. For every segmentation algorithm, no maximum number of iterations was set.

A.3.2.1 Computational set-up

We calculated the total time demanded in each method on a 3.40GHz Intel(R) Core(TM) i7 CPU and 8GB RAM machine, using Matlab(R) version 2018a.

A.3.2.2 Statistical tests

The Kolmogorov-Smirnov (MASSEY, 1951) normality test is the most indicated when dealing with data presenting several similar values. If the Kolmogorov-Smirnov test rejects

the null hypothesis that the data follows a normal distribution, the Wilcoxon (WILCOXON; KATTI; WILCOX, 1970) test may be employed to analyze if there are significant statistical differences among the data (results). In the Wilcoxon test, the null hypothesis is that data from two paired-sample groups came from populations having the same distribution, against the opposite alternative.

A.3.3 Segmentation methods

To evaluate FINE, we considered BGrowth (RAMOS *et al.*, 2019a) and GrowCut (VEZHNEVETS; KONOUCHINE, 2005) segmentation methods.

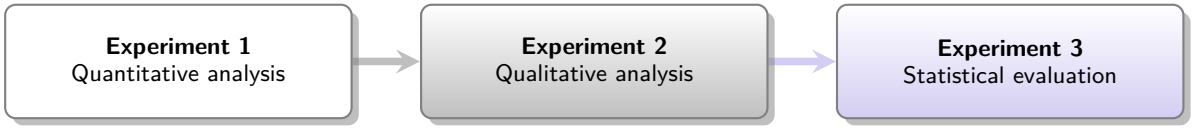
A.3.4 Comparison measures

We considered the measures of Precision (Equation 2.8), Sensitivity (Equation 2.9), F-Measure (Equation 2.11), and Run-Time.

A.4 Experiments, results and discussion

We analyze the experiments in three main parts, as depicted in Figure 51 and detailed as follows. First, in Section A.4.1, we perform a quantitative analysis using the comparison measures (Section A.3.4). Next, in Section A.4.2, we assess the results qualitatively by showing a few segmentation results, both healthy vertebral body and VCF. Finally, in Section A.4.3, to further assess the results, we perform statistical validation between the results of our proposed method against the manual inside annotation and the static threshold (fixed in $\Delta = 20\%$).

Figure 51 – Experimental design to validate FINE.



Source: Elaborated by the author.

A.4.1 Quantitative Analysis

For a better assessment, we divided the results in four parts as follows.

Healthy] vertebral body, which comprises only health vertebral bodies.

Benign VCFs, which considers only osteoporosis cases.

Malignant VFCs, which includes only bone metastasis situations.

Overall results, which merges the results of the three previous parts.

Figure 52 reports the data distribution as well as the average F-Measure, Precision, and Sensitivity achieved by GrowCut and BGrowth, with the three distinct inside annotation. For every case, our proposed approach achieved the best F-Measure and Sensitivity results, the static threshold achieved the second-best results, and the manual inside annotation presented the third-best results. For the Precision measure, our proposed method made the third-best results while the other techniques presented very close results. In average, the difference between our approach and the competitors is small, up to 2%. On the other hand, analyzing the F-Measure (balance between Precision and Sensitivity), we note that our method achieved the best results for each one of the segmentation methods.

On average, the static threshold ($\Delta = 20\%$) took 10 ± 3 ms to estimate the inside annotation, while FINE took 14 ± 4 ms. Our method took only 4 ms more to determine the inside annotation. In general, manually annotating the inside/outside takes several seconds. By using FINE, we reduced the time spent on manual annotation by half and, at the same time, produced higher Sensitivity and F-Measure. To a greater extent, our proposed dynamic threshold FINE achieves higher Sensitivity without losing much Precision.

Analyzing the overall Run-Time in Figure 53, FINE presented faster Run-Time than the manual annotation approach. Compared to the static threshold ($\Delta = 20\%$), FINE obtained similar or quicker results. Although the Run-Time is not a lot shorter than the competitor methods, we contribute to the state-of-the-art by advancing towards the automation of segmentation of medical images.

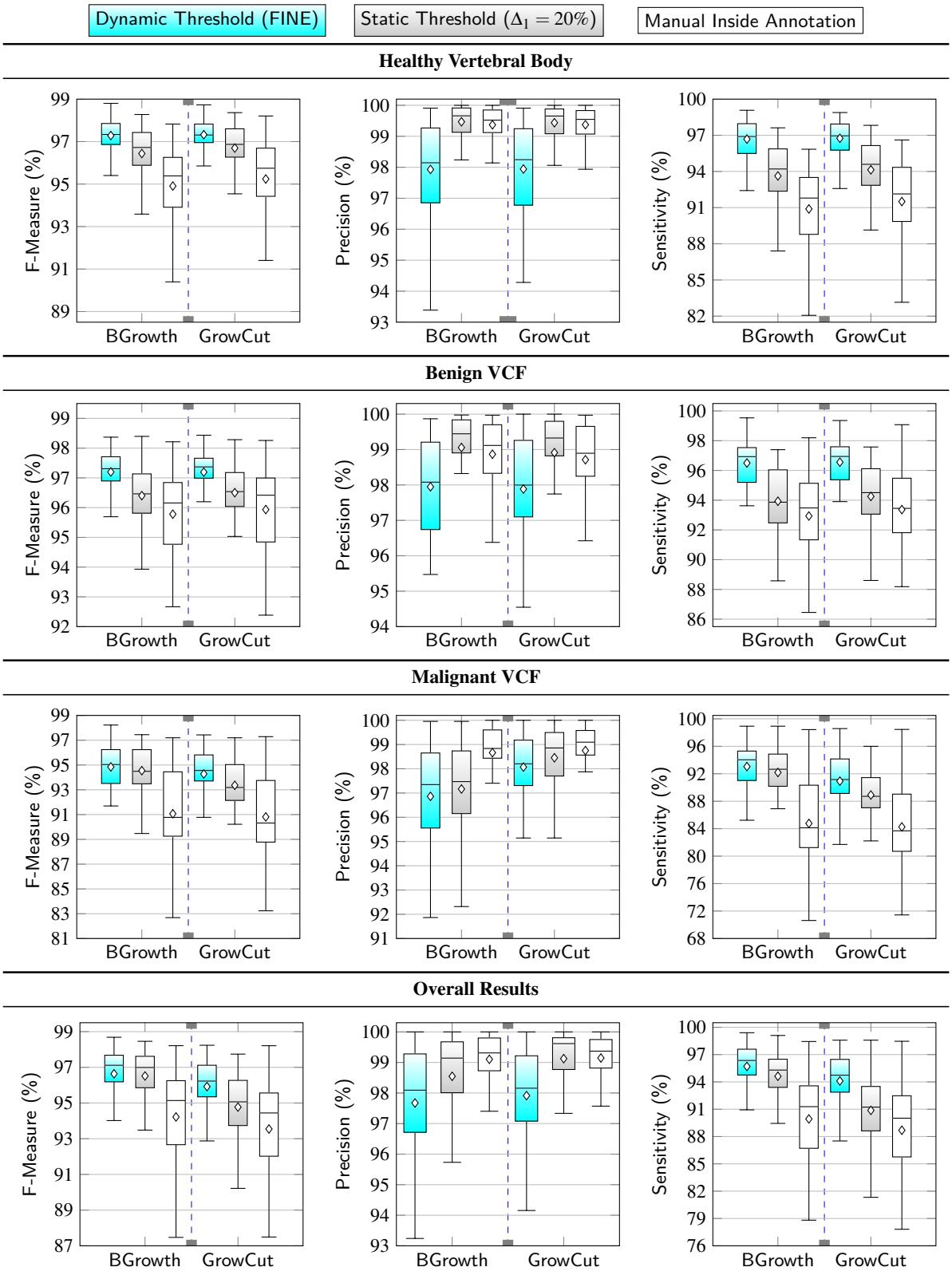
A.4.2 Qualitative Analysis

Figure 54 and Figure 55 depict the segmentation results for each segmentation method for a healthy vertebral body and a malignant VCF, respectively. All in all, FINE presented the best adherence to the borders of the vertebral bodies in comparison to the manual annotation and the static threshold ($\Delta = 20\%$).

Although the manual annotation leads to higher Precision, in general, the segmentation algorithms do not grow much, leaving a lot of the vertebral body content unsegmented (lower Sensitivity). Likewise, for the static threshold, a Δ value too high would not work on irregular shapes, as it happens most frequently in malignant and benign compression fractures. A Δ too low would not generate enough inside annotation and, therefore, does not contribute to a better segmentation. Therefore, defining a fixed threshold depends on the segmentation task, which includes manually analyzing the shape of the vertebral body for each image.

Unsegmented vertebral body content might lead to issues when analyzing the shape and texture of the region. Predominantly, the manual inside annotation and the static threshold segment mostly the bright or homogeneous parts of the vertebral body. In general, the malignant

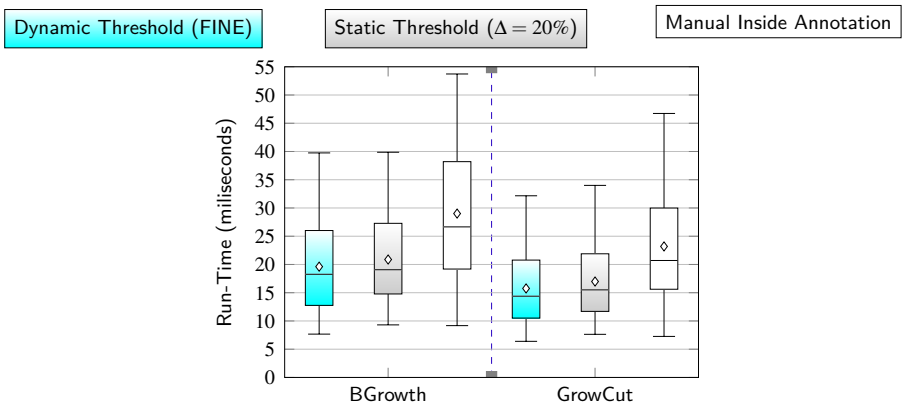
Figure 52 – Data distribution of the results.



Source: Adapted from Ramos *et al.* (2020).

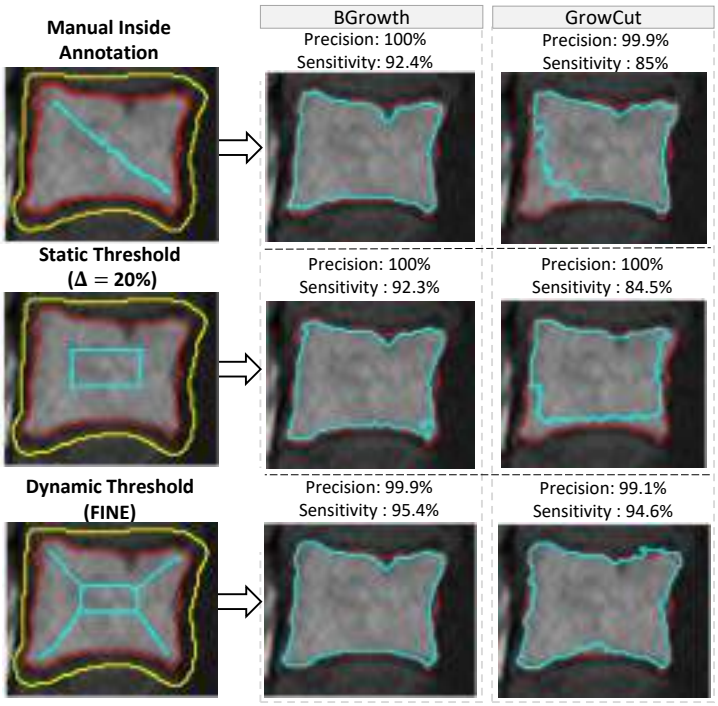
and benign VCFs present irregular shape and different intensities within the same vertebral body. Therefore, a segmentation method that achieves a higher Sensitivity (segments most of the

Figure 53 – Run-Time overall results comparison.



Source: Adapted from Ramos *et al.* (2020).

Figure 54 – Segmentation results for a healthy vertebral body. Ground-truth in red, outside (manual) annotation in yellow, and inside annotation (on the left) and resulting segmentation result (on the right) in cyan.



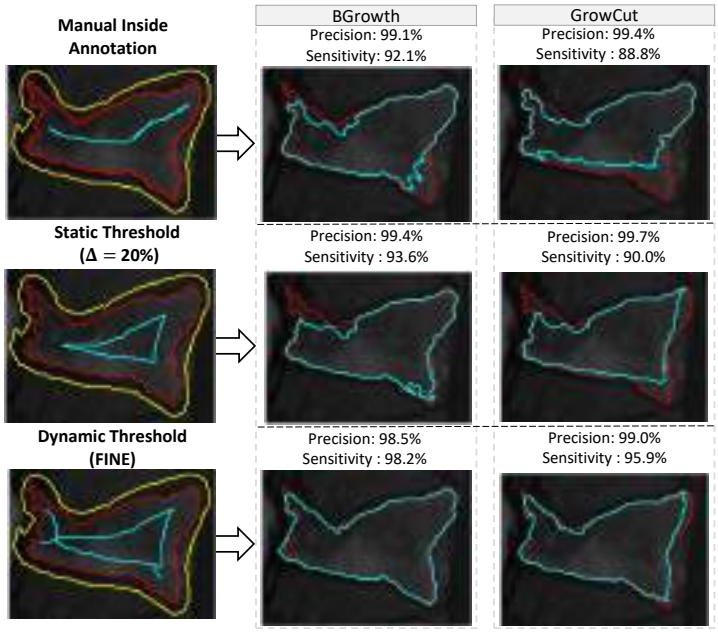
Source: Adapted from Ramos *et al.* (2020).

vertebral body content) is attractive in this case.

A.4.3 Statistical Validation

For all measures, the Kolmogorov-Smirnov test rejected the null hypothesis that the data follows a normal distribution. Accordingly, we employed the Wilcoxon test to evaluate the statistical significance of our proposal in comparison to the manual annotations and the static

Figure 55 – Segmentation results for a malignant VCF. Ground-truth in red, outside (manual) annotation in yellow, and inside annotation (on the left) and resulting segmentation result (on the right) in cyan.



Source: Adapted from Ramos *et al.* (2020).

threshold ($\Delta = 20\%$). Table 18 reports the Wilcoxon test results. Other methods outperformed FINE in terms of Precision values. Nonetheless, we were able to improve the estimation of interior regions in all settings regarding F-Measure, Sensitivity, and Run-Time measures. The obtained results emphasize that FINE can speed up the inside annotation of vertebral bodies and further improve the quality of the segmentation.

Table 18 – Wilcoxon's test result.

Segmentation Technique	FINE × Manual Annotation				FINE × Static Threshold			
	F-Measure	Precision	Sensitivity	Run-Time	F-Measure	Precision	Sensitivity	Run-Time
BGrowth	✔	✗	✔	✔	✔	✗	✔	✔
GrowCut	✔	✗	✔	✔	✔	✗	✔	✔

Source: Adapted from Ramos *et al.* (2020).

Note – ✔ means significantly better results for FINE and ✗ means the opposite.

A.5 Final considerations

In this chapter, aimed at minimizing the time spent on manual annotation, we proposed Fast INside Estimation (FINE), which finds the best thresholds values to determine the inside annotation. The experimental results with a representative database showed that the segmentation methods presented higher average Sensitivity (96%) with FINE. In general, the malignant and Benign VCFs display irregular shape and different intensities within the same vertebral body. In this case, higher Sensitivity values are particularly crucial so that one could extract shape and texture features with higher certainty.

ANNEX

A

SOURCE CODES

In this Annex, we present the most relevant source codes of our methods. First, in Section A.1, we present the source codes for BGrowth method, introduced in Chapter 3. Next, in Section A.2, we show the source codes for 3DBGrowth method, which we detailed in Chapter 4. Finally, in Section A.3, we report the source codes for FastCleverSeg and EANIS. We introduced both approaches in Chapter 6.

We reiterate that, in order for our methods to achieve better result, we recommend the following settings:

- The image I must be a 256 gray-scale image (conversion can be performed using Equation 2.2);
- The labels matrix L must present integer values corresponding to regions in the image. For example, 1 foreground, 0 undefined, and 2 background. The method also works with more than 2 regions.

Every C mex function will require the headers in Source code 1.

Source code 1 – Header for every Matlab C mex function in this Annex.

```
1: #include <math.h>
2: #include <matrix.h>
3: #include <mex.h>
4: #include <cstdlib>
5: #include <cstdint>
6: #include <vector>
7: /* Keep compatibility with earlier versions of ML */
8: #ifndef MWSIZE_MAX
9: typedef int mwSize;
10: typedef int mwIndex;
11: typedef int mwSignedIndex;
12: #if (defined(_LP64)||defined(_WIN64))&&!defined(MX_COMPAT_32)
13: /* Currently 2^48 based on hardware limitations */
```

```
14: # define MWSIZE_MAX    281474976710655UL
15: # define MWINDEX_MAX   281474976710655UL
16: # define MWSINDEX_MAX  281474976710655L
17: # define MWSINDEX_MIN -281474976710655L
18: #else
19: # define MWSIZE_MAX    2147483647UL
20: # define MWINDEX_MAX   2147483647UL
21: # define MWSINDEX_MAX  2147483647L
22: # define MWSINDEX_MIN -2147483647L
23: #endif
24: #define MWSIZE_MIN    0UL
25: #define MWINDEX_MIN   0UL
26: #endif
27: using namespace std;
```

A.1 Balanced Growth – BGrowth

We show Matlab code for BGrowth method in Source code 2.

Source code 2 – BGrowth Matlab code.

```
1: function [L, W, its] = BGrowth(I, L)
2:     I = double(I); si = size(I); sl = size(L);
3:     assert(numel(unique(L))==3,'labels must 2, 0 or 1');
4:     assert(all(sl(1:2)==si(1:2)),'labels/image size error');
5:     % 8-neighbors
6:     Nx = [-1,1,0,0,-1,-1,1,1]; Ny = [0,0,-1,1,1,-1,1,-1];
7:     gMax = max(I(:)); [dimx, dimy] = size(I);
8:     W(dimx, dimy) = 0; W(L ~= 0) = 1.0; W(L == 0) = 0;
9:     converged = false; its = 0;
10:    while (~converged)
11:        its = its + 1; converged = true;
12:        for i=2:dimx-1
13:            for j=2:dimy-1
14:                for n=1:8
15:                    in = i + Nx(n); jn = j + Ny(n);
16:                    gDiff = 1 - abs(I(i,j) - I(in,jn))/gMax;
17:                    str = W(in, jn)*gDiff;
18:                    if ( (str - W(i, j)) > 0.01)
19:                       L(i, j) = L(in,jn); W(i,j) =  (str+W(i,j))/2;
20:                       converged = false;
21:                    end
22:                end
23:            end
24:        end
25:    end
26: end
```

A.2 Volumetric Balanced Growth – 3DBGrowth

We show Matlab C mex function for 3DBGrowth in Source code 3.

Source code 3 – 3DBGrowth Matlab C mex function.

```
1: void mexFunction(int nlhs,mxArray *plhs[],int nrhs,const mxArray *prhs[]){
2:     /*** Check inputs ***/
3:     if(nrhs != 2) mexErrMsgTxt("Must have Image and Labels as arguments.");
4:     if(nlhs > 2) mexErrMsgTxt("Too many output arguments.");
5:     if(mxIsComplex(prhs[0]) || mxGetNumberOfDimensions(prhs[0]) < 2
6:             || mxGetNumberOfDimensions(prhs[0]) > 3
7:             || mxIsSparse(prhs[0]) || !mxIsUint8(prhs[0]))
8:         mexErrMsgTxt("Image must be 2D or 3D unsigned integer (8 bits).");
9:     if(mxIsComplex(prhs[1]) || mxGetNumberOfDimensions(prhs[1]) < 2
10:             || mxGetNumberOfDimensions(prhs[1]) > 3
11:             || !mxIsUint8(prhs[1]))
12:         mexErrMsgTxt("Labels must be 2D or 3D unsigned integer (8 bits).");
13:      // declare variables
14:     mxArray *strens_m, *labels_m, *labelsn_m, *I_m, *mxTheta;
15:     mwSize numdims;
16:     const mwSize *dims, *dims2;
17:     uint8_t *labels, *I;
18:     long i,j,k,m, idxq, idxp, its = 0, neigh[26], idx = 0;
19:     float *strens, *strensn, MAX_ITS = 999, maxC = 256, C = 0, theta, g;
20:     bool converged;
21:     // gets image dimensions
22:     numdims = mxGetNumberOfDimensions(prhs[0]);
23:     dims = mxGetDimensions(prhs[0]);
24:     dims2 = mxGetDimensions(prhs[1]);
25:     if(!(dims[0] == dims2[0] && dims[1] == dims2[1] && dims[2] == dims2[2]))
26:         mexErrMsgTxt("Labels and Image must be the same size.");
27:     int dimx, dimy, dimz, dimxy, dimxyz, colors;
28:     dimx = (int)dims[0]; dimy = (int)dims[1]; dimz = (int)dims[2];
29:     dimxy = dimx*dimy;
30:     if (numdims == 2)  dimz = 1; // not 3D matrix
31:     dimxyz = dimxy*dimz;
32:     if(dimx < 3 || dimy < 3) // Matrix too small for neighbors comparison
33:         mexErrMsgTxt("Matrix size has to be larger than 3x3x1.");
34:     // sets up pointers
35:     I_m = mxDuplicateArray(prhs[0]);//CREATE IMAGE's COPY
36:     labels_m = plhs[0] = mxDuplicateArray(prhs[1]);//CREATE LABELS' COPY
37:     // MALLOC SPACE ON MEMORY FOR THE STRENGTHS
38:     strens_m = mxCreateNumericArray(numdims, dims, mxSINGLE_CLASS, mxREAL);
39:     I = (uint8_t *) mxGetPr(I_m);
40:     labels = (uint8_t *) mxGetPr(labels_m);
41:     strens = (float *) mxGetPr(strens_m);
42:     // Define Neighbours
43:     for(k = -1; k <= 1; k++)
44:       for(i = -1; i <= 1; i++)
45:         for(j = -1; j <= 1; j++)
46:           if (!(i == 0 && j == 0 && k == 0))
47:              neigh[idx++] = i+j*(int)dims[0]+k*((int)dims[0]*(int)dims[1]);
48:     long index = 0;
49:      // Determine neighborhood size at each entry and initilize seeds
50:      vector<unsigned char> m_NBSIZE = vector<unsigned char>(dimxyz,0);
```

```
51:       for(i = 1; i < dimx - 1; i++)
52:          for(j = 1; j < dimy - 1; j++)
53:              for(k = 1; k < dimz - 1; k++) {
54:                  index = i + j*dimx + k*dimxy;
55:                  m_NBSIZE[index] = 26;
56:                  ((uint8_t)labels[index])<1?strens[index]=0:strens[index]=1;
57:              }
58:      // Start main loop
59:      converged = false;
60:      while(!converged){
61:          converged = true;
62:          for(k = 1; k < dimz-1; k++)
63:              for(j = 1; j < dimy-1; j++)
64:                  for(i = 1; i < dimx-1; i++) {
65:                      idxp = i + j*dimx  + k*dimxy;
66:                      if(labels[idxp] != 0) {
67:                          for(m=0; m<m_NBSIZE[idxp]; m++){//for every neighbor
68:                              idxq = idxp + neigh[m];
69:                              C = fabs(I[idxp] - I[idxq]);
70:                              g = 1-C/maxC;
71:                              theta = g*strens[idxp];
72:                              if( (theta - strens[idxq]) > 0.01 ){
73:                                  strens[idxq] = (theta + strens[idxq])/2;
74:                                  labels[idxq] = labels[idxp];
75:                                  converged = false;
76:                              }
77:                          }
78:                      }
79:                  }
80:      }
81:      mxDestroyArray(I_m);
82:      mxDestroyArray(strens_m);
83: }
```

A.3 Fast Clever Segmentation – FCleverSeg

We show Matlab C mex function for FastCleverSeg in Source code 4, how we estimated the inside annotation in Section A.3.1, and detail EANIS source codes in Section A.3.2.

Source code 4 – FastCleverSeg Matlab C mex function.

```
1: void mexFunction(int nlhs,mxArray *plhs[],int nrhs,const mxArray *prhs[]){
2:      /*** Check inputs ***/
3:      if(nrhs < 2) mexErrMsgTxt("Must get Image and Labels as arguments.");
4:      if(nlhs > 2) mexErrMsgTxt("Too many output arguments.");
5:      if(mxIsComplex(prhs[0]) || mxGetNumberOfDimensions(prhs[0]) < 2
6:              || mxGetNumberOfDimensions(prhs[0]) > 3
7:              || mxIsSparse(prhs[0]) || !mxIsUint8(prhs[0]))
8:          mexErrMsgTxt("Image must be a 2D or 3D unsigned integer (8 bits).");
9:      if(mxIsComplex(prhs[1]) || mxGetNumberOfDimensions(prhs[1]) < 2
10:             || mxGetNumberOfDimensions(prhs[1]) > 3
11:             || !mxIsUint8(prhs[1]))
12:         mexErrMsgTxt("Labels must be 2D or 3D unsigned integer (8 bits).");
```

```
13:     // Declare variables
14:     mxArray *strens_m, *labels_m, *I_m;
15:     mwSize numdims;
16:     const mwSize *dims, *dims2;
17:     uint8_t *labels, *I;
18:     long i,j,k,m, idxq, idxp, its = 0, neigh[26], idx = 0;
19:     float *strens, MAX_ITS = 999, maxC = 255, C = 0, g, *theta;
20:     bool converged;
21:     // Gets image dimensions
22:     numdims = mxGetNumberOfDimensions(prhs[0]);
23:     dims = mxGetDimensions(prhs[0]);
24:     dims2 = mxGetDimensions(prhs[1]);
25:     if(!(dims[0] == dims2[0] && dims[1] == dims2[1] && dims[2] == dims2[2]))
26:         mexErrMsgTxt("Labels and Image must be the same size.");
27:     int dimx, dimy, dimz, dimxy, dimxyz;
28:     dimx = (int)dims[0]; dimy = (int)dims[1]; dimz = (int)dims[2]; dimxy =
       dimx*dimy;
29:     if (numdims == 2) dimz = 1; // not 3D matrix
30:     dimxyz = dimxy*dimz;
31:     if(dimx < 3 || dimy < 3) // Matrix too small for neighbors comparison
32:         mexErrMsgTxt("Matrix size has to be larger than 3x3x1.");
33:     mxArray *mxTheta = mxDuplicateArray(prhs[2]);
34:     theta = (float *)  mxGetPr(mxTheta);;
35:     float t = *theta;
36:     I_m = mxDuplicateArray(prhs[0]);//CREATE IMAGE's COPY
37:     labels_m = plhs[0] = mxDuplicateArray(prhs[1]);//CREATE LABELS' COPY
38:     // MALLOC SPACE ON MEMORY FOR THE STRENGTHS
39:     strens_m = plhs[1] = mxCreateNumericArray(numdims, dims, mxSINGLE_CLASS,
        mxREAL);
40:     I = (uint8_t *) mxGetPr(I_m);
41:     labels = (uint8_t *) mxGetPr(labels_m);
42:     strens = (float *) mxGetPr(strens_m);
43:     long index = 0;
44:      // Determine neighborhood size at each vertice and Initilize seeds
45:     for (k = -1; k <= 1; k++)
46:       for (i = -1; i <= 1; i++)
47:         for (j = -1; j <= 1; j++)
48:           if (!(i == 0 && j == 0 && k == 0))
49:             neigh[idx++] = i+j*(int)dims[0]+k*((int)dims[0]*(int)dims[1]);
50:      // Determine neighborhood size at each vertice and Initilize seeds
51:      vector<unsigned char> m_NBSIZE = vector<unsigned char>(dimxyz, 0);
52:      for(i = 1; i < dimx - 1; i++)
53:          for(j = 1; j < dimy - 1; j++)
54:              for(k = 1; k < dimz - 1; k++) {
55:                  index = i + j*dimx + k*dimxy;
56:                  m_NBSIZE[index] = 26;
57:                  ((uint8_t)labels[index])<1?strens[index]=0:strens[index]=1;
58:              }
59:     vector<bool> visited = vector<bool>(dimxyz, false);
60:     //start main loop
61:     converged = false;
62:     double diff = 0, s = 0.0;
63:     while(!converged){
64:         converged = true;
65:         for(k = 1; k < dimz-1; k++)
66:             for(j = 1; j < dimy-1; j++)
67:                 for(i = 1; i < dimx-1; i++) {
```

```
68:                         idxp = i + j*dimx  + k*dimxy;
69:                         if(labels[idxp] != 0 && !visited[idxp]) {
70:                             visited[idxp] = true;
71:                             for(m = 0; m <  m_NBSIZE[idxp]; m++) {
72:                                 idxq = idxp + neigh[m];
73:                                 C = maxC - fabs(I[idxp] - I[idxq]);
74:                                 g = C/maxC;
75:                                 s = g*strens[idxp];
76:                                 diff = s-strens[idxq];
77:                                 if(diff > t) {
78:                                    visited[idxq] = false;
79:                                    strens[idxq]=(s+strens[idxq]+strens[idxp])/3;
80:                                    labels[idxq] = labels[idxp];
81:                                    converged = false;
82:                                 }
83:                             }
84:                         }
85:                     }
86:     }
87:     mxDestroyArray(I_m);
88: }
```

A.3.1 Estimation of the inside annotation

We present Matlab code for estimation of the inside annotation in Source code 5. We used those codes for FastCleverSeg validation in Chapter 6. Note that we consider labels with 1 values as foreground and exterior annotations were performed on the first, middle, and last slice (i.e. cropped VOI with annotations).

Source code 5 – Matlab code for the estimation of the inside annotation.

```
1: function ANT =  EstimateInteriorAnnotation(ANT)
2:     tam = size(ANT, 3);
3:     meio = round(tam/2);
4:     ANT(:,:,2:meio-1) = 0;
5:     ANT(:,:,meio+1:end-1) = 0;
6:     pos = find(squeeze(sum(sum(ANT==1))) > 0);
7:     se = strel( 'disk', 3);
8:     AN(size(ANT,1), size(ANT,2)) = 0;
9:     for i=1:numel(pos)
10:         clear aux
11:         aux = squeeze(ANT(:,:,pos(i)));
12:         AN(:) = 0;
13:         clear filledOutside filledOutside totalIns
14:         filledOutside = imfill(aux > 0, 'holes');
15:           totalOut = sum(filledOutside(:));
16:             totalIns = sum(filledOutside(:));
17:             while totalIns/totalOut > 0.2
18:                 backup = filledOutside;
19:                 filledOutside = imerode(filledOutside , se);
20:                 totalIns = sum(filledOutside(:));
21:                 if totalIns < 5
22:                     filledOutside = backup;
```

```
23:                     break;
24:                 end
25:             end
26:             AN(filledOutside > 0) = 1;
27:             AN(aux > 0) = 2;
28:             ANT(:,:,pos(i)) = AN;
29:     end
30:     ANT(:, :, 2) = ANT(:, :, 1);
31:     ANT(:, :, end-1) = ANT(:, :, end);
32: end
```

A.3.2 Estimation of Annotation on Intermediary slices – EANIS

We show Matlab code for EANIS in Source code 6.

The function **bresenham_line3d()** was acquired from <https://github.com/vistalab/vistasoft/blob/master/external/NIfTI_Shen/bresenham_line3d.m>.

Source code 6 – EANIS Matlab Code

```
1: function [AN3D, rt] = FillNonAnnotatedSlices(I, AN3D)
2:     slicesANT = find(squeeze(sum(sum(AN3D == 1))) > 1);
3:     tic
4:     for idx=1:length(slicesANT)
5:         AN3D = rasterLines(AN3D,slicesANT(idx),slicesANT(idx+1),1,1);
6:     end
7:     rt = toc;
8:     for idx=1:slicesANT(end)
9:         AN3D(:, 2, idx) = 2;
10:        AN3D(2, :, idx) = 2;
11:        AN3D(end-2, :, idx) = 2;
12:        AN3D(:, end-2, idx) = 2;
13:    end
14: end
15: function AN3D = rasterLines(AN3D,pos1,pos2,valor,v2)
16:     BB = bwboundaries(AN3D(:, :, pos1) == valor);
17:     BB2 = bwboundaries(AN3D(:, :, pos2) == valor);
18:     if ~isempty(BB) && ~isempty(BB2)
19:         for ix=1:length(BB)
20:             clear vector
21:             vector = BB{ix, :};
22:             [vals idx] = sortrows(vector, [1 2]);
23:             B = [vector(idx(1):end,:); vector(1:idx(1)-1,:)];
24:             zcoord(size(B,1)) = 0;
25:             zcoord(:) = pos1;
26:             for iy=1:length(BB2)
27:                 clear vector2 B2 idx2
28:                 vector2 = BB2{iy, :};
29:                 [vals idx2] = sortrows(vector2, [1 2]);
30:                 B2 = [vector2(idx2(1):end,:); vector2(1:idx2(1)-1,:)];
31:                 clear zcoord
32:                 zcoord(size(B2,1)) = 0;
33:                 zcoord(:) = pos2;
34:                 if (length(B) < length(B2))
```

```
35:                         cont = 1;
36:                         for j=1:length(B)
37:                             P1 = [B(j, 1) B(j, 2) pos1];
38:                             P2 = [B2(j, 1) B2(j, 2) pos2];
39:                             [X,Y,Z] = bresenham_line3d(P1, P2);
40:                             cont = cont + 1;
41:                             for i=2:length(X)-1
42:                                 AN3D(X(i),Y(i),Z(i)) = v2;
43:                             end
44:                         end
45:                         c = 1;
46:                         for j=length(B)+1:length(B2)
47:                             P1 = [B(c, 1) B(c, 2) pos1];
48:                             P2 = [B2(j, 1) B2(j, 2) pos2];
49:                             [X,Y,Z] = bresenham_line3d(P1, P2);
50:                             for i=2:length(X)-1
51:                                 AN3D(X(i),Y(i),Z(i)) = v2;
52:                             end
53:                             if c == size(B,1)
54:                                 c = 1;
55:                             else
56:                                 c = c + 1;
57:                             end
58:                         end
59:                     else
60:                         cont = 1;
61:                         for j=1:length(B2)
62:                             P1 = [B2(j, 1) B2(j, 2) pos2];
63:                             P2 = [B(j, 1) B(j, 2) pos1];
64:                             [X,Y,Z] = bresenham_line3d(P1, P2);
65:                             cont = cont + 1;
66:                             for i=2:length(X)-1
67:                                 AN3D(X(i),Y(i),Z(i)) = v2;
68:                             end
69:                         end
70:                          c = 1;
71:                         for j=length(B2)+1:length(B)
72:                             P1 = [B(j, 1) B(j, 2) pos1];
73:                             P2 = [B2(c, 1) B2(c, 2) pos2];
74:                             [X,Y,Z] = bresenham_line3d(P1, P2);
75:                             cont = cont + 1;
76:                             for i=2:length(X)-1
77:                                 AN3D(X(i),Y(i),Z(i)) = v2;
78:                             end
79:                             if c == size(B2,1)
80:                                 c = 1;
81:                             else
82:                                 c = c + 1;
83:                             end
84:                         end
85:                     end
86:                 end
87:             end
88:         else
89:             disp('Empty annotation')
90:         end
91: end
```


ICMC
USP
SÃO CARLOS