



UNIVERSITAT DE BARCELONA

Final Degree Project

Biomedical Engineering Degree

**“ Pipeline Design for Automated
Vertebrae Segmentation for Paediatric
Scoliosis Applications “**

Barcelona, 5th June, 2024

Author: Marta Maldonado Montes

Director/s: Sandra Bernaus Tomé

Gerard Marturià i Navarro

Tutor: Manel Puig i Vidal

Abstract

Scoliosis is abnormal lateral curvature of the spine exceeding the 10 degrees, being adolescent idiopathic scoliosis (AIS) the most prevalent type, identified in children between 10 and 16 years of age. Depending on the severity of the curvature, there are different treatment methods, with surgical treatment being the one of choice for cases with a curvature greater than 45 degrees. However, it presents numerous perioperative complications, therefore, the surgical planning of these cases is used to give support to the surgeon when performing the surgery. Efforts have been made to develop Artificial Intelligence (AI) based models for automatic vertebrae and spine segmentation, as this task is time-consuming and repetitive, and is used in numerous medical applications, including surgical planning. The aim of this project, started from scratch at the Hospital Sant Joan de Déu, is to design and develop a pipeline that integrates automatic segmentation methods to be used in AIS cases. Different automatic segmentation methods have been studied, among which one has been chosen and is used in the present project. Each of the steps to be carried out to implement the pipeline proposed with the method chosen are indicated, explained and detailed, including data collection, setup of the working environment, use of image pre-processing techniques, and the execution of the model. Additionally, this model has been re-trained with data from adolescents with scoliosis condition from the Hospital to improve its performance. Results show that the re-training of the model allows to obtain a better segmentation of the spine and vertebrae from the input CT images.

Key words: Automatic segmentation, Computed tomography images, Adolescent idiopathic scoliosis (AIS), Surgery planning

Acknowledgements

First of all, I would like to thank my directors, Sandra Bernaus and Gerard Marturià, for the effort and the hours spent to help me in the development of this project. In addition, for giving me the opportunity to start a project with them at the Hospital Sant Joan de Déu.

I would also like to thank my tutor, Manel Puig, for his indications and recommendations during the development of the project.

I would like to thank my family for the unconditional support and encouragement they have given me throughout the entire project.

List of figures

Figure 1: Spine (A) and vertebra (B) anatomy [5].	11
Figure 2: Cobb angle measurement [14].	12
Figure 3: On the left, a preoperative spine model with a 3D printed guide. On the right, an intraoperative photograph demonstrating the positioning of the guide [21].	14
Figure 4: U-Net architecture [33].	16
Figure 5: Input CT image (A), final vertebrae segmentation superimposed on the input CT scan (B) [35].	17
Figure 6: Graphical scheme of the Lessmann et al. [37] proposed model.	18
Figure 7: Structure of PaDBNs model [38].	19
Figure 8: Spine CT scan, vertebrae segmentation and labelling, vertebrae volume rendering, performed with SpineR technology, indicated from left to right [42].	21
Figure 9: Sagittal image of the segmentation mask of 'sub-verse502' patient in VerSe dataset. .	26
Figure 10: Instructions to install the MONAI Label Module in 3D Slicer.	28
Figure 11: Diagram with the programs, file formats and a brief description for each of the steps performed in the pre-processing, execution and results stages.	29
Figure 12: T9 vertebra mask split into the main vertebra mask, in green, and the rib mask, in purple.	30
Figure 13: On the left, tab where it is indicated how to import the STLs. On the right, it is displayed the Data module with the different segmentations grouped under the same.	31
Figure 14: Default ColorTable, on the left. Modified ColorTable with 24 labels, one for each vertebra, on the right.	32
Figure 15: On the left, Indications for applying the ColorTable to the segmentations and exporting the latter in NIFTI format. On the right, the resulting mask with the corresponding intensity for each vertebra.	33
Figure 16: Initial steps to start with MONAI Label module in 3D Slicer, with the server running. .	35
Figure 17: Instructions to re-train a model in MONAI Label module.	36
Figure 18: Work breakdown structure of the Project.	42
Figure 19: Gantt chart of the Project.	44

List of tables

Table 1: Scoliosis classification according to Cobb angle.....	13
Table 2: Summary of the pre-processing steps with a corresponding illustrative image.	34
Table 3: Performance of the localization_vertebra model in terms of Dice Scores.	37
Table 4: Performance of the re-trained localization_vertebra model in terms of Dice Scores.....	38
Table 5: Tasks duration definition.	43
Table 6: SWOT matrix analysis of the Project.....	45
Table 7: Hardware, software and human resources costs of the Project.	46

Table of contents

Abstract	1
Acknowledgements	2
List of figures	3
List of tables	4
1. Introduction	7
1.1. Motivation and aim of the Project	7
1.2. Origin of the Project.....	8
1.3. Objectives and scope of the Project	8
1.4. Structure and methodology	9
1.5. Limitations	10
2. Background	11
2.1. General Concepts.....	11
2.1.1. Vertebral Column and Vertebrae anatomy.....	11
2.1.2. Scoliosis condition	12
2.1.3. Clinical Assessment, Diagnosis and Curve Measurement.....	12
2.1.4. AIS Surgery Preoperative Planning	13
2.1.5. Medical Imaging and Image Segmentation.....	14
2.1.6. DICOM, NIfTI and STL file formats	15
2.1.7. Terminal, Local Environment, CPU and GPU	15
2.2. State of the art.....	16
2.2.1. U-Net Model for Automated Spine and Vertebrae Segmentation.....	16
2.2.2. Fully Convolutional Network for Vertebrae Localization and Segmentation	17
2.2.3. Deep Learning Approach for Automatic CT Vertebra Segmentation.....	18
2.3. State of the situation.....	19
3. Market analysis	20
3.1. Sectors targeted by Automatic Image Segmentation.....	20
3.2. Players in Medical Imaging.....	20
3.3. Bonescreen software.....	20
3.4. Future market perspectives	21
4. Conception Engineering	22
4.1. Python	22
4.2. ITK-SNAP	22

4.3.	Seg3D	23
4.4.	MONAI Label and 3D Slicer.....	23
4.5.	Selection of the proposed solution.....	24
5.	Detail Engineering.....	25
5.1.	Data collection	25
5.1.1.	Hospital Sant Joan de Déu dataset.....	25
5.1.2.	VerSe: Large Scale Vertebrae Segmentation Challenge	25
5.2.	Working Environment Setup.....	26
5.2.1.	System requirements	26
5.2.2.	Local Environment Creation and Setup	27
5.2.3.	Installing 3D Slicer and MONAI Label Module	28
5.3.	CT image and Vertebrae Segmentation mask pre-processing	28
5.3.1.	Vertebrae Segmentation mask pre-processing.....	29
5.3.2.	CT image pre-processing.....	34
5.4.	Execution.....	35
5.4.1.	localization_vertebra model	35
5.4.2.	Re-train the localization_vertebra model.....	36
5.5.	Results	37
5.5.1.	localization_vertebra model of the Radiology application	37
5.5.2.	Re-trained localization_vertebra model.....	38
5.6.	Discussion	40
6.	Execution Schedule	42
6.1.	Work Breakdown Structure (WBS)	42
6.2.	GANTT Chart.....	42
7.	Technical Feasibility	45
8.	Economic Feasibility	46
9.	Regulation and Legal aspects.....	47
10.	Conclusions and Future lines	48
11.	References	50
Annexes		55
	Annex A. Erosion pre-processing technique Python code.....	55
	Annex B. Dice score evaluation metric Python code	56

1. Introduction

1.1. Motivation and aim of the Project

The spine, which keeps humans upright and connects the different parts of our skeleton – head, shoulders, ribcage, arms, pelvis, and legs – together, is the central support structure of the human being. Its natural S-shaped curvature makes it stable: it helps keeping the balance when being in an upright position, acts like a shock absorber when human walks, and protects the vertebrae from fractures [1]. In addition, the bones that make up the spine also protect the spinal cord, which runs through the vertebral foramen of the vertebrae from the brainstem to the lower back, and whose role is to conduct information from the brain to the periphery and vice versa [2]. Vertebral column conditions include traumatic or osteoporotic fractures, herniated disc, spinal deformities (scoliosis, lordosis and kyphosis), craniocervical junction instability, spinal cord damage (myelopathy), nerve compression or impingement (radiculopathy) and wear and tear of joint cartilage (facet arthropathy) [3]. These conditions cause pain and, in many cases, can limit movement, so appropriate treatment is necessary, which varies depending on the disease, but sometimes includes orthopaedic braces and surgery.

This project focuses on scoliosis disease, which is an abnormal lateral curvature of the spine exceeding the 10 degrees and commonly combined with a rotation of the vertebrae, affecting 2-4% of the global population, and being more prevalent among girls [4]. This can be roughly translated to 162-324 million people, considering that the current world population is about 8.1 billion people. More specifically, it will be focused on Adolescent Idiopathic Scoliosis (AIS) condition, which is the type of scoliosis most prevalent worldwide, identified in children between 10 and 16 years of age. Scoliosis left untreated, it might lead to cardiopulmonary complications, causing shortness of breath and difficulty breathing. Also, to develop cardiovascular dysfunction, due to the decrease in space for the heart, causing pain and a reduction of the blood flow [5]. To prevent these disease-associated problems from developing, there are different treatment methods depending on the degree of the curvature. Individuals with curves of 10 to 25 degrees are monitored for surveillance, usually at 3, 6 or 12-month intervals. AIS whose curvature is between 25 and 40 degrees are candidates for bracing, to keep the curve from progressing as the child grows. And those adolescents who are skeletally immature with curves over 40 to 45 degrees and those mature patients with a measurement of 50 degrees or greater, are candidates for surgical treatment. However, the latter presents numerous perioperative complications such as need for blood transfusions, incorrect implant placement, bar rupture, pulmonary, vascular or neurological injuries after surgery. Therefore, preoperative planning is essential in AIS surgery to perform the procedure with greater precision and diligence, allowing reducing the risks associated with it by forcing the surgeon to study the patient's characteristics, history, and vertebral deformity, and to identify anatomical details and key technical points of the intervention. Moreover, to avoid unnecessary prolongation of operative time [6].

In addition, the integration of artificial intelligence (AI) into medical imaging has guided an era of transformation in healthcare. AI, and in particular deep learning algorithms, has demonstrated remarkable capabilities in extracting insights from medical images. It has opened new possibilities in image segmentation and quantification. By employing sophisticated algorithms, AI can

accurately delineate structures of interest within medical images, such as tumours, blood vessels, bone, or cells. Therefore, this segmentation capability is invaluable in treatment planning, as it enables clinicians to precisely target areas for intervention, optimize surgical procedures, and facilitate personalized treatment plans [7].

In view of the above, the importance of a safe, accurate and optimised surgical planning procedure for the surgical treatment of AIS is the main motivation for the development of the present project. Specifically, the aim of the project is to optimize the vertebrae segmentation procedure during surgical planning of adolescent idiopathic scoliosis cases by using AI models.

1.2. Origin of the Project

This project was born out of the need to automate the segmentation task during the surgical planning procedure of adolescent idiopathic scoliosis cases, at the Hospital Sant Joan de Déu. This project is the beginning of an innovative path towards automatic segmentation of vertebrae and spine of AIS cases, which currently does not exist in the Hospital Sant Joan de Déu. Therefore, the present project does not follow any other project, it has been started from scratch, so there is a lack of previous experience and knowledge of the topic on the part of the team.

As indicated, this project has its origin in the Hospital Sant Joan de Déu and has been conducted in collaboration with the Paediatric Computational Imaging Center (PeCIC) and the 3D4H Unit of the Innovation Department.

1.3. Objectives and scope of the Project

The primary objective of this project is to design a pipeline that integrates solutions for automatic segmentation, with CT images, to be applied to the use case of adolescent idiopathic scoliosis for surgical planning. The pipeline will include the collection, preparation and pre-processing stages of the data to be used, the steps to setup the working environment and the execution processes of the proposed solution for automatic segmentation. Each of these steps and stages will be explained and detailed rigorously so that this pipeline can be reproduced and used by other users, who may or may not be familiar with the concepts, methods and techniques described in the present project. To accomplish this main goal, the secondary objectives are in the following order:

- i. Conduct a literature search on adolescent idiopathic scoliosis, and on the available automatic segmentation algorithms.
- ii. Study different software solutions for use in the segmentation task and to justify the solution chosen for the development of this project.
- iii. Study the chosen software solution in depth using different research tools such as web pages, online tutorials, discussion forums or recorded webinars available on the Internet.
- iv. Define the technical requirements and conditions necessary to setup the working environment where the chosen software will be used, and the corresponding pre-trained models.
- v. Test the chosen model with data from paediatric patients of the Hospital Sant Joan de Déu.
- vi. Re-train the model with data from paediatric patients of the Hospital Sant Joan de Déu.

- vii. Evaluate the performance of the chosen segmentation model and the re-trained model using the Dice score metric.

From the segmentations of each of the vertebrae, the planning engineer designs a customized surgical guide for each of them, so that the surgeon has a guide to ensure the correct placement of the pedicle screws in the intervention. However, this project does not detail how the design of these surgical guides is performed, therefore, the impact that the optimization of the segmentation process has on this stage of the indicated surgical planning is not evaluated.

1.4. Structure and methodology

For the achievement of the objectives previously described, this project integrates different stages, which are specified below.

Firstly, a theoretical study on the vertebral column and vertebrae anatomy, the scoliosis condition and on the AIS surgical planning has been carried out. In addition, the main concepts related to medical imaging and image segmentation, and file formats are detailed. Moreover, literature research on the state of the art of automatic vertebrae and spine segmentation algorithms has been carried out. All previous information can be found in section 2. In the next section, the market analysis for automatic segmentation is studied. In relation to this, section 4 exposes and explores possible software and methods that can be developed to perform the task of segmentation automatically. The details of all the steps and stages implemented to develop the chosen solution, including the steps to setup the working environment where it will be developed, as well as all the procedures, methods and techniques used in the different processes are indicated in Section 5. The different tasks to be performed in the project and their execution time, the technical feasibility of the proposed solution and the economic viability of the same, are defined in sections 6, 7 and 8, respectively. Finally, the legislation applicable to the project developed is indicated in Section 9.

Being this a retrospective project, the CT images used are from 18 adolescent paediatric patients at Hospital Sant Joan de Déu who have already been treated for AIS by surgical intervention. In this scenario, the images used in this project are from the project with code APG001, approved by the Drug Research Ethics Committee (CEIm), a committee attached to the Fundació de Recerca Sant Joan de Déu [8].

The tasks involved in the review, correction and performance of the pre-processing technique of the segmentations of each of the cases for their subsequent use have been carried out in the 3D4H Unit department. On the other hand, the tasks focused on the development, setup and execution of the proposed automatic segmentation software have been carried out in the PeCIC Center, at the Hospital Sant Joan de Déu. The evaluation stage of the results obtained has been carried out at home.

Furthermore, I have had the opportunity to observe from inside the operating theatre a surgical intervention of a case of adolescent idiopathic scoliosis, performed by surgeon Alejandro Peiró García, being able to observe how he used the 3D printed model of the spine and the patient's personalised printed surgical guides for the placement of the pedicle screws.

1.5. Limitations

The scope of the project has been defined considering three main limitations, defined below. Firstly, the temporal limitation, given that the Final Degree Project has an execution period of one semester and a corresponding delivery date must be met. In relation to the time constraint, the search for models that allow automatic segmentations from which to start the elaboration of the project has taken a long time, which has been a limitation for the development of the project.

Secondly, the size of the dataset used. This is due to two factors: the first is that surgical planning for adolescent idiopathic scoliosis cases has been carried out since 2023, therefore, the number of patients for whom vertebrae segmentations are available is small. In addition, no open-source datasets of CT medical images of the spine, with or without scoliosis, of paediatric adolescents are available. The second factor is that all those paediatric cases in the Hospital Sant Joan de Déu dataset in which the curvature of the scoliosis had a Cobb angle (see section 2.1.3) greater than 55 degrees were discarded from the study due to the vertebrae anatomy of these cases being very different from those in the medical images used to pre-train the model. Therefore, this second limitation not only restricts the generalizability of the findings but also the training of the proposed model.

There have also been limitations with the availability of the infrastructures to be used. As indicated in section 1.4, all the tasks, excluding the evaluation stage of the results, were carried out in the corresponding departments at the Hospital Sant Joan de Déu, using the computers provided by the directors, where they work, as the different tasks involved the use of licensed software that only those computers had. Therefore, we had to organise, which was sometimes difficult to coincide due to incompatibility of schedules, which delayed the execution of some tasks.

2. Background

This chapter focuses on explaining the concepts that will allow a better understanding of the terms that will appear throughout the report. In addition, it will detail where the technology stands and what has been done so far.

2.1. General Concepts

This section presents various concepts for a better understanding of the terms used throughout the report for the development of the project.

2.1.1. Vertebral Column and Vertebrae anatomy

The vertebral column provides flexible support for the body's physical structure, head and trunk, transmits the weight of the upper body to the lower limbs, and supports the nervous system, enabling movement and sensation. Its shape, viewed from the side, is not straight, it has four slight natural curvatures that distribute the effort required for movement in daily activity. These are: cervical, thoracic, lumbar, and sacral curves (*Figure 1*) [9].

In humans, the spine is composed of 33 vertebrae that include 7 cervical, 12 thoracic, 5 lumbar, 5 sacral, and 4 coccygeal. The spine protects the spinal cord, located at the vertebral foramen, a central lumen within each vertebral body, and the spinal nerves, that emerge from the main cord at each vertebral level [10]. Between adjacent vertebrae there are intervertebral disks, that are the cartilaginous structures responsible for providing cushion and absorb the shock produced by movement. Each vertebra features a body, pedicle, transverse process, lamina, and spinous process (*Figure 1*). The first, is the thick anterior portion that supports most of the weight. Extending posteriorly from the lateral aspects of the vertebral body are the pedicles, that form most of the vertebral arch's sides [5]. From the vertebral arch, they extend the transverse process, and the most posterior part of the arch are the laminae, that meet at the spinous process.

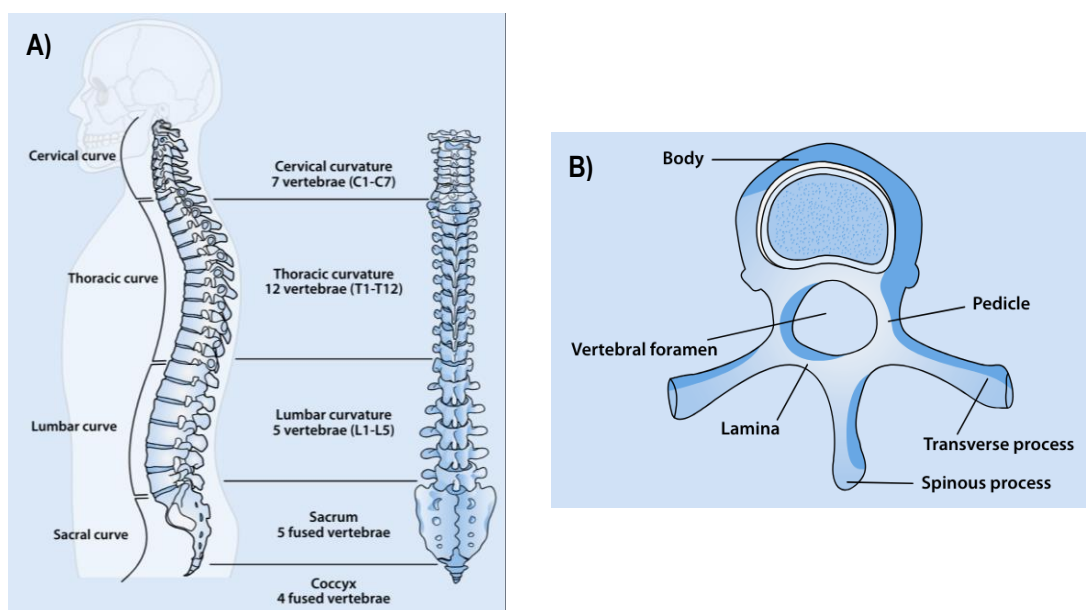


Figure 1: Spine (A) and vertebra (B) anatomy [5].

2.1.2. Scoliosis condition

Scoliosis is an abnormal lateral curvature of the spine that affects the morphology of the trunk, and consequently, the vertebral column viewed from the front has a C- or S-shaped curve. This condition can be caused by a neurological or neuromuscular disease, or by a congenital deformation. However, when the cause is unidentifiable it is known as idiopathic. Scoliosis can appear at any age, although it is most often identified at an early age, typically at 10 to 16 years [4]. Below, the different types of scoliosis are defined:

- Idiopathic scoliosis. It is the most common form, but the curvature's origin is unknown. Moreover, studies suggest a genetic component in idiopathic scoliosis, which affects girls more than it does boys in all age groups [5].
- Congenital scoliosis. It is present at birth, and results from a vertebral malformation, where bones are asymmetrical, and the spine may be partially formed [11].
- Neuromuscular scoliosis. It results from insufficient muscle and tendon stabilization, caused by a neurological or degenerative muscular disorder such as muscular atrophy or dystrophy.

Idiopathic scoliosis can be subdivided according to the age of onset as [12]:

- Infantile idiopathic scoliosis. Is diagnosed in children between birth and 3 years of age.
- Juvenile idiopathic scoliosis. Diagnosed between 3 to 9 years of age, and accounts for 12 to 20 percent of all paediatric cases.
- Adolescent idiopathic scoliosis. Diagnosed in children between the ages of 10 and 18 years old. 80% of all paediatric cases of idiopathic scoliosis fall into this category.

2.1.3. Clinical Assessment, Diagnosis and Curve Measurement

Symptoms associated with scoliosis may include uneven shoulders, pain in the back, one shoulder blade to be more prominent or visible than the other, one hip appearing higher than the other, or one side of the rib cage being higher than the other when bending forward [13]. In the initial evaluation of scoliosis, both the patient history and a physical examination, as well as radiographs are crucial. The latter is indicated when during the physical examination alterations in spinal alignment have been identified. On the imaging scans, such as X-rays, CT scans and MRIs, which have been taken on patients in a standing position, a physician measures the degree of spinal curvature to establish a diagnosis of scoliosis. The most common quantification of scoliosis is the Cobb angle, and its measurement involves estimating the angle between the two tangents of the upper and lower end-vertebrae of the curve deformity, respectively (*Figure 2*) [14].

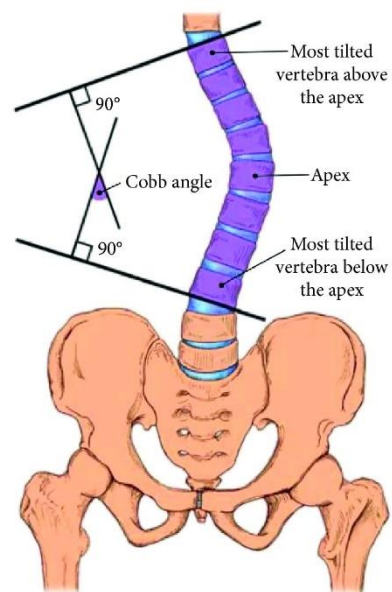


Figure 2: Cobb angle measurement [14].

Table 1 shows the classification of scoliosis according to the magnitude of the curve using the Cobb angle. Scoliosis occurs when the scoliotic curve in the coronal plane is greater than 10 degrees.

Table 1: Scoliosis classification according to Cobb angle.

Cobb angle (degrees)	Classification
0° - 10°	Spinal curve
10° - 20°	Mild scoliosis
20° - 40°	Moderate scoliosis
40° - 45°	Severe scoliosis
> 45°	Very severe

In a diagnosis of adolescent idiopathic scoliosis, the spine shows a curvature in the coronal plane combined with reduced kyphosis in the sagittal plane and rotation in the axial plane [15]. To describe and classify AIS and recommend treatment, it is also used the Lenke Classification, which describes the type of curve. This classification system considers the coronal curve (1-6), the sagittal deformity (-, N, or +), and the lumbar spine modifier (A, B, C) [16]. However, the formal education of Lenke Classification is beyond the scope of this section.

2.1.4. AIS Surgery Preoperative Planning

The decision to surgically treat an AIS is based on the type of curve, the proximity to skeletal maturity and the balance in the coronal and sagittal plane. Most authors place the magnitude of the curve at 50° for the indication for surgical treatment. This premise is deduced from the fact that curves greater than 45° have a tendency towards progression and that from 50° onwards they are cosmetically more evident [17]. For patients their curve magnitude is lower than 45°, bracing can be an effective means of controlling some form of early onset idiopathic scoliosis. However, bracing is intended to prevent progression, not to correct the curve.

The indications for surgical treatment are summarised below:

- Curve > 50 degrees.
- Curve > 40 in a skeletally immature patient.
- Progression despite conservative treatment.
- “Unacceptable” deformity.

As indicated at the beginning of the report, preoperative planning is essential in AIS surgery to determine the fusion levels and to perform the procedure with greater precision and diligence. The surgical planning engineer has at his disposal the clinical radiographs, more specifically, the CT scans taken from the patient in a standing position and with the arms raised. The doctor indicates which vertebrae need to be fused, and therefore, which vertebrae require customized surgical guides, to be designed by the engineer, to determine the corresponding screw trajectories. The use of patient-specific 3D printed guides for pedicle screw insertion has been demonstrated to increase the effectiveness of pedicle screw placement by more than 95% [18] [19].

The planning engineer works mainly with segmentations, which represent the different parts or regions that can be divided from an image, in terms of volume and shape (*Section 2.1.5*). First, with the segmentation of the entire spine. This is obtained from the CT image and with the use of a threshold tool to determine which Hounsfield units (HU), which is a relative quantitative measurement of radio density used by radiologists in the interpretation of CT images [20], and therefore, which body structures are to be considered in the segmentation. From this initial segmentation, the vertebrae are separated into different segmentations using a semi-automated tool, to allow working with every vertebra independently and facilitate a better visualization of every part of it for the correct placement of the guide and screws. This segmentation procedure is time-consuming for the engineer, for that reason, having an automatic segmentation tool to speed up this process will have an important improvement in surgical planning.

In the operating room, screw placement is achieved by mounting the guide onto the posterior elements of the respective vertebra level to be instrumented. This enables mechanical guidance of a drill bit that is placed through a drill sleeve that is inserted into the cannulated portion of the guide [21].



Figure 3: On the left, a preoperative spine model with a 3D printed guide. On the right, an intraoperative photograph demonstrating the positioning of the guide [21].

2.1.5. Medical Imaging and Image Segmentation

Medical Imaging is the use of imaging modalities and processes to represent the interior human body for clinical diagnosis and medical intervention by the visual representation of tissues and organs [22]. There are different types of medical imaging techniques which use different technologies. Among the most used techniques are X-ray, computed tomography (CT) and magnetic resonance imaging (MRI).

Digital medical imaging is an $f(x,y)$ function in a spatial coordinate-partitioned grey scale that can be represented by a matrix in which the intersection of each row and column identifies a single pixel within an image. The value of each pixel in the matrix identifies the grey level at that point (x,y) on a scale of integer values that represent black, white and shades of grey, being these the lowest value, the highest value and intermediate values, respectively [23]. Furthermore, CT images can be acquired volumetrically, in form of a volume of parallel spaced slices, so that a pixel in the

image represents a voxel, with a “height” equal to the thickness of the slice. This matrix of numerical values that represent the image, is what processing and analysis tools work with.

A key step in image processing and analysis is segmentation. It is essential for accurately delineating anatomical structures and pathological regions of interest (ROIs) from medical images, allowing clinicians to isolate and analyse specific areas, facilitating accurate measurements, quantitative analysis, and treatment planning. Image segmentation is particularly valuable in areas such as tumour delineation, organ volume estimation, tissue characterization and surgical guidance [24]. Medical image segmentation involves classifying each pixel in the image with the correct label so that pixels shearing the same label have certain characteristics.

2.1.6. DICOM, NIfTI and STL file formats

In medical imaging annotation, imaging standards and file formats play an important role. The extraction of ROIs from medical images can be performed from CT scans, X-rays, and MRI files. However, two of the most common medical imaging data formats are DICOM and NIfTI. Digital Imaging and Communications in Medicine (DICOM) is one of the most widely deployed healthcare messaging Standards in the world and defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use [25]. The Neuroimaging Informatics Technology Initiative (NIfTI) is a data format for the storage of medical images, focusing on Functional Magnetic Resonance Imaging (fMRI). The main difference between these two data formats is that DICOM image files and the associated data is made up of 2D layers, allowing to view different slices of an image, whereas NIfTI images and data are stored in a 3D format, to overcome spatial orientation challenges. Another file format commonly used are STL files, acronym for stereolithography, used for 3D printing and computer-aided design (CAD). A STL file is made up of a series of linked triangles that describe the surface geometry of a 3D model or object. The more complex the design, the more triangles used, and the higher the resolution [26].

2.1.7. Terminal, Local Environment, CPU and GPU

The terminal provides a command line interface (CLI) for accessing files and giving instruction to the system. When it is executed, it opens a text window that displays a welcome message and the ‘prompt’, which is a text string with the path to the current directory, initially the user’s home directory, ending with the character ‘>’. From this point, the user can give orders to the system by typing different commands [27].

Local environments are a type of Python environments that allow to create workspaces, where packages can be install without affecting other environments, and users can interact directly to accomplish specific tasks. A conda environment is a subtype of local environment, that is managed using the ‘conda’ package manager. To create a conda environment in the terminal, the command to be typed is the following: ‘conda create -- name <name-of-the-environment>’ [28].

A central processing unit (CPU) is a hardware component and the central computing unit of a server, which converts data into digital signals and perform mathematical operations on them. Therefore, the CPU is the main component that processes signals and makes computation possible [29]. On the other hand, a graphics processing unit (GPU) is composed of smaller and more

specialised cores that accelerate the execution of computationally intensive applications. While the CPU performs sequential serial processing, the GPU has a parallel architecture to solve multiple tasks at the same time [30].

2.2. State of the art

Over the years, spinal imaging analysis has received sustained attention from the medical imaging community. The use of artificial intelligence (AI), machine learning (ML) and deep learning (DL) in healthcare has driven advances in the management of spinal diseases, although it was from the 1980s onwards that image analysis became a significant root of AI in healthcare [31].

2.2.1. U-Net Model for Automated Spine and Vertebrae Segmentation

U-Net is the most popular convolutional neural network model for the segmentation of biomedical image. It consists of two parts: an encoder and a decoder. The former, is used to extract features from the given dataset, and the latter, is used to predict the segmented mask [32]. The figure below shows the standard architecture of U-Net.

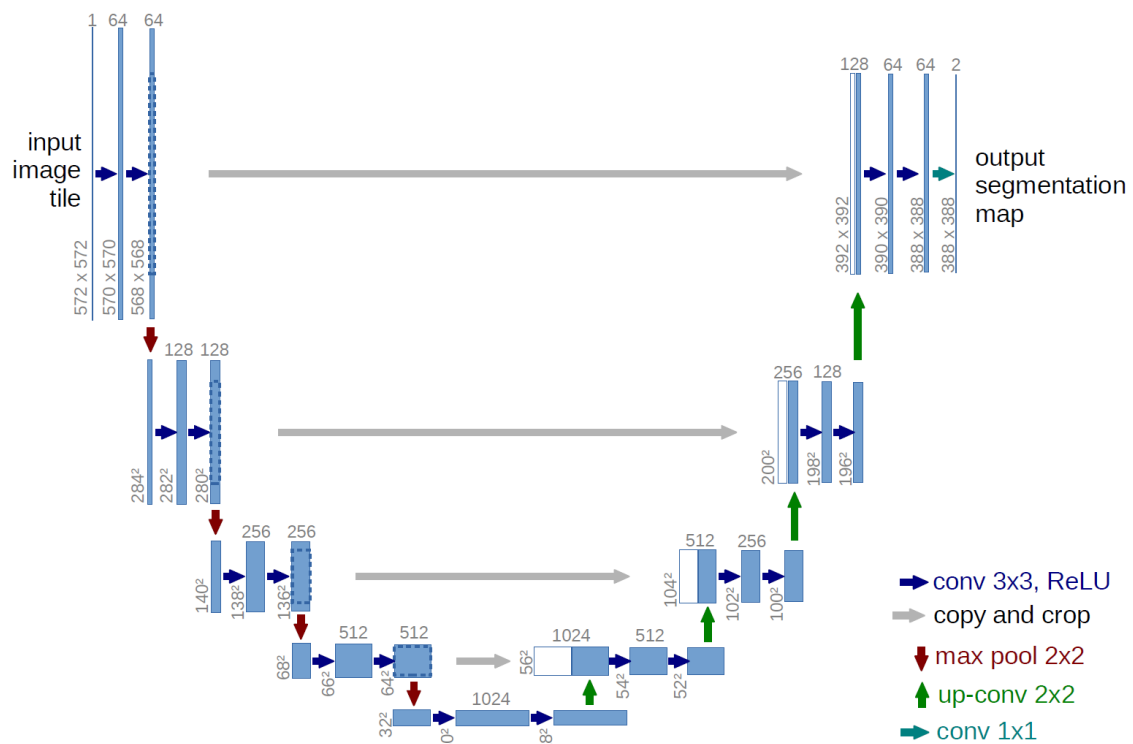


Figure 4: U-Net architecture [33].

The encoder, which is the first half in the architecture shown in *Figure 4*, reduces the size of the image using max-pooling layers, that extract the maximum of 2×2 neighboring features [32]. Therefore, it extracts relevant features from the incoming data. The decoder, the second half of the architecture, uses the extracted features from the encoder part and predicts a segmentation mask. Contrary to the first part, the decoder uses up-convolutional layers to upscale the images to a size closer to the original.

Saeed et al. [32] proposed a model, that uses CT images, for the automated segmentation of the spine based on a modified version of the standard U-Net architecture. In the proposed model, CHASPP layers are used after each max-pooling layer in the encoder part to enhance the performance of the machine learning model as these focus on objects in the dataset. Moreover, in each encoder part, residual blocks are inserted that extract deep features from the dataset and pass it to the decoder layers. The decoder part of this proposed model contains attention modules that focus on the regions of interest. For the evaluation of the model, the VerSe dataset (see section 5.1.2) was used. The model achieved 94.58% Dice score, coefficient that measures the similarity between two sets of data [34].

2.2.2. Fully Convolutional Network for Vertebrae Localization and Segmentation

Payer et al. [35] proposed a fully automatic three-step approach for vertebrae localization and segmentation. First, a variant of the U-Net is used to perform heatmap regression of the spinal centerline, which is the line passing through all vertebral centroids, to localize the predicted x and y coordinates of the spine. This variant of the U-Net performs average instead of max-pooling in the encoder part. Second, the SpatialConfiguration-Net (SC-Net) proposed in [36] is used to localize the vertebrae. It performs heatmap regression of the N target vertebrae, and for each of the N target, the network predicts all N output heatmap volumes. For each predicted heatmap volume, multiple local maxima are detected using a threshold. From this, the first and last vertebrae are determined by taking the maxima with the largest value that is closest to the top or bottom of the volume, respectively. The final predicted landmark ensures that consecutive vertebrae are not closer than 12.5 mm and farther away than 50 mm [35]. For the last step, vertebrae segmentation, it is used a U-net set up for binary segmentation to separate each vertebra individually from the background. Finally, the individual predictions of the vertebrae are transformed and resampled back to their position in the original volume.

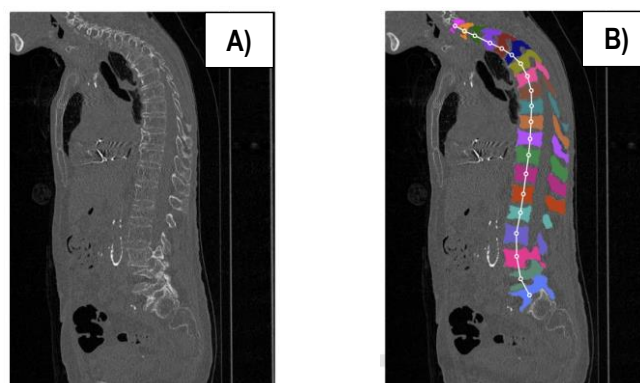


Figure 5: Input CT image (A), final vertebrae segmentation superimposed on the input CT scan (B) [35].

Lessmann et al. [37] proposed an iterative fully convolutional neural network (FCN) for automatic vertebra segmentation and identification. The model approaches vertebra segmentation as an instance segmentation problem, therefore, all vertebrae are detected as instances within the same class. However, contrary to generic instance segmentation, in this model the instances are not independent of each other, these are known to be close to each other in the image. This knowledge

is incorporated into the algorithm. The method applies a patch-based vertebra-by-vertebra segmentation approach, following either a top-down or bottom-up scheme, in which the image is analysed in patches large enough to contain at least one vertebra. The network segments a single vertebra in this patch and the anatomical knowledge, incorporated previously to the algorithm, that the following vertebra must be located in close proximity is used to reposition the patch for segmentation of the following vertebra [37]. Moreover, the network combines a memory component that retains information about already segmented vertebrae and uses it to segment only the following not yet segmented vertebra. A binary label for each voxel in an image patch is predicted and indicates whether the voxel belongs to the current instance or not. The proposed model obtained a Dice score of 96.3% for vertebra segmentation. *Figure 6* shows a graphical scheme of the model.

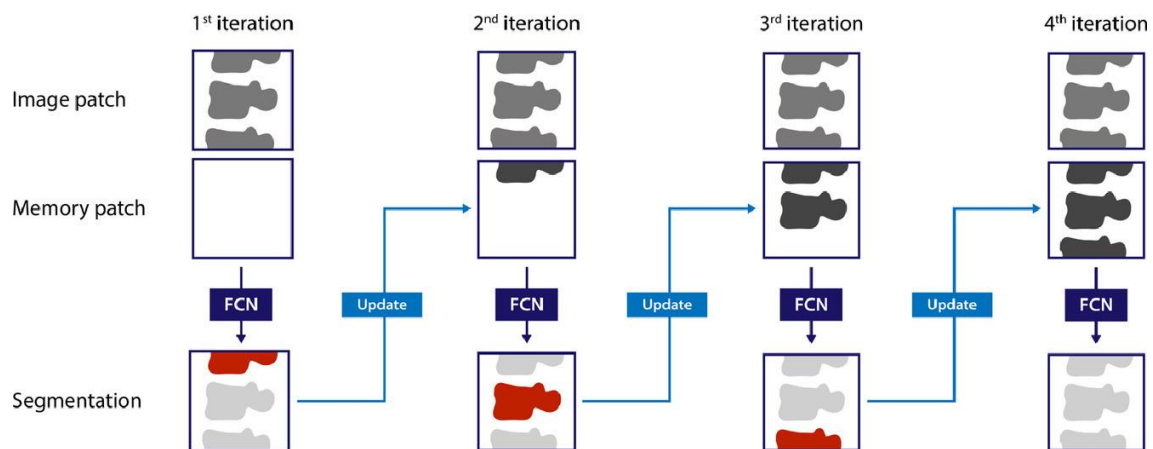


Figure 6: Graphical scheme of the Lessmann et al. [37] proposed model.

2.2.3. Deep Learning Approach for Automatic CT Vertebra Segmentation

Qadri et al. [38] proposed a deep learning approach named patch-based deep belief networks (PaDBNs), that automatically selects the features from CT image patches and measures the differences between classes. Patch-based methods label each voxel in the target images by comparing image patches centered on the voxel, with patches from an atlas library, and assign the most probable label based on the closest and best match patch from the library. The deep belief network (DBN) is a deep learning model composed of Restricted Boltzmann Machines (RBMs), which are the primary units of DBN. It is trained in an unsupervised manner, it learns from unlabeled data, to extract features. The structure of the PaDBNs proposed model is composed of an input layer, two hidden layers and an output layer (*Figure 7*). The input layer and the first hidden layer construct the first RBM and the first and second hidden layers the second RBM. The two hidden layers are the ones used to extract the features from the vertebrae and the two nodes in the output layer enable the network to select the most relevant features for representation of each image patch [38]. To test the trained model, image patches that have not been seen by the model are introduced. First, it predicts the image class patch and generates a probability map by adding all distributions. From this, it reconstructs the image, obtaining the final segmentation image of the vertebra. This method achieved an accuracy of 93.3% and a sensitivity of 91.1%.

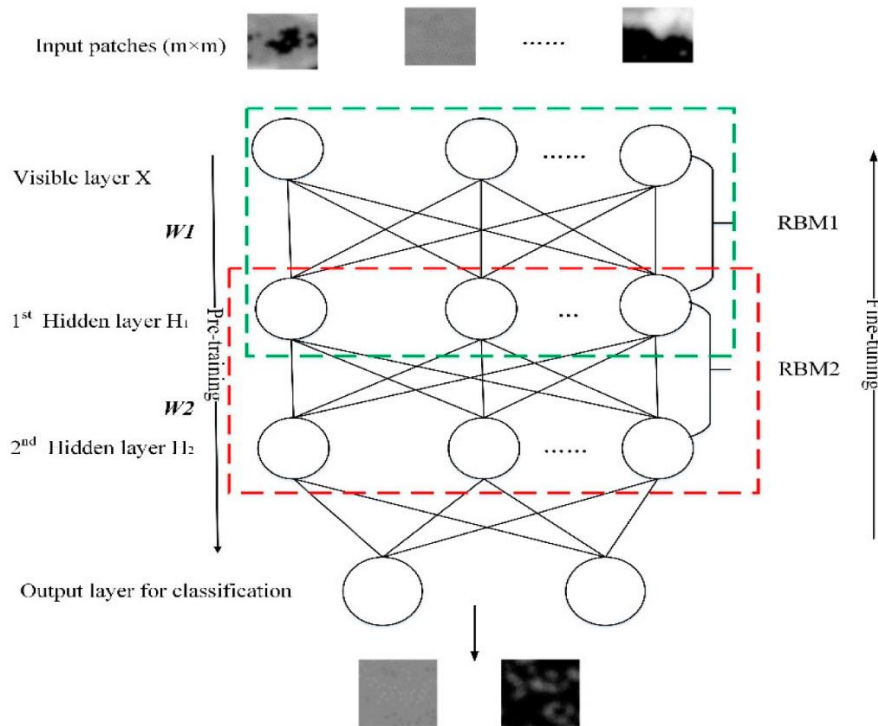


Figure 7: Structure of PaDBNs model [38].

2.3. State of the situation

Regarding the previous section, it can be seen that the use of DL in spine imaging studies is the focus of attention for tasks related to the localization and segmentation of the vertebral column, as well as of each of the vertebral bodies that compose it. This is due to the fact that all subsequent analyses, such as the detection and grading of fractures and the analysis of spinal shape, curvature, and deformity such as scoliosis, rely on these initial tasks [39].

Surgical planning for scoliosis treatment allows having an estimate of the length of the implants, the density of the pedicle screws, and the strategic point of entrance and orientation to place them. Consequently, it has been observed to improve intraoperative precision and the speed with which the surgical procedure is performed. The shorter operative time has also been associated with a lower complication rate in spinal deformity surgery in paediatric patients [6].

With an incidence of screw misplacement as high as 15 - 30% [19] in spine deformity surgery, the need for technical tools to work with individual vertebrae to plan pedicle screw trajectories for each vertebra has become acute. Therefore, considering that the first task to be performed and from which the following planning steps are developed is vertebrae segmentation, having models that allow this task to be carried out accurately and quickly is crucial.

3. Market analysis

The market for automatic image segmentation in healthcare is experiencing an important growth driven by the implementation of technologies that integrate AI models, specifically, those based on deep learning. The following sections aim to indicate which sectors are targeted by automatic segmentation of medical images, the main multinationals present in this market, the future perspectives of this market, and a product launched on the market focused exclusively on the task of vertebrae segmentation.

3.1. Sectors targeted by Automatic Image Segmentation

From the medical perspective, manual segmentation can be time-consuming, repetitive, subjective, and prone to inter-observer variability [40]. Therefore, the use of automated segmentation techniques with integrated DL-based approaches have gained significant attention in different specialities, beyond radiology, which is the most widespread.

Among all the applications of medical image segmentation in different medical specialities, some are the following: aiding in segmentation for radiation therapy planning, assisting in surgical planning and navigation, facilitating organ segmentation for volumetric analysis and diagnosis, supporting cardiac image analysis, assisting in neuroimaging for brain structure delineation and lesion detection, and enabling precise image-guided interventions [40]. As can be observed, there are several medical specialities that can benefit from the use of automatic segmentation and the advantages that this AI-based technique offers. Therefore, the main specialities targeted by automatic segmentation are: radiotherapy, cardiology, neurology, also oncology and orthopaedics.

3.2. Players in Medical Imaging

North America is the leading region in the global AI in medical imaging market, due to the high adoption of advanced healthcare technologies, growing prevalence of chronic diseases, and important investments in research and development. In this region, is the United States the largest market, with leading players such as GE HealthCare, IBM Watson Health and Philips Healthcare [41]. However, the Asia-Pacific region, due to the growing geriatric population and rising healthcare expenditure in countries such as China, India and Japan, is expected to witness important growth in the market of AI medical imaging [41]. In this region, Hitachi and Fujifilm Holding Corporation are the major players.

3.3. Bonescreen software

In this section, it is presented a medical imaging software, Bonescreen, focused exclusively to assist radiologists and clinicians in assessing bone health. One of its products offered is SpineR. It is an AI-based technology that segments and labels all vertebrae, identifies subregions and special points of interest in CT scans [42], which are tasks commonly needed in automated surgery planning or 3D printing. Moreover, SpineR can identify if there are transitional vertebrae, which are a congenital vertebral anomaly of the L5-S1 junction in the spine. If this alteration is not correctly identified, it may contribute to incorrect identification of a vertebral segment, leading to wrong-level

spine surgery [43]. *Figure 8* shows the different stages that can be achieved with SpineR, from the initial CT scan.

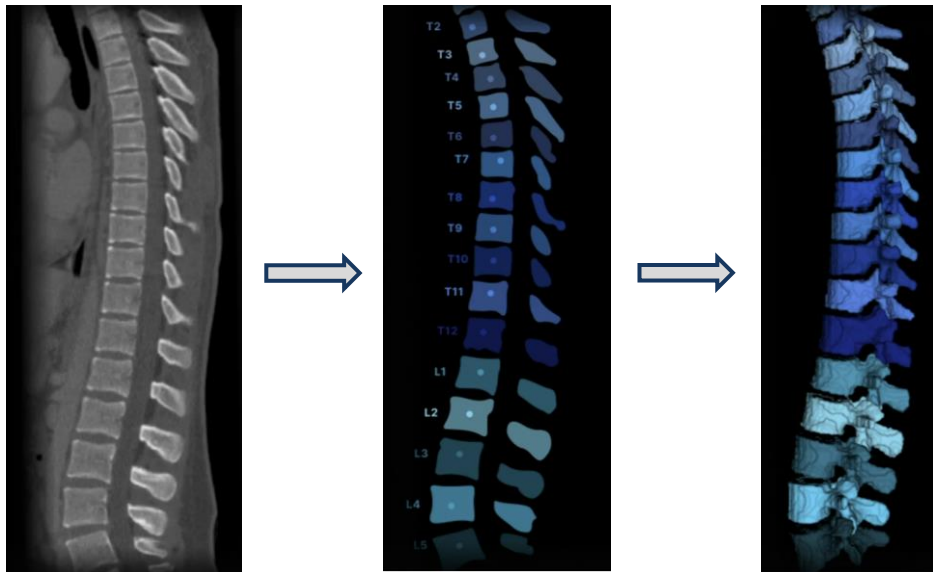


Figure 8: Spine CT scan, vertebrae segmentation and labelling, vertebrae volume rendering, performed with SpineR technology, indicated from left to right [42].

3.4. Future market perspectives

Although numerous advances have been made in the development of models for automatic image segmentation, their implementation and market launch face several challenges. These include the need for large annotated datasets, generalization across different imaging modalities, robustness to variations in image quality, and the interpretability of deep learning models [40].

The diagnostic imaging market is expected to register a Compound Annual Growth Rate (CAGR) of nearly 6.1% between 2023 and 2029 [44]. This increase is important for the further development of automatic segmentation models as the use of DL-based approaches for diagnosing spinal conditions is the most widely used clinical application in spine imaging, enhancing clinical decision-making, improving patient outcomes, and facilitating personalized medicine and treatment planning [45] [40].

4. Conception Engineering

As indicated previously, segmentation in radiology is crucial for many applications such as visualization, radiotherapy or 3D printing. However, it is often considered tedious and uninteresting. With the advances in AI, semi-automatic segmentation and interactive segmentation tools can be used. Hereafter, the possible solutions to tackle the segmentation automatization and optimization process for use in CT imaging are specified in this section. However, it is important to note that one of the requirements for this search is that these tools are open-source and free.

4.1. Python

A possible solution to tackle the segmentation automatization process for use with CT images consists of developing code using the Python programming language. One approach consists of using one of the models explained in the Conception Engineering section. However, the code of the networks described in the different sub-sections of each corresponding model is not available in most cases. Other possibility is to create a Python code from scratch, in which different machine learning libraries such as PyTorch, Keras or TensorFlow, and different DL architectures, such as U-Net or CNN, would be used in its implementation. Therefore, this proposed solution implies a high level of knowledge in code programming, as well as considerable amount of time to carry it out, which is a limitation present in the project. Furthermore, a third approach considered is the adaptation of an available open automatic segmentation model, for use in paediatrics cases with scoliosis condition. This approach presents the same drawback as described above. The previously solutions indicated, although they allow automating the segmentation process, require a high computational level, as well as a high knowledge of code programming, in addition to being time-consuming. The latter conflicts with one of the stated limitations of this project, the temporal constraint. Therefore, other solutions that include tools with the possibility to be applied or modified for automatic segmentation are searched and studied.

4.2. ITK-SNAP

ITK-SNAP is a free software application used to segment structures in 3D medical images. The images that can be processed by this software are MRI, CT, and PET images. The application provides semi-automatic segmentation with the use of active contour methods as well as tools for manual delineation of anatomical structures present in the image. Segmentations can take place in the three orthogonal cut planes, these are shown in separated view windows in the application, and the results can be visualized in a 3D volume rendering. Moreover, the software supports different input formats, among these are DICOM and NIFTI, which are the most widely used. The stages to implement the semi-automatic segmentation in ITK-SNAP are four: selection of the region of interest in the image, pre-segmentation, initialization, and evolution [46]. A ROI containing the structure of interest is defined, and it is on this that all subsequent segmentation operations will be applied. Then, the anatomical image is transformed to a new image called 'speed image' which has values that range from -1 to 1, with the aim to make the speed values closed to 1 inside of the structure of interest, and close to -1 for the rest of the image. In the initialization stage, several seeds are placed inside of this structure of interest, which will be grown in order to form the corresponding segmentation. Finally, in the last stage, the seeds evolve, and expand over the

regions with positive speed values, and on the contrary, they contract over the regions with negative speed values. This final segmentation generated can be displayed as a three-dimensional rendering for better visualization of its shape and can be saved in a variety of image formats, two of which are NIfTI and NRRD. However, although the steps to perform the semi-automatic segmentation are well defined, the software does not provide the option to generate a model to reproduce the procedure for other cases, so it would not be possible to automate the process directly with this program.

4.3. Seg3D

Seg3D is a free volume segmentation and processing tool developed by the National Institutes of Health (NIH) Center for Integrative Biomedical Computing. It combines a manual segmentation interface with image processing and segmentation algorithms from the Insight Toolkit (ITK) [47]. This last, ITK, is a cross-platform open-source library that offers a comprehensive set of software tools for image analysis. Seg3D can process CT and MRI images, and the common biomedical image formats that it supports are DICOM, NIfTI and NRRD. This software provides a semi-automatic segmentation tool combined with manual contouring of the anatomical structures in the images. Its implementation consists of placing seeds in the region of interest, which the segmentation tool will use to find all connected tissues or structures of the same grey scale range, and segment them. These seed points can be placed in different slices of the image. The final segmentation obtained can be rendered in a 3D representation. If parts of the tissues or structures of interest are missing in the segmentation generated, more seed points can be added, and the segmentation tool can be run again. Finally, the correct segmentation can be exported to DICOM and NRRD formats. Nevertheless, Seg3D does not offer the option to generate or train a model to reproduce this procedure for other cases, consequently, it is not possible to automate this segmentation process with this program.

4.4. MONAI Label and 3D Slicer

MONAI Label is an intelligent image labelling and learning tool that uses AI assistance to address labelling and segmentation tasks in order to reduce the time and effort of annotating new datasets [48]. This tool is built under Project MONAI, an open-source, freely available collaborative frameworks for annotating, building, training, deploying, and optimizing AI workflows in healthcare, started initially by NVIDIA [49]. These frameworks provide high-quality, user-friendly, and reproducible software that facilitates easy integration. MONAI Label trains an AI model for a specific task, by utilizing user interactions, and continuously learns and updates that model as it receives additional annotated images. This tool is supported by several viewers, however, the platform used in the field of radiology is 3D Slicer. This platform is a free, open-source software for visualization, processing, segmentation, registration, and analysis of medical, biomedical, and other 3D images and meshes; for planning and for navigating image-guided procedures [50]. The 3D Slicer software offers different modules that can be installed for use on its platform, including the MONAI Label module. Therefore, to be able to work with the MONAI Label tool, it is necessary to have the 3D Slicer platform installed. The input image formats supported by this software are NIfTI and DICOM, and the most common images it can process are MRI and CT.

MONAI Label offers three different template applications that can be downloaded in the local

environment. One of them is the Radiology template that uses the CNN U-Net as the default network. This application, Radiology, contains example models to do both interactive and automated segmentation of radiological (3D) images. Among these pre-trained models are the segmentation Spleen model, the DeepEdit model and the Multistage Vertebra Segmentation model. The latter is a multistage approach for segmenting several structures on a CT image, and has two different stages, the spine localization model, to localize the spine as a single label, and the vertebra localization model, to segment the vertebrae. These two stages can be used together or independently [51]. Considering the objective of the project, the vertebra localization model would be the one chosen. This model has been pre-trained over the VerSe dataset (see section 5.1.2), which contains a total of 374 CT scans from 355 patients with a mean age of $\sim 59 (\pm 17)$ years [52]. The vertebra localization model has as input one channel for the CT image, and as output N channels representing the segmented vertebrae. This solution proposed is engaging as it allows the segmentation process to be automated.

4.5. Selection of the proposed solution

With the possible solutions previously explained and defined, the benefits and drawbacks of each of them are resumed in the following table.

	ADVANTAGES	DISADVANTAGES
Python Programming	Can be used with CT images. Allows to create a reproducible model for automatic the segmentation process.	Requires a high computational level and a high knowledge of code programming. Conflicts with the temporal limitation as it is a time-consuming solution.
ITK-SNAP	This software can process CT images. It is an open-source software. Has well defined literature.	Can not generate a model to reproduce the automatic segmentation process.
Seg3D	It can be used with CT images. It is an open-source software.	Does not offer the option to generate a model to make the automatic segmentation process reproducible.
MONAI Label and 3D Slicer	It has open-source frameworks and software. Has pre-trained models on vertebrae automatic segmentation. Can process CT images.	The available model is pre-trained with data of adult subjects with normal anatomy of the spine. Low literature on how to pre-train the available model.

Considering the different points expressed in the previous table for each of the solutions, it has been concluded to use the MONAI Label and 3D Slicer approach, as it is the one that allows to automate the segmentation process and make it reproducible, being within the temporal constraints of the project. Therefore, it is on the basis of this solution I have found and studied, that this project has been carried out, with the supervision of the directors.

5. Detail Engineering

In this section, the procedures carried out with the chosen solution, 3D Slicer and MONAI Label module, will be shown in the different corresponding subsections, which cover data collection, the setup of the working environment, the pre-processing methods to be applied on the medical images and segmentations, the execution of the pre-trained model in the Radiology application, the re-training of this model and the execution of this last.

5.1. Data collection

Image annotation is the practice of labelling images in a given dataset to train AI and machine learning models. Therefore, image annotation generates the training data that supervised AI models can learn from. Consequently, an error in the manual labels will be replicated in the model.

5.1.1. Hospital Sant Joan de Déu dataset

Hospital Sant Joan de Déu (HSJD) performs surgery to treat AIS since 2019. However, it was not until 2023 that these were performed with surgical guides customized for each patient. Therefore, the available dataset consists of 38 CT scans from paediatric patients with adolescent idiopathic scoliosis. However, those subjects with a curvature greater than 55° have been considered complex cases and have not been considered for the final dataset. Therefore, the latter consists of 18 CT scans. Nevertheless, in the final dataset, data from two patients with a high average spinal curvature ($>55^\circ$) have been left to evaluate how the model would perform with these complex cases. The dataset has been anonymised and approved for the intended purposes by the CEIm.

The dataset corresponds to data of patients who have already been treated for this pathology by surgery, therefore, the corresponding segmentation mask of the spine and vertebrae used for the surgical planning of these cases has already been done by the planning engineer in charge. However, to use the vertebra localization model in the Radiology application of MONAI Label and to evaluate the results with the highest accuracy, further pre-processing techniques will need to be applied to all CT images and segmentation masks in the HSJD dataset (see section 5.3). These final segmentation masks after pre-processing are considered the ground truths, for the evaluation of the performance of the model. These pre-processing steps will be carried out according to the different observations indicated in section 5.1.2.

5.1.2. VerSe: Large Scale Vertebrae Segmentation Challenge

The *Large Scale Vertebrae Segmentation Challenge* (VerSe) was organised in conjunction with the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) in 2019 and 2020, with a call for algorithms tackling the labelling and segmentation of vertebrae [52]. As indicated in section 4.4, the VerSe dataset consists of CT scans collected from 355 patients with a mean age of $\sim 59(\pm 17)$ years and with normal anatomy of the spine. This information is important to consider as it is the dataset used for pre-training the vertebra localization model, that will be used in this project with the HSJD dataset. After analysing the structure of the VerSe dataset, the following observations are extracted:

- It contains a sub-directory-based arrangement for each subject, and for each patient directory, there is a NIfTI file with a vertebrae segmentation mask, see *Figure 9*, that is what the model returns after its execution, and a NIfTI file with the CT image.
- The segmentation masks are of different spine coverages, and do not contain the ribs segmented, only the vertebrae.
- The segmentation masks have the different vertebrae labelled from 1 to 24, starting with the cervical C1 up to the lumbar L5. Moreover, it contains some cases where the patient has one more lumbar, L6, which is assigned with the label 25. This is resumed as follows:
 - Labels 1 to 7 correspond to the cervical spine (C1-C7)
 - Labels 8 to 19 correspond to the thoracic spine (T1-T12)
 - Labels 20 to 25 correspond to the lumbar spine (L1-L6)
- Each vertebra in the segmentation mask has a different level of intensity in the grey scale (*Figure 9*), corresponding to the label assigned to it. Therefore, the same vertebra in all segmentation masks in the VerSe dataset is labelled and assigned with the same label and intensity level, respectively.

All these observations will be considered for the pre-processing of the ground truths of the HSJD dataset, which will be used for the evaluation of the segmentation masks obtained with the vertebra localization model.

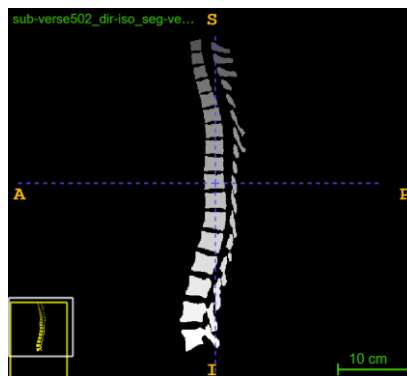


Figure 9: Sagittal image of the segmentation mask of 'sub-verse502' patient in VerSe dataset.

5.2. Working Environment Setup

This section details the different steps to be carried out to setup the environment where the programs and applications necessary for the development of this project must be installed and downloaded. Moreover, this section details how to install the MONAI Label module in 3D Slicer.

5.2.1. System requirements

Below are specified the technical requirements to be fulfilled for the correct functioning of the applications and programs, and in particular, for the correct functioning of the vertebra localization model in the MONAI Label module.

- MONAI Label supports Ubuntu and Windows operating systems (OS) with GPU/CUDA enabled [53]. Therefore, the device must have an NVIDIA graphics card.
- Have installed PyTorch and the CUDA Toolkit [54].
- Have a least 12 GB of GPU RAM [54].
- Have Conda installed, which can be downloaded from the Anaconda official website.
- Have python 3.8 or 3.9 version installed [53].

5.2.2. Local Environment Creation and Setup

The steps below define the commands to create a local environment on a Windows OS and install the applications and packages needed for the development of the project.

1. **Open the System Terminal** by searching in the OS search engine in the task bar the word *Terminal*. All the following steps will be performed on the terminal.

2. **Install MONAI from conda-forge.**

```
conda install -c conda-forge monai
```

3. **Create an Environment** with Python version 3.9, the packages Torch, Torchvision, Torchaudio of PyTorch with the specific CUDA versions indicated, and the CUDA toolkit version 11.3.

```
conda create -n <name-of-the-environment> -c pytorch python=3.9
thorch==1.7.1+cu110 torchvision==0.8.2+cu110 torchaudio==0.7.2
-f https://download.pytorch.org/whl/torch_stable.html
```

```
pip install cudatoolkit=11.3
```

Where `<name-of-the-environment>` is the name that the environment will have.

4. **Activate the Python Local Environment.**

```
conda activate <name-of-the-environment>
```

The next steps will be carried out in the environment that has been activated.

5. **Install MONAI Label.**

```
pip install monailabel
```

6. **Download the application Radiology from MONAI Label.**

```
monailabel apps --download --name radiology --output <app-
output-path>
```

Where `<app-output-path>` is the path where the folder *apps*, containing the Radiology app, will be downloaded.

7. Start the Server.

```
monailabel start_server --app <app-output-path> --studies
<HSJD-images-path> --conf models <model-name>
```

<HSJD-images-path> corresponds to the path of the images that will be used in the model, and <model-name> refers to the name of the corresponding model to be used, in this case the `localization_vertebra` model.

After this step, the server is running locally and can be used in 3D Slicer. On the other hand, to stop the server, press Ctrl + C.

Every time the MONAI Label is wanted to be used in 3D Slicer, the commands in steps 4 and 7 must be executed in the Python local environment created.

5.2.3. Installing 3D Slicer and MONAI Label Module

The 3D Slicer software is the one used to interact with the MONAI Label module. The steps to setup the MOANI Label Slicer Module are below.

1. **Download and install 3D Slicer**, from their main website slicer.org. The versions of 3D Slicer supported by the MONAI Label module are from version 5.0 onwards, this included.
2. **Install MONAI Label Module in 3D Slicer**. With 3D Slicer opened, the following instructions must be followed.
 - a. View → Extension Manager → Active Learning → MONAI Label → Install
 After these steps, restart 3D Slicer.

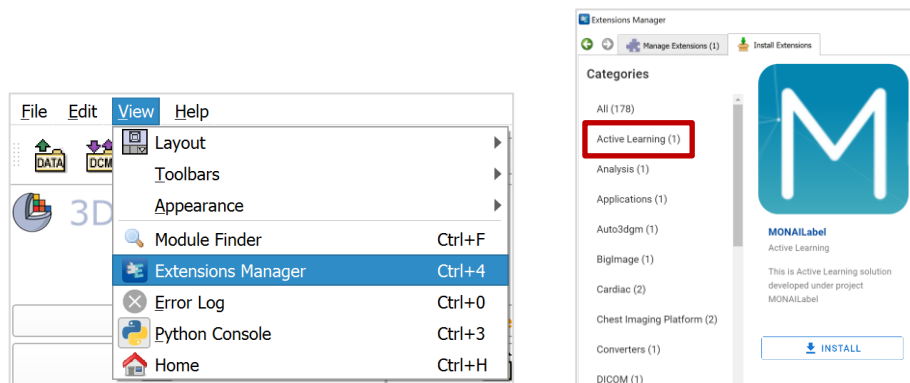


Figure 10: Instructions to install the MONAI Label Module in 3D Slicer.

To find the installed module, it can be searched using the *Module Finder* tool, which can be seen in Figure 10. Then, press *switch to module*, to be ready to be used.

5.3. CT image and Vertebrae Segmentation mask pre-processing

With the observations extracted from the VerSe dataset in the previous part, this section indicates the steps to follow to reproduce these observations in the HSJD dataset, therefore, the pre-processing steps to be carried out. In addition to the 3D Slicer, other software are used to perform

the pre-processing, which are the Materialise Mimics Medical, Materialise 3-matic Medical software and MRlcron program. The first two are used by surgical planning engineers in HSJD to perform the surgery planning. *Figure 11* shows a diagram indicating the programs to be used in each of the different steps, as well as a brief description of what is done in each of them, in the pre-processing, in the execution and in the results stages. In addition, the file format in which the images and/or masks are exported is indicated for each step.

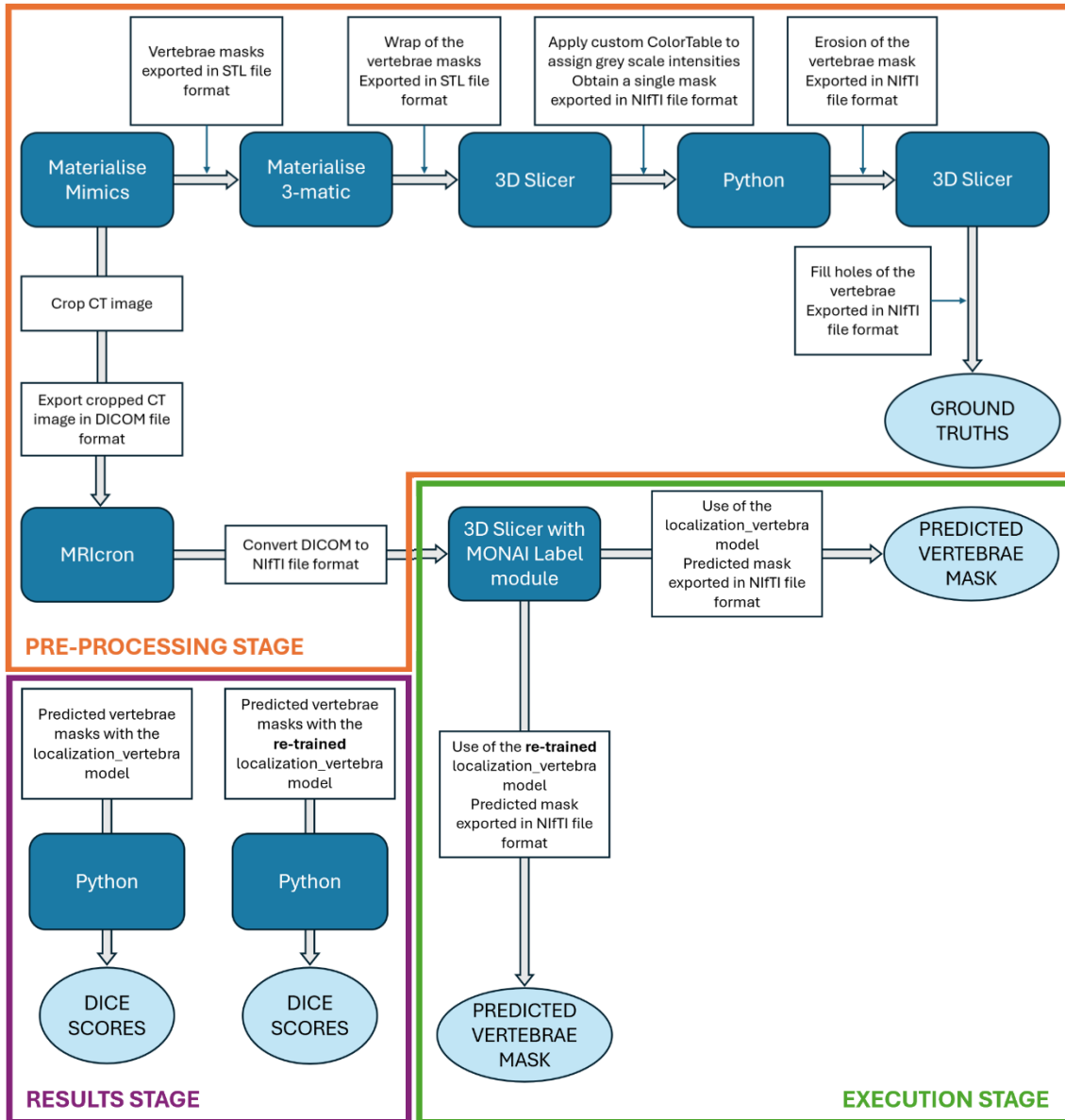


Figure 11: Diagram with the programs, file formats and a brief description for each of the steps performed in the pre-processing, execution and results stages.

5.3.1. Vertebrae Segmentation mask pre-processing

Firstly, the ribs are removed from the vertebrae segmentation masks, and secondly, each of the vertebrae are labelled with the same criteria used in the VerSe dataset. These two steps are performed with the Materialise Mimics Medical software. With these steps, the individual masks of

the vertebrae, without ribs, for the same case are obtained, which must be pre-processed with other programs and techniques, as indicated in the upper part of the diagram in *Figure 11*.

5.3.1.1. Removal of the ribs from the vertebrae segmentation masks and labelling

In relation to the first step, the ribs appear segmented in the masks as in the surgical planning it does not influence the design process of the customized surgical guides. *Figure 12* shows the phases (A, B, C, D) carried out to remove the ribs from the vertebra segmentation mask, for the concrete case of the T9 vertebra, in turquoise. Using the tool *Split Mask* in Mimics, different parts of the initial mask can be separated into different ones by indicating with different colours which are the regions to be treated separately (B). Therefore, the mask of no interest can be eliminated (C), and the starting mask can be replaced by the correct mask in which only the vertebra is segmented (D). This procedure is carried out for all the vertebrae where the ribs have also been segmented in the mask. Each vertebra is assigned a different colour to facilitate the visualization of each of them.

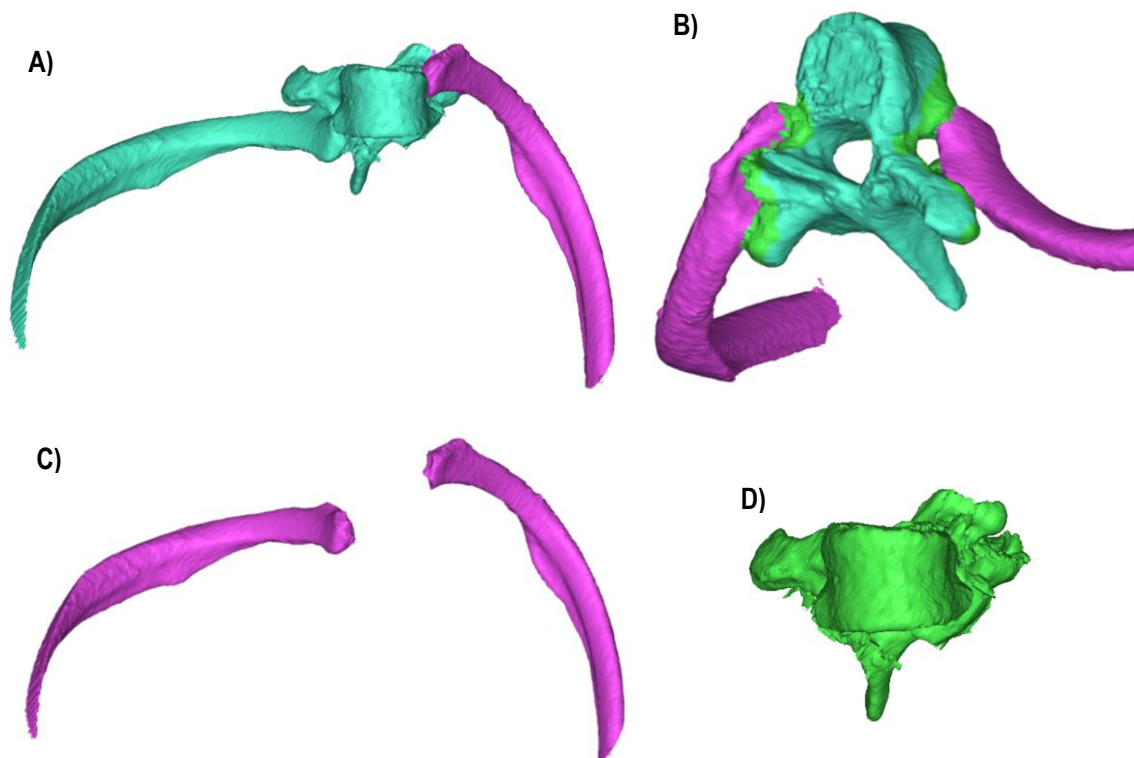


Figure 12: T9 vertebra mask split into the main vertebra mask, in green, and the rib mask, in purple.

However, there were cases in which the segmentation masks of the vertebrae did not have the transverse process correctly segmented. Being this a part of the vertebra that must be correctly segmented, the masks of these cases were redone. For this purpose, the patient's CT image is opened in the Mimics program, and by means of the tool *New Mask* incorporated in the program, a mask of the entire bone structure present in the image is obtained. This is achieved by defining a threshold with a minimum and a maximum Hounsfield unit value, which delimit the organs and structures in the body that are not of interest from those that are, that is the bone. However, in this process, because the cancellous bone has many holes and therefore air, the latter has a different HU than bone, the inner part of the vertebrae is not segmented correctly (see image A) in *Table 2*).

This will be discussed below. Nevertheless, from this new generated mask, the steps to be carried out to have separate masks for each vertebra are the same as the ones explained above.

Before exporting the vertebrae masks for each case, these are first labelled and then exported in STL format. STL files contain surface meshes, which consist of a set of polygonal faces, that taken together, form a surface covering of the object [55]. Therefore, those holes in the cancellous bone of the vertebrae, are not covered when exported into STL file format (see image A) in *Table 2*). Additional pre-processing techniques must be applied to these, explained in the following sections.

5.3.1.2. Removal of the holes in the vertebra STL files with 3-matic software

For this purpose, the Materialise 3-matic Medical software is used. Vertebra STLs, which is a file format supported by this software, are imported by dragging and dropping them into the main scene of the platform. Then, for each of the vertebrae, the *Wrap* tool is applied. This function allows to take into account the holes, that have not been considered in the surface mesh in the previous step, when exporting them with the 3-matic software in STL format. The latter are the STLs to be used in the following steps. Image B) in *Table 2* shows the improvement in the vertebra mask achieved with this process.

5.3.1.3. Assign grey scale intensities to the vertebrae masks

The next step consists of assigning the correct intensity value to each vertebra mask, and exporting all the vertebra masks grouped in one, as shown in *Figure 9*. The indicated procedures are carried out in 3D Slicer.

1. Import the STL files of a HSJD case by dragging and dropping them into the main scene. A tab will appear indicated how it is wanted to import the STLs. Choose *Segmentation*. These will appear in the **Data** module, separately. However, these must be grouped under a same segmentation, to be all exported in the same mask.

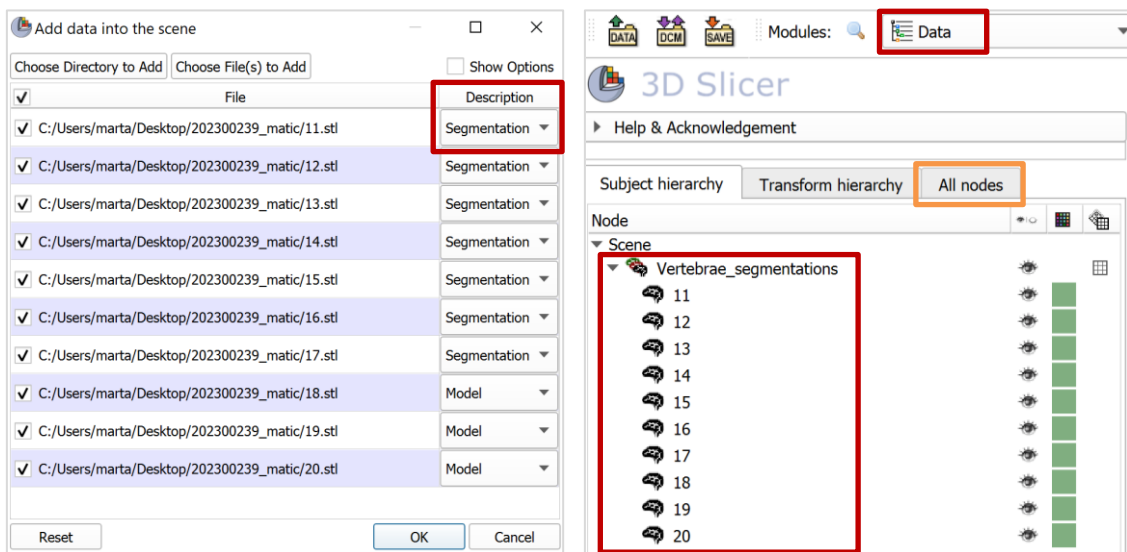


Figure 13: On the left, tab where it is indicated how to import the STLs. On the right, it is displayed the Data module with the different segmentations grouped under the same.

2. Import the corresponding CT image for a better visualization of the vertebra segmentations. Drag and drop the image into the scene and choose *Volume* instead of *Segmentation*, in the tab shown in the right image of *Figure 13*.
3. Create the ColorTable that will be used to assign the intensity labels to all vertebrae. Right click on “Vertebrae_segmentations” containing all the segmentations and choose *Export visible segments to binary labelmap*. The *labelmap* operation doesn’t allow overlap between two different structures, therefore, a voxel can only be assigned to one of the labels, ensuring that each vertebra will have its correct intensity value. With this action, a ColorTable is generated in *All nodes* section, indicated with an orange rectangle in *Figure 13*. Right click on the ColorTable and choose *Edit properties*. *Figure 14*, on the left, shows the displayed scene. However, the ColorTable does not contain the labels for the 24 vertebrae the spine has, therefore, it needs to be modified.
 - a. In *Number of Colors* indicate 25, where 0 is the background and the remaining 24 colours are the different vertebrae.
 - b. The *Label* of each *Color* is assigned from 1 to 24, as these are the labels of the vertebrae STLs (assigned with the Materialise Mimics software), which are the ones used in the VerSe dataset. Therefore, the labels of the ColorTable must be the same as the labels of the vertebrae STLs imported. Moreover, each of the labels can be assigned a different colour for a better visualization. In the right image of *Figure 14*, the correct ColorTable is shown.
 - c. Save the ColorTable.

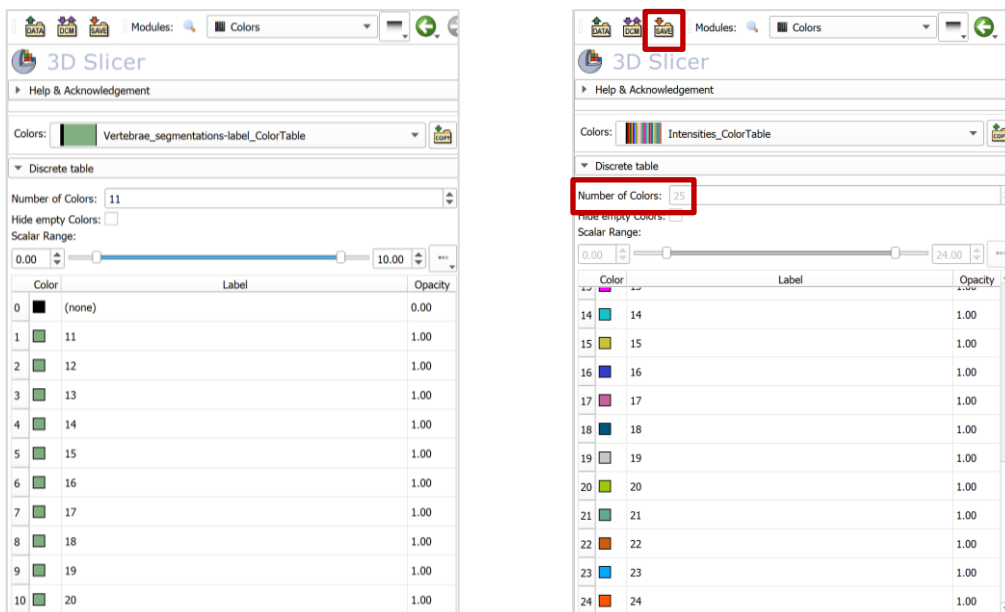


Figure 14: Default ColorTable, on the left. Modified ColorTable with 24 labels, one for each vertebra, on the right.

4. Apply the ColorTable created to ‘Vertebrae_segmentations’ and export it in NIfTI format. This is carried out in the **Segmentations** module (see *Figure 15*).
 - a. Choose the *Active segmentation*. In this case, ‘Vertebrae_segmentations’.

- b. In *Export to files* select as *Reference volume* the corresponding image previously imported, NIFTI as the *File format*, and the ColorTable created.
- c. Export to the specified *Destination folder*. The resulting mask is shown below.

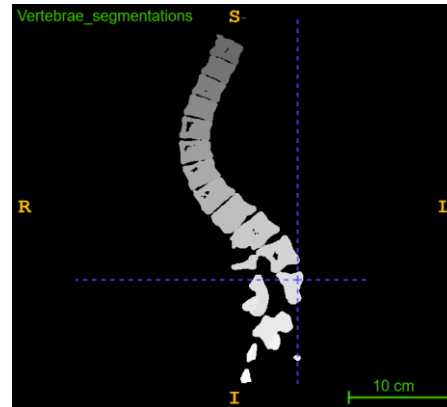
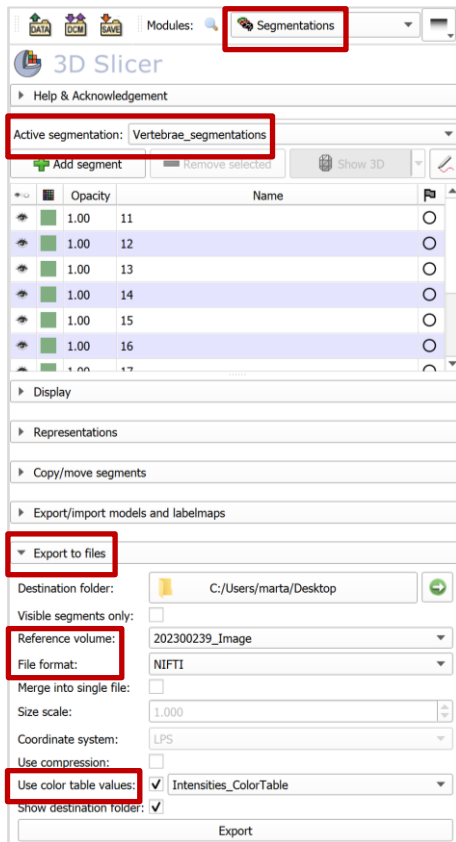


Figure 15: On the left, Indications for applying the ColorTable to the segmentations and exporting the latter in NiftI format. On the right, the resulting mask with the corresponding intensity for each vertebra.

5.3.1.4. Perform Erosion to the vertebrae masks

After analysing the vertebrae masks, it is observed that the boundary of the vertebrae in the final mask obtained in step 4 of section 5.3.1.3. are larger after applying the *Wrap* tool with the Materialise 3-matic software, compared to the initial STLs of the vertebrae without any pre-processing step. Since the morphology of the initial vertebrae is being modified, it needs to be solved. For this purpose, a Python code has been created, which can be seen in more detail in Annex A, which aims to shrink the vertebrae masks generated with Materialise 3-matic, by removing pixels from their boundary. This concept is known as erosion. Image C) in Table 2 shows the result after the erosion, for a concrete vertebra.

5.3.1.5. Removal of the holes still present in the vertebrae masks

Observed in image C) in Table 2, the mask is still presenting holes in the vertebrae, although the previous step with the Materialise 3-matic software has been performed (section 5.3.1.2). Another approach to remove the remaining holes is to use the *Smoothing* function available in 3D Slicer. This function is applied to imported STLs files in stage 5.3.1.3 grouped in

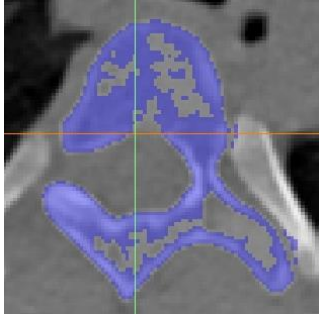
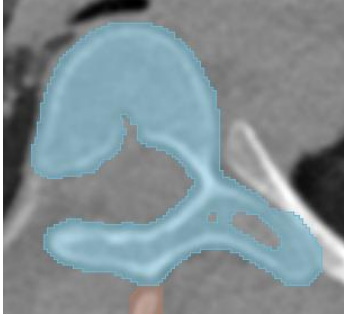
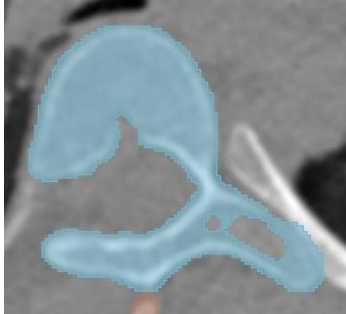
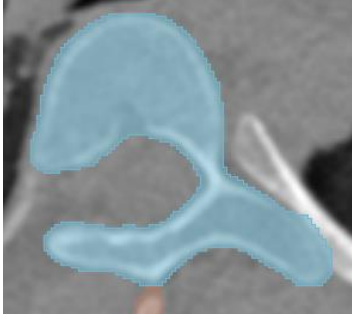
'Vertebra_segmentations'. Therefore, the steps described in section 5.3.1.3. and 5.3.1.4. must be performed into the new 'Vertebrae_segmentation' mask exported. To apply the smoothing function:

1. In Segment Editor, chose the *Segmentation* to be modified, and a vertebrae mask.
2. Select the *Smoothing* function, and the *Closing (fill holes) method*.
3. In the viewer part, 'paint' the holes present in the vertebra selected and click *Apply*.
4. Repeat these steps for each of the vertebrae present in the grouped segmentation.

Image D) in *Table 2* shows the final vertebrae mask, after applying all pre-processing steps, including fill holes smoothing function.

The following table summarises in images, of a vertebra viewed in the axial plane, the different pre-processing steps described in the previous sections to obtain the most accurate ground truths, saved in NIfTI format, as the ground truths are the masks used to evaluate the performance of the *localization_vertebra* model.

Table 2: Summary of the pre-processing steps with a corresponding illustrative image.

PRE-PROCESSING METHODS APPLIED TO THE VERTEBRAE STLS MASKS	
<p>A) Materialise Mimics - Export STLs</p> 	<p>B) Materialise 3-matic - Wrap</p> 
<p>C) Python - Erosion</p> 	<p>D) 3D Slicer - Smoothing</p> 

5.3.2. CT image pre-processing

HSJD CT images are cropped using the Materialise Mimics software. The crop is performed taking as reference for each different case the segmented vertebrae of the corresponding ground truth. Therefore, the image is cropped only along the Z-axis, so the upper vertebrae and those in the

lower part of the spine that are not segmented in the ground truth are not considered in the image. The cropping step is performed with the aim to obtain the most accurate result when evaluating the performance of the model, with the mask predicted by the model and the corresponding the ground truth, as the model will have segmented the same vertebrae than the ones in the ground truths.

After the image cropping, the image is exported in DICOM format from the Mimics software. However, although this file extension is accepted by 3D Slicer, it is not the format in which MONAI Label module in 3D Slicer must receive the images, they have to be in NIFTI format. The conversion from DICOM to NIFTI is performed with MRICron, a medical imaging visualization tool.

5.4. Execution

This section is divided into two main sections, one dealing with the execution of the `localization_vertebra` model of the Radiology application in the MONAI Label module, and the other with the re-training of this same model and its posterior execution. The predicted masks obtained after the execution of the models indicated in each of the sub-sections, are the masks to be used to compute the evaluation of the corresponding model.

5.4.1. `localization_vertebra` model

With the CT images of all HSJD cases in the same folder and the server running, the `localization_vertebra` model can be run in 3D Slicer MONAI Label module. The procedure is described below.

1. To connect to the server, click on the green circle icon.
2. Fetch one of the images of the folder indicated when starting the server, by clicking in the *Next Sample* button. The image will be displayed in the viewer scene.

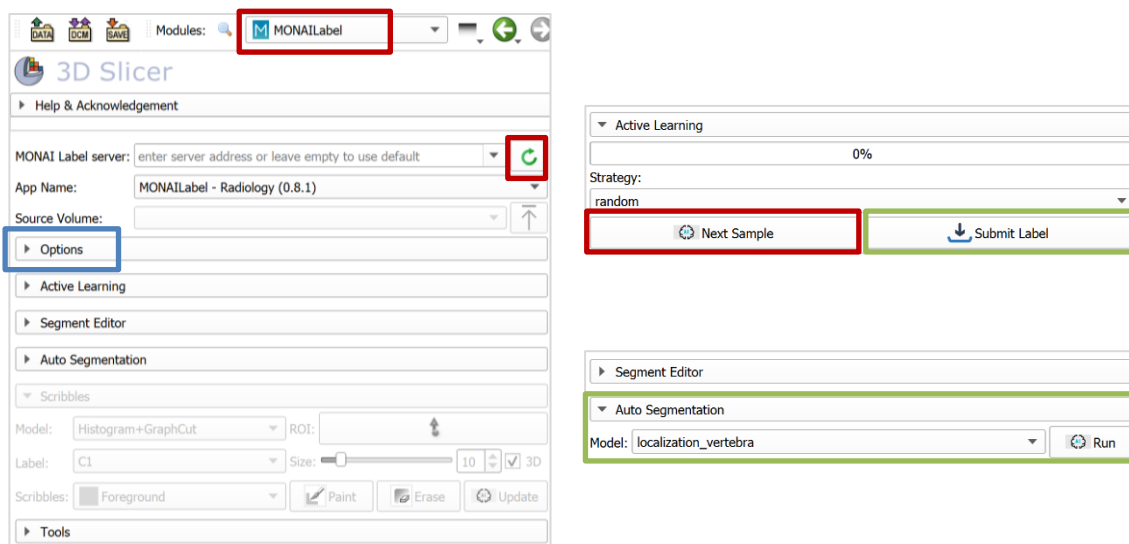


Figure 16: Initial steps to start with MONAI Label module in 3D Slicer, with the server running.

3. By using the *Auto Segmentation* functionality, the pre-trained model specified, `localization_vertebra`, can be used to segment the vertebrae from the CT image displayed. To save the vertebrae mask predicted by the model after the execution of the auto

segmentation, click on *Submit Label* button, indicated with the green rectangle in *Figure 16*. The mask is saved in a sub-folder generated inside the folder path specified when starting the server.

- Steps 2 and 3 above are repeated for all the images. However, only the predicted masks of the same subjects from the CT scans used to test the re-trained model will be used for the evaluation.

5.4.2. Re-train the *localization_vertebra* model

The pre-trained model *localization_vertebra* has been trained with data of patients with no pathological spine conditions (VerSe dataset). Therefore, the model does not correctly localise the vertebrae where the curvature of the scoliosis is more noticeable. To improve the performance of the pre-trained model for localising vertebrae of subjects with scoliosis, this is re-trained with the ground truths of the HSJD dataset. However, the re-trained model is saved as a new model inside the Radiology application folder. The percentages used for the training, validation and testing sets are 70, 20 and 30, respectively, so that in the training, excluding the validation, there is a minimum of 60% images with respect to the entire HSJD dataset. The 20% of the validations set is with respect to the train set and can be modified in the *Options* section (blue rectangle in *Figure 16*). These percentages in images correspond to 13, 2 and 5, respectively. Therefore, to prepare the data, train and test ground truths must be in separate folders.

The re-training procedure is carried out in the MONAI Label module in 3D Slicer.

- Open a new terminal and start another server, specifying the path in which there are the test CT scans. Connect to the server from 3D Slicer MONAI Label module.
- Name the folder containing the train ground truths as *final*, as this is the name by which the model recognises the masks to be used for re-training. Place this folder in the same path as the test CT scans, inside another folder named *labels*. Important to consider that the test CT scans and the corresponding ground truths must have the same name for the model to run, otherwise an error is raised.
- In MONAI Label module, inside *Active Learning*, click on *Train* to start the re-training of the model indicated. The model trains with the data placed in the *final* folder.

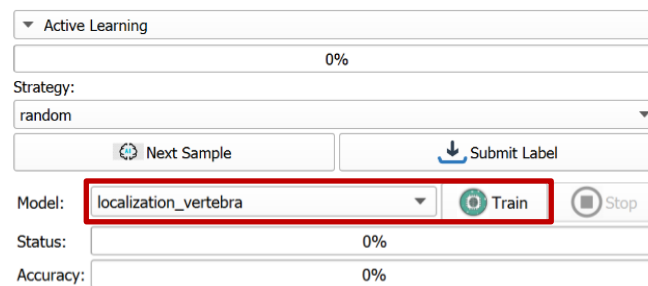


Figure 17: Instructions to re-train a model in MONAI Label module.

The *Status* bar moves according to the status of the training process of the model, and the *Accuracy* bar shows the accuracy of the model after the training.

5.4.2.1. Test the re-trained localization_vertebra model with the test CT scans

After training the localization_vertebra model, the model is used to predict the new vertebrae mask of the test CT scans. For each of them, the following steps are carried out.

1. Click *Next Sample*. The image displayed in the viewer corresponds to the test CT scans, as this is the folder specified on the running server.
2. Use the *Auto Segmentation* functionality. The bar below *Active Learning* heading scrolls according to the number of test CT scans the model has segmented out of the total number of test CT scans.
3. Save the new predicted mask by submitting the label using *Submit Label*. The mask is saved in a sub-folder generated inside the folder specified when starting the server.

5.5. Results

As firstly indicated in the diagram in *Figure 11*, the measure to evaluate the performance of the two models used, the localization_vertebra model and the re-trained localization_vertebra model, is the Dice score. This coefficient measures the similarity or overlap between two sets of data. The values provided by the Dice score range from 0 to 1, where a score of 0 indicates no overlap between the two sets, and a score of 1 indicates perfect overlap, therefore, the predicted segmentation is identical to the ground truth [34]. To compute the Dice scores the segmentation masks to be evaluated must be treated as sets of pixels and represented as binary masks, where every pixel is assigned either as part of the segment or as part of the background. These post-processing techniques have been performed using Python. The Python code implemented is detailed in Annex B. The two models previously indicated are evaluated in the different sub-sections below.

5.5.1. localization_vertebra model of the Radiology application

The Dice score is computed between the masks predicted by the localization_vertebra model without this being re-trained of the same subjects from the CT scans used in the test set, and the ground truths of the corresponding subjects obtained after performing all pre-processing steps, specified in section 5.3.

Table 3 shows the performance in terms of Dice scores for each pair of predicted and ground truth masks of the subjects from the test CT scans. The average Dice score has also been computed.

Table 3: Performance of the localization_vertebra model in terms of Dice Scores.

Performance in terms of Dice Scores localization_vertebra model	
Test subject 1	0.50
Test subject 2	0.83
Test subject 3	0.69
Test subject 4	0.60
Test subject 5	0.67
Average Dice Score	0.66

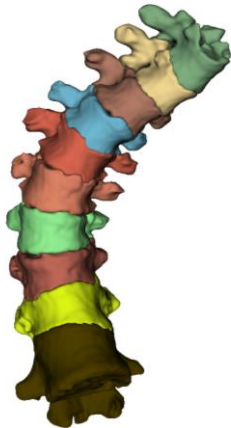

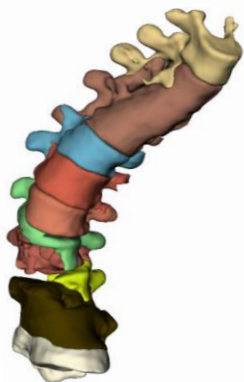
5.5.2. Re-trained localization_vertebra model

In this section a similar procedure is carried out. The Dice score is computed between the masks of the test CT scans predicted by the re-trained localization_vertebra model, and the ground truths of the corresponding subjects. *Table 4* shows the obtained Dice scores for each pair of predicted and ground truth masks.

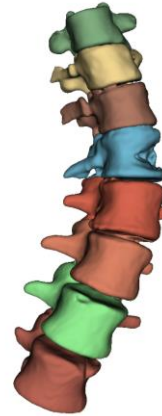
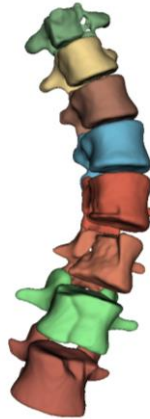
Table 4: Performance of the re-trained localization_vertebra model in terms of Dice Scores.

Performance in terms of Dice Scores Re-trained localization_vertebra model	
Test subject 1	0.81
Test subject 2	0.89
Test subject 3	0.86
Test subject 4	0.85
Tests subject 5	0.87
Average Dice Score	0.86

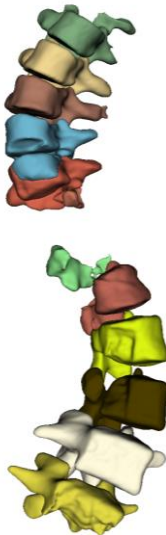
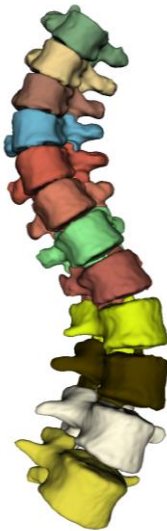
Moreover, *Table 5* shows the 3D volume rendering of the ground truth masks, the masks predicted by the localization_vertebra model, indicated under the header 'Predicted mask 1', and the masks predicted by the re-trained localization_vertebra model, under the header 'Predicted mask 2', of the test CT scans.

3D Volume Rendering of the ground truths and the masks predicted by the two vertebra_localization models		
Ground truth	Predicted mask 1	Predicted mask 2
Test subject 1		
		

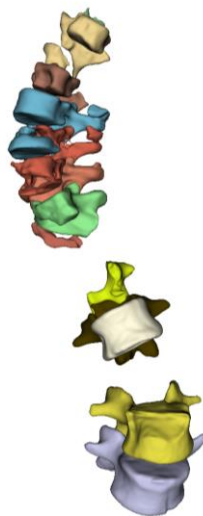
Test subject 2

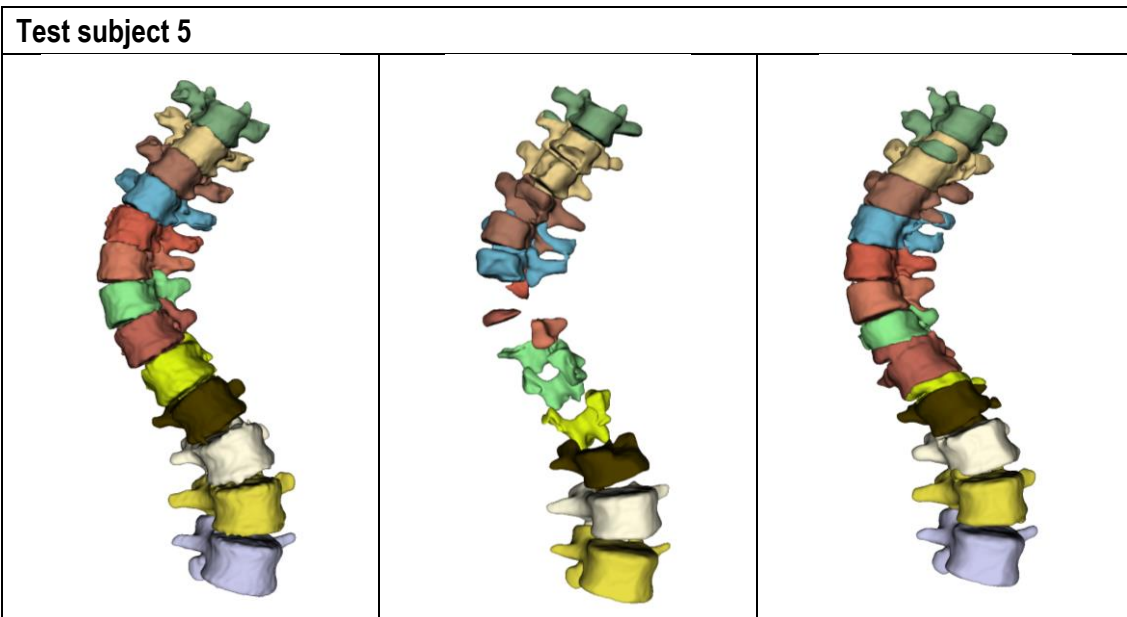


Test subject 3



Test subject 4





Qualitatively, it is observed that those predicted masks with the *localization_vertebra* model, under the header 'Predicted mask 1', in which a large majority of the vertebrae have not been segmented (pair 1 and pair 4), the corresponding Dice score is not higher than 0.65, the lowest value being 0.5 for pair 1. However, after re-training the model, the minimum Dice score value obtained is 0.81, corresponding to pair 1, whereas the higher one is 0.89. Quantitatively and qualitatively, an improvement in the performance of the model after being re-trained with the HSJD dataset is observed.

5.6. Discussion

The evaluation metric used to assess the similarity between the ground truths and the predicted masks obtained after the execution of the corresponding models has been the Dice coefficient. Regarding the vertebrae segmentation task, the Dice scores obtained in *Table 3* show that the *localization_vertebra* model has a very low accurate performance, as the average Dice Score obtained is of 0.66 out of 1. Therefore, vertebrae are either not segmented or partially segmented, as can be seen in *Table 5*.

In comparison, from the results shown in *Table 4*, obtained with the re-trained model, the average dice score is higher than 0.66, being 0.86. This increase in the average Dice score translates into a better performance of the model in vertebrae segmentation. However, due to the small number of samples in the test set used, the resulting average Dice score is not reliable enough to extract results that can be generalised to other datasets. On the other hand, the Dice scores indicate that re-training the starting model, the *localization_vertebra* model, with the data of interest, allows to get results closer to 1 than without re-training, providing a better segmentation of the vertebrae than with the starting model.

The low value of the average Dice score of the pre-trained model may have several explanations:

- The starting model has been pre-trained with the VerSe dataset. Therefore:
 - It has been pre-trained with CT scans of subjects with a mean age of ~ 59 (± 17) years, not paediatric.
 - Those subjects have a normal anatomy of the spine, do not have spinal conditions.
- A lack of data to train the model, as the surgical planning of Adolescent Idiopathic Scoliosis surgery were not performed until 2023.
- A lack of data to test the model. Therefore, a coarse segmentation has a significant impact on the computation of the average Dice score.
- Another possible consideration lies in the numerous pre-processing steps that must be carried out to obtain the most accurate ground truths possible, mostly manually, although Python code has been used in some techniques to reduce possible variability in the execution.
 - Subjectivity and inter-observer variability. To extract the initial mask from the CT scan, from which all pre-processing methods are performed, a threshold with a minimum and a maximum Hounsfield unit value, defined by the planning engineer, is used.

6. Execution Schedule

The different systems used for the organization and planning of the tasks to be performed throughout the project are detailed in the following sub-sections.

6.1. Work Breakdown Structure (WBS)

WBS is a managing tool that breaks the project in different tasks, for a better organisation and planification of the execution of these. The corresponding WBS of this project has been carried out and is shown below.

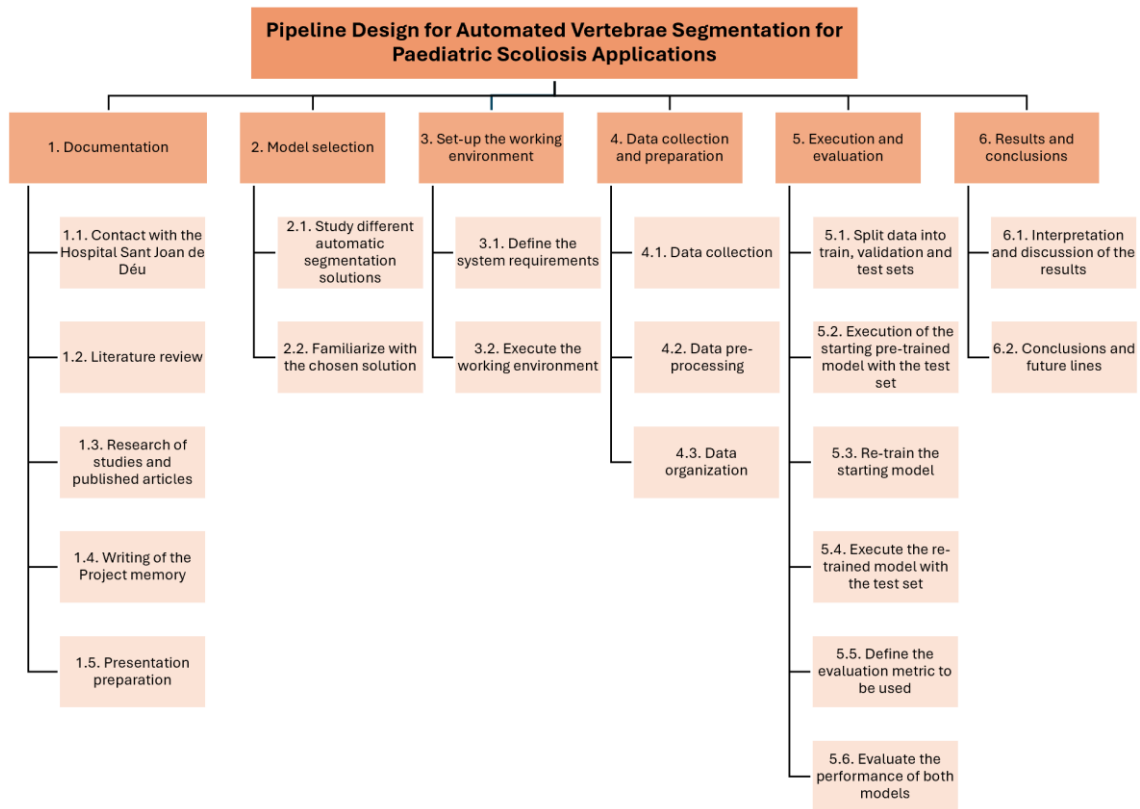


Figure 18: Work breakdown structure of the Project.

6.2. GANTT Chart

A Gantt chart is another strategy for the management of the project, and therefore, of the activities and tasks that are performed during the development of the same. It allows to visually recognise which are the activities being executed or have been completed, as they are represented against time. To be able to represent the Gantt chart, first it has been necessary to assign to each of the tasks defined in the WBS its corresponding duration, in weeks. In those tasks where the duration has been less than one week, it has been indicated in days. However, notice that the Gantt works on a weekly basis. This information is displayed in the following Table 5.

Table 5: Tasks duration definition.

Tasks	Duration (weeks)
1.1. Contact with the Hospital Sant Joan de Déu	1
1.2. Literature Review	3
1.3. Research of studies and published articles about automatic segmentation	5
1.4. Writing of the project memory	11
1.5. Presentation preparation	1
2.1. Study different automatic segmentation solutions	2
2.2. Familiarize with the chosen solution	14
3.1. Define the system requirements	5
3.2. Execute the working environment	1
4.1. Data collection	2
4.2. Data pre-processing	4
4.3. Data organization	2
5.1. Split data into train, validation and test sets	2 days
5.2. Execution of the starting pre-trained model with the test set	4
5.3. Re-train the starting model	4
5.4. Execute the re-trained model with the test set	1
5.5. Define the evaluation metric be used	1 day
5.6. Evaluate the performance of both models	3
6.1. Interpretation and discussion of the results	1
6.2. Conclusions and future lines	1

Below is shown the Gantt Diagram performed using the table above.

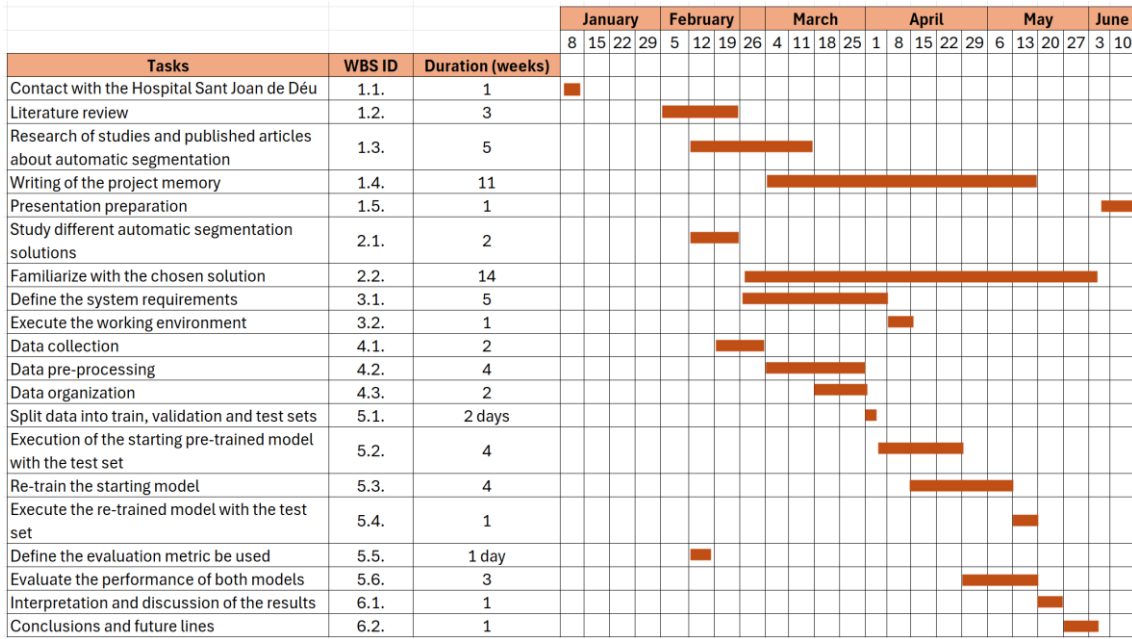


Figure 19: Gantt chart of the Project.

7. Technical Feasibility

This section focuses on the analysis of both internal and external factors that can have an important impact on the feasibility of the project. For this purpose, the strengths of the project and the weaknesses to be addressed for improvement, as well as, from the external point of view, the threats surrounding the project and the opportunities that are presented to the same, are identified and exposed in the matrix below, known as SWOT matrix. This strategic planning technique allows to visually recognise each of the previous parts indicated.

Table 6: SWOT matrix analysis of the Project.

STRENGTHS	WEAKNESSES
<ul style="list-style-type: none"> - Innovative and unique. - Reproducible. - Use of paediatric data. - Accelerate the segmentation task. 	<ul style="list-style-type: none"> - Small size dataset. - Shortage of training samples. - Manual tasks in the pre-processing stage. - Lack of previous experience and knowledge of the topic by the team. - Lack of generalizability of the findings.
OPPORTUNITIES	THREATS
<ul style="list-style-type: none"> - Increasing research and demand for the use of automated methods in medical imaging. - Demand within different medical specialities. - Few availability of models trained on paediatric data. 	<ul style="list-style-type: none"> - Expensive software licenses.

Starting with the internal factors of the project, this presents both strengths and weaknesses. The pipeline design elaborated for the automatic vertebrae and spine segmentation of AIS cases, is unique and innovative, since these techniques are not currently used in the Hospital Sant Joan de Déu, but it is a need to be fulfilled to optimize the segmentation task performed during the surgical planning of this cases. On the other hand, the shortage of training data available for the training of the model does not allow to generalize the findings, which represents a weakness of the project. Also, the pre-processing tasks that are carried out manually, since these influence directly on the performance of both, the starting and the re-trained models. Moreover, the lack of previous experience and knowledge of the topic by the team, since this knowledge has been acquired during the development of the project.

Analysing the external factors detailed in the SWOT matrix, which correspond to the lower part of the matrix, numerous opportunities are presented to this project. First due to the high demand for automatic segmentation models to be applied in the healthcare sector. In addition, as there are few available models trained with paediatric data, this enhances the project towards numerous possibilities for this designed pipeline to be used in other paediatric hospitals. On the other hand, the high licensing costs of some of the software to be used pose a threat for the project.

8. Economic Feasibility

The aim of this section is to analyse the costs associated with the development of this project. These are presented in *Table 7* and are organized according to whether they correspond to hardware, software or to human resources.

Table 7: Hardware, software and human resources costs of the Project.

	COSTS
SOFTWARE	
3D Slicer and MONAI Label module	-
Materialise Mimics Medical	37.000 €
Materialise 3-matic Medical	
MRlcron	-
HARDWARE	
NVIDIA Quadro RTX 5000 16 GB	1.374 €
HP 1920 x 1080 Monitor	240 €
HP Workstation Z4 intel XEON	2.961 €
HP Spectre laptop	2.499 €
HUMAN RESOURCES	
Biomedical engineering student	6.375 €
TOTAL:	50.449 €

Referring to the software used for the development of this project, all of them are free with open access, except for those of the Materialise company, which require a license. However, both Materialise Mimics Medical and Materialise 3-matic Medical are available under the same license, and the corresponding price indicated in *Table 7* is an estimate given by the surgical planning engineer, as these are the software used for surgical planification.

As for the hardware used, the NVIDIA graphics card [56], the HP monitor [57] and HP workstation [58] are the components used in the installations of the 3D4H Unit and PeCIC departments. On the other hand, the HP Spectre laptop [59] is the personal laptop of the author of this project.

Regarding human resources, only the author's working hours have been taken into account to evaluate the costs, as the work of the supervisors during the development of the project has not implied an additional increase in their salary. The corresponding human resources cost has been calculated considering an average of 25 hours of work per week by the author, and a total of 17 weeks, which is the period establish for the development of the Final Degree Project. Therefore, considering a salary of 15 €/hour for a biomedical engineering student, the total cost of human resources amounts to: 6.375 €.

The total cost of the project is 50.449 €.

9. Regulation and Legal aspects

When working with medical data of patients, these could be the main medical images but also the information extracted from them, there are several regulations and laws that must be followed to ensure the correct data management and protection to guarantee the privacy of the patient personal information.

Regarding the use of health data in research projects, the following law, published in the *Boletín Oficial del Estado (BOE)*, must be complied: *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y Garantía de los derechos digitales* [60]. The purpose of this law is to increase the rights of individuals and the obligations of companies in the processing of personal data. Moreover, this regulation establishes data protection requirements and responsibilities in companies on how to proceed with personal information, as well as the rights of users and consumers [61].

Another regulation to be considered, which has also been published in the *BOE*, is: *Orden SSI/81/2017, de 19 de enero*. This law establishes the basic guidelines to ensure and protect the right to patient privacy for students and residents in the health sciences [62].

Moreover, concerning the use of AI-based models in this project, legal aspects regarding its application in healthcare must also be taken into account. *Real Decreto 817/2023, de 8 de noviembre*, also published in the *BOE*, establishes harmonized rules in the field of artificial intelligence with the aim of ensuring respect for the fundamental rights of citizens, and also generating confidence in the use and development of AI in the society and the economy [63].

10. Conclusions and Future lines

The realization of this project has allowed the development of a detailed pipeline for the implementation of an automatic segmentation model of the vertebrae and spine from CT images, with the aim of optimizing the time spent by the planning engineer on the segmentation task during the surgical planning of adolescent idiopathic scoliosis cases.

The literature in this field shows how the use of artificial intelligence and deep learning in imaging studies has allowed significant advances in the management of spinal diseases, becoming DL-approaches a widely used clinical application for treatment, diagnosis, clinical decision-making and treatment planning, as these algorithms have demonstrated important capabilities in extracting relevant data from medical images. In reference to the surgical treatment planning of idiopathic adolescent scoliosis cases, numerous articles state that the use of 3D printed surgical guides substantially reduces the misplacement of pedicle screws, and the complications associated to it. However, the manual segmentation procedure that is carried out during the planification is considered a repetitive task, subjective, and predisposed to inter-observer variability. The developed pipeline aims to tackle this issue, by indicating and defining the steps to be carried out to implement the proposed model for automatic vertebrae and spine segmentation, in order to optimize this task and make it reproducible, reducing the variability of results between users.

This project has succeeded in finding an open-source automatic segmentation model that could be used for the main objective of the present project. Moreover, it has succeeded in describing in detail all the processes and steps to be carried out for the correct use of the proposed model. Firstly, in defining the system requirements to be fulfilled to be able to use the model, otherwise, errors would raise when trying to use it. Moreover, in studying which was the working setup environment needed to correctly execute the model and be able to work with. In addition, this project has also achieved defining the pre-processing methods needed to be performed to obtain the vertebrae and spine segmentations masks with the same characteristics as those used to pre-train the model in the Radiology application of MONAI Label. After the implementation of all the previous steps, the project has succeeded in executing the model without any error while running, and therefore, obtaining the segmentation masks predicted by this model, to use them for their later evaluations. Furthermore, after an extensive research and numerous trials, the initial model has been successfully re-trained with data from paediatric cases with scoliosis from the Hospital Sant Joan de Déu, in order to give to the model algorithm data to learn from, expecting it to predict better the segmentation masks. The evaluation of the performance of the two models, initial and re-trained, has been computed using the Dice score metric, that evaluates the overlap between two given segmentation masks. Note that the resulting Dice scores are not of the individual vertebrae but correspond to each of the vertebrae together to form the spine, and score on this last. The results show that after re-training the initial model, a higher Dice score is obtained, therefore, the prediction of the segmented masks is better.

As a project started from scratch, this is the beginning of an innovative path to implement the automatic vertebrae and spine segmentation models in the surgical planning procedure of AIS cases at the Hospital Sant Joan de Déu. In view of this, there are future lines for this project, which are indicated below.

- Re-train the model with a larger dataset of CT scans of AIS cases to obtain more accurate segmentation masks, as it has been concluded that the Dice score obtained for the test set with the re-trained model is higher than with the starting model, therefore, resulting on a better prediction of the segmentation mask.
- Reduce the number of pre-processing steps that are carried out manually, by using other software or applications that allow performing these indicated processes with semi-automated tools incorporated on them.
- In relation to the above, simplify the pipeline by finding other possible programs or applications that allow performing different pre-processing tasks in the same step, taking into account the indicated characteristics the segmentations resulting from these tasks must fulfil for the correct use of the model.

11. References

- [1] "In brief: How does the spine work? - InformedHealth.org - NCBI Bookshelf." [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK279468/>
- [2] M. Harrow-Mortelliti, V. Reddy, and G. Jimsheleishvili, "Physiology, Spinal Cord - StatPearls - NCBI Bookshelf." [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK544267/>
- [3] "Columna vertebral: enfermedades usuales | Centro Médico ABC." [Online]. Available: <https://centromedicoabc.com/revista-digital/tipos-de-enfermedades-comunes-en-la-columna-vertebral/>
- [4] G. Bin Lee, D. T. Priefer, and R. Priefer, "Scoliosis: Causes and Treatments," *Adolescents 2022, Vol. 2, Pages 220-234*, vol. 2, no. 2, pp. 220–234, Apr. 2022, doi: 10.3390/ADOLESCENTS2020018.
- [5] B. L. Mesiti, "Scoliosis: An Overview," *Radiol Technol*, vol. 93, no. 1, pp. 55–72, 2021.
- [6] R. M. Egea-Gámez, M. Galán-Olleros, A. González-Menocal, and R. González-Díaz, "[Translated article] How do I plan adolescent idiopathic scoliosis surgery? Systematization of a preoperative planning method," *Rev Esp Cir Ortop Traumatol*, vol. 68, no. 1, pp. T73–T85, Jan. 2024, doi: 10.1016/J.RECOT.2023.11.002.
- [7] L. Pinto-Coelho, "How Artificial Intelligence Is Shaping Medical Imaging Technology: A Survey of Innovations and Applications," *Bioengineering*, vol. 10, no. 12, Dec. 2023, doi: 10.3390/BIOENGINEERING10121435.
- [8] "Drug Research Ethics Committee | Research | Recerca Sant Joan de Déu." [Online]. Available: <https://www.sjdrecerca.org/en/research/ceim/>
- [9] "Escoliosis - Barcelona Spine Institute - BSI." [Online]. Available: <https://www.barcelonaspineinstitute.com/es/patologias-columna/escoliosis/>
- [10] C. DeSai, V. Reddy, and A. Agarwal, "Anatomy, Back, Vertebral Column - StatPearls - NCBI Bookshelf." [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK525969/>
- [11] J. S. Blanco, S. Mackie Doyle, J. H. Heyer, and R. F. Widmann, "Scoliosis in Children and Teenagers: An Overview | HSS." [Online]. Available: https://www.hss.edu/conditions_scoliosis-pediatric-patient.asp
- [12] "Idiopathic Scoliosis | Boston Children's Hospital." [Online]. Available: <https://www.childrenshospital.org/conditions/idiopathic-scoliosis>
- [13] "Scoliosis Symptoms, Test & Treatment | NIAMS." [Online]. Available: <https://www.niams.nih.gov/health-topics/scoliosis>

- [14] M. H. Horng, C. P. Kuok, M. J. Fu, C. J. Lin, and Y. N. Sun, "Cobb Angle Measurement of Spine from X-Ray Images Using Convolutional Neural Network," *Comput Math Methods Med*, vol. 2019, 2019, doi: 10.1155/2019/6357171.
- [15] B. J. (Barend) van Royen, "Understanding the Lenke Classification for Adolescent Idiopathic Scoliosis (AIS)," *Curr Probl Diagn Radiol*, vol. 52, no. 4, pp. 233–236, Jul. 2023, doi: 10.1067/J.CPRADIOL.2023.01.003.
- [16] R. P. Menger and A. H. Sin, "Adolescent Idiopathic Scoliosis - StatPearls - NCBI Bookshelf." [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK499908/>
- [17] E. Cáceres, A. Molina, and A. Llado, "Tratamiento quirúrgico de la escoliosis idiopática del adolescente," *Rehabilitacion (Madr)*, vol. 43, no. 6, pp. 293–298, Nov. 2009, doi: 10.1016/S0048-7120(09)72834-8.
- [18] C. Faldini *et al.*, "Accuracy of Patient-Specific 3D-Printed Guides for Pedicle Screw Insertion in Spine Revision Surgery: Results of a Retrospective Study," *Surg Technol Int*, vol. 41, Nov. 2022, doi: 10.52198/22.STI.41.NS1642.
- [19] R. Cecchinato, P. Berjano, A. Zerbi, M. Damilano, A. Redaelli, and C. Lamartina, "Pedicle screw insertion with patient-specific 3D-printed guides based on low-dose CT scan is more accurate than free-hand technique in spine deformity patients: a prospective, randomized clinical trial," *European Spine Journal*, vol. 28, no. 7, pp. 1712–1723, Jul. 2019, doi: 10.1007/S00586-019-05978-3/TABLES/9.
- [20] T. D. DenOtter and J. Schubert, "Hounsfield Unit - StatPearls - NCBI Bookshelf." [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK547721/>
- [21] S. Jeong, A. Yang, M. M. Dhodapkar, S. S. Jabbouri, A. Jonnalagadda, and D. Tuason, "3D printed pedicle screw guides reduce the rate of intraoperative screw revision in adolescent idiopathic scoliosis surgery," *The Spine Journal*, vol. 23, no. 12, pp. 1894–1899, Dec. 2023, doi: 10.1016/J.SPINEE.2023.08.001.
- [22] L. Zou, "Meta-learning for healthcare," *Meta-Learning*, pp. 299–330, Jan. 2023, doi: 10.1016/B978-0-323-89931-4.00007-9.
- [23] M. Koenigkam Santos, J. Raniery Ferreira Júnior, D. Tadao Wada, A. Priscilla Magalhães Tenório, M. Henrique Nogueira Barbosa, and P. Mazzoncini De Azevedo Marques, "Artificial intelligence, machine learning, computer-aided diagnosis, and radiomics: advances in imaging towards to precision medicine," *Radiol Bras*, vol. 52, no. 6, p. 387, Nov. 2019, doi: 10.1590/0100-3984.2019.0049.
- [24] "Medical Image Segmentation Definition | Encord." [Online]. Available: <https://encord.com/glossary/medical-image-segmentation-definition/>
- [25] "About DICOM: Overview." [Online]. Available: <https://www.dicomstandard.org/about>

- [26] “STL files explained | Learn about the STL file format | Adobe.” [Online]. Available: <https://www.adobe.com/creativecloud/file-types/image/vector/stl-file.html>
- [27] “Terminal de sistemas Windows.” [Online]. Available: <https://aprendeconalf.es/docencia/python/herramientas/terminal-windows/>
- [28] “Using Python Environments in Visual Studio Code.” [Online]. Available: <https://code.visualstudio.com/docs/python/environments>
- [29] “¿Qué es una CPU? Explicación de la unidad central de procesamiento: AWS.” [Online]. Available: <https://aws.amazon.com/es/what-is/cpu/>
- [30] “¿GPU vs. CPU? ¿Qué es la computación por GPU? | NVIDIA.” [Online]. Available: <https://www.nvidia.com/es-la/drivers/what-is-gpu-computing/>
- [31] M. Yagi, K. Yamanouchi, N. Fujita, H. Funao, and S. Ebata, “Revolutionizing Spinal Care: Current Applications and Future Directions of Artificial Intelligence and Machine Learning,” *J Clin Med*, vol. 12, no. 13, p. 4188, Jul. 2023, doi: 10.3390/JCM12134188.
- [32] M. U. Saeed, N. Dikaios, A. Dastgir, G. Ali, M. Hamid, and F. Hajje, “An Automated Deep Learning Approach for Spine Segmentation and Vertebrae Recognition Using Computed Tomography Images,” *Diagnostics*, vol. 13, no. 16, Aug. 2023, doi: 10.3390/DIAGNOSTICS13162658.
- [33] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.
- [34] “Dice score - OECD.AI.” [Online]. Available: <https://oecd.ai/en/catalogue/metrics/dice-score>
- [35] C. Payer, D. Darko, H. Bischof, and M. Urschler, “Coarse to Fine Vertebrae Localization and Segmentation with SpatialConfiguration-Net and U-Net”, doi: 10.5220/0008975201240133.
- [36] C. Payer, D. Štern, H. Bischof, and M. Urschler, “Integrating spatial configuration into heatmap regression based CNNs for landmark localization,” *Med Image Anal*, vol. 54, pp. 207–219, May 2019, doi: 10.1016/J.MEDIA.2019.03.007.
- [37] N. Lessmann, B. van Ginneken, P. A. de Jong, and I. Išgum, “Iterative fully convolutional neural networks for automatic vertebra segmentation and identification,” *Med Image Anal*, vol. 53, pp. 142–155, Apr. 2019, doi: 10.1016/J.MEDIA.2019.02.005.
- [38] S. F. Qadri *et al.*, “Automatic Deep Feature Learning via Patch-Based Deep Belief Network for Vertebrae Segmentation in CT Images,” *Applied Sciences 2019, Vol. 9, Page 69*, vol. 9, no. 1, p. 69, Dec. 2018, doi: 10.3390/APP9010069.

- [39] H. Liebl *et al.*, “A computed tomography vertebral segmentation dataset with anatomical variations and multi-vendor scanner data,” *Scientific Data* 2021 8:1, vol. 8, no. 1, pp. 1–7, Oct. 2021, doi: 10.1038/s41597-021-01060-0.
- [40] A. Ahsan, “Medical Image Segmentation: Advancements and Applications,” *J Med Diagn Methods*, vol. 12, no. 3, pp. 1–1, Jun. 2023, doi: 10.35248/2168-9784.23.12.416.
- [41] “AI in Medical Imaging Market Size Envisioned at USD 14,423.15 Million by 2032.” [Online]. Available: <https://www.towardshealthcare.com/insights/ai-in-medical-imaging-moving-from-hype-to-reality>
- [42] “Bonescreen: AI for Precise Spine Health Assessment.” [Online]. Available: <https://www.bonescreen.de/>
- [43] A. Apazidis, P. A. Ricart, C. M. Diefenbach, and J. M. Spivak, “The prevalence of transitional vertebrae in the lumbar spine,” *The Spine Journal*, vol. 11, no. 9, pp. 858–862, Sep. 2011, doi: 10.1016/J.SPINEE.2011.08.005.
- [44] Mordor Intelligence, “Diagnostic Imaging Market - Size, Share & Trends Analysis.” [Online]. Available: <https://www.mordorintelligence.com/industry-reports/medical-imaging-market>
- [45] C. Constant *et al.*, “The use of deep learning in medical imaging to improve spine care: A scoping review of current literature and clinical applications,” *North American Spine Society Journal*, vol. 15, p. 100236, Sep. 2023, doi: 10.1016/J.XNSJ.2023.100236.
- [46] itk-SNAP, “TutorialSectionAutoSegmentation.” [Online]. Available: <http://www.itksnap.org/pmwiki/pmwiki.php?n=Documentation.TutorialSectionAutoSegmentation>
- [47] Center for Integrative Biomedical Computing, “Seg3D.” [Online]. Available: <https://www.sci.utah.edu/cibc-software/seg3d.html>
- [48] “MONAI Label - Product Page.” [Online]. Available: <https://monai.io/label.html>
- [49] “MONAI - Home.” [Online]. Available: <https://monai.io/>
- [50] “3D Slicer image computing platform | 3D Slicer.” [Online]. Available: <https://www.slicer.org/>
- [51] “MONAILabel/sample-apps/radiology at main · Project-MONAI/MONAILabel · GitHub.” [Online]. Available: <https://github.com/Project-MONAI/MONAILabel/tree/main/sample-apps/radiology>
- [52] A. Sekuboyina *et al.*, “VerSe: A Vertebrae labelling and segmentation benchmark for multi-detector CT images,” *Med Image Anal*, vol. 73, p. 102166, Oct. 2021, doi: 10.1016/J.MEDIA.2021.102166.
- [53] “Installation — MONAI Label 0.8.2 Documentation.” [Online]. Available: <https://docs.monai.io/projects/label/en/latest/installation.html>

- [54] “System Requirements - NVIDIA Docs.” [Online]. Available: <https://docs.nvidia.com/clara/monai/requirements.html>
- [55] D. C. Simkins and J. B. Alford, “The Role of Computational Tools in Biomechanics,” *Biomechanics of the Female Pelvic Floor*, pp. 351–366, Mar. 2016, doi: 10.1016/B978-0-12-803228-2.00017-9.
- [56] “PNY Nvidia Quadro RTX 5000 16GB GDDR6 | PcComponentes.com.” [Online]. Available: <https://www.pccomponentes.com/pny-nvidia-quadro-rtx-5000-16gb-gddr6>
- [57] “HP P24v G4 23.8" IPS FullHD | PcComponentes.com.” [Online]. Available: <https://www.pccomponentes.com/hp-p24v-g4-238-ips-fullhd>
- [58] “HP Z4 G5 Intel Xeon W3-2423/32GB/1TB SSD/RTX A2000 | PcComponentes.com.” [Online]. Available: <https://www.pccomponentes.com/hp-z4-g5-intel-xeon-w3-2423-32gb-1tb-ssd-rtx-a2000>
- [59] “HP Portátiles para Casa y Empresas | HP Store Spectre - Tienda Oficial.” [Online]. Available: https://www.hp.com/es-es/shop/list.aspx?sel=NTB&ctrl=f&fc_seg_home=1&fc_sb_spectre=1
- [60] “BOE-A-2018-16673 Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.” [Online]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>
- [61] “Ley Orgánica de Protección de Datos - LOPDGDD 3/2018 | Grupo Atico34.” [Online]. Available: <https://protecciondatos-lopd.com/empresas/nueva-ley-proteccion-datos-2018/>
- [62] “BOE-A-2017-1200 Orden SSI/81/2017, de 19 de enero, por la que se publica el Acuerdo de la Comisión de Recursos Humanos del Sistema Nacional de Salud, por el que se aprueba el protocolo mediante el que se determinan pautas básicas destinadas a asegurar y proteger el derecho a la intimidad del paciente por los alumnos y residentes en Ciencias de la Salud.” [Online]. Available: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2017-1200
- [63] “BOE-A-2023-22767 Real Decreto 817/2023, de 8 de noviembre, que establece un entorno controlado de pruebas para el ensayo del cumplimiento de la propuesta de Reglamento del Parlamento Europeo y del Consejo por el que se establecen normas armonizadas en materia de inteligencia artificial.” [Online]. Available: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2023-22767

Annexes

Annex A. Erosion pre-processing technique Python code

Python code used to apply the erosion (indicated in section 5.3.1.4) morphological operator function to the vertebra masks, after the *Wrap* operation has been applied with the Materialise 3-matic Medical software.

```
In [1]: 1 import os
2 import numpy as np
3 import nibabel as nib
4 import skimage
5
6 from nibabel import load, save, Nifti1Image
7 from skimage.morphology import erosion

In [2]: 1 path_initial_mask = r"\\Users\\marta\\Desktop\\masks_to_erode"
2 path_eroded_mask = r"\\Users\\marta\\Desktop\\masks_eroded"
3
4 # walk() module is used to display the file names in the specified directory tree
5 # For each directory tree, walk() generates 3-tuple that contain:
6 # The path of the directory, a list of sub-directories inside the current directory, and the filenames
7
8 for dirpath, dirnames, filenames in os.walk(path_initial_mask, topdown = False):
9     for name in filenames:
10
11         # To concatenate in the dirpath the path segment 'name' that corresponds to the filename for each image
12         mask_path = os.path.join(dirpath, name)
13
14         # Load an image of the specified path
15         mask_nifti = nib.load(mask_path)
16
17         # Call the get_fdata() method to get the array data of the loaded image
18         mask_image = mask_nifti.get_fdata()
19
20         # Execute the erosion function to the array data of the image
21         new_image = erosion(mask_image)
22
23         # Adapt the loaded image (mask_nifti) to the new_image data shape and affine
24         final_image = nib.Nifti1Image(new_image.astype(int), mask_nifti.affine, mask_nifti.header)
25
26         # Save the eroded image in the specified path with the extension '.gz' (NIfTI)
27         nib.save(final_image, os.path.join(path_eroded_mask, name + ".gz"))
28
```


Annex B. Dice score evaluation metric Python code

Python code executed to compute the Dice Scores, used to evaluate the overlap between the ground truths and the corresponding predicted masks by both, the localization_vertebra model and the re-trained localization_vertebra model.

Import libraries

```
In [1]: 1 import os
        2 import numpy as np
        3 import nibabel as nib
```

1. Create a function to compute the Dice score.

The following function is available on VerSe's GitHub and has been adapted for use in this project

```
In [2]: 1 def compute_dice (mask1, mask2, empty_score = 1.0):
        2     """
        3     Function that computes the Dice score, which measures the similarity or
        4     overlap between two sets of data.
        5     Dice score ranges from 0, no overlap, to 1, perfect overlap.
        6
        7     mask1 and mask2 are the parameters of the function, corresponding to
        8     the ground truth and the predicted mask by the model, or vice versa,
        9     the order is irrelevant.
        10
        11     The function returns the the Dice score as a float.
        12     """
        13
        14     mask1 = np.asarray(mask1).astype(np.bool) # asarray = convert input to a Numpy array
        15     mask2 = np.asarray(mask2).astype(np.bool) # astype = convert Numpy array to boolean
        16
        17     if mask1.shape != mask2.shape: # The two images must have the same shape to apply the function 'np.logical_and'
        18         raise ValueError("mask1 and mask2 must have the same shape")
        19
        20     mask_sum = mask1.sum() + mask2.sum()
        21
        22     if mask_sum == 0: # When both mask arrays are empty, the function returns the empty_score value
        23         return empty_score
        24
        25     # DICE SCORE
        26     overlap = np.logical_and(mask1, mask2) # Compares element by element of the two arrays and returns a boolean
        27     dice_score = 2*overlap.sum() / mask_sum # Mathematical expression to compute the Dice score
        28
        29     return dice_score
        30
```

2. Take all the Nifties inside the folder where the GROUND TRUTHS are stored and add them to a list.

To obtain all the ground truths of this folder, the following pre-processing techniques have been applied to the initial Materialise Mimics masks:

- Wrap (Materialise 3-matic)
- Erosion (Python)
- Closing smoothing method, fill holes (3D Slicer)

70% of these ground truths, corresponding to the train set, have been used in the re-training phase of the localization_vertebra model.

```
In [3]: 1 folder_grounds_path = r"C:\Users\marta\Desktop\ground_truths"
```

```
In [4]: 1 ground_truth_masks = [] # Empty List where the Loaded NIFTI ground truths are stored
        2
        3 for mask in os.listdir(folder_grounds_path): # Lists all files in the folder
        4     if mask.endswith('.nii.gz'): # To ensure the files have the extension '.nii.gz'
        5         ground_filepath = os.path.join(folder_grounds_path, mask)
        6         # To concatenate in the 'folder_path' the path segment 'mask'
        7         ground_mask = nib.load(ground_filepath) # Load a mask of the specified path
        8         ground_truth_masks.append(ground_mask) # Add the Loaded masks to the empty List created
        9
        10 print(f"Total number of GROUND TRUTHS loaded: {len(ground_truth_masks)}")
```

3. Take all the Nifties of the folder where the PREDICTED MASKS with the pre-trained model from teh Radiology application are stored and add them to a list.

The predicted masks inside this folder have been obtained with the pre-trained localization_vertebra model from the Radiology application.

```
In [5]: 1 folder_masks_path = r"C:\Users\marta\Desktop\predicted_masks_model"

In [6]: 1 predicted_masks_model = [] # Empty List where the Loaded NIFTI ground truths are stored
2
3 for mask in os.listdir(folder_masks_path): # Lists all files in the folder
4     if mask.endswith('.nii.gz'): # To ensure the files have the extension '.nii.gz'
5         predicted_filepath = os.path.join(folder_masks_path, mask)
6         # To concatenate in the 'folder_path' the path segment 'mask'
7         predicted_mask = nib.load(predicted_filepath) # Load a mask of the specified path
8         predicted_masks_model.append(predicted_mask) # Add the Loaded masks to the empty list created
9
10 print(f"Total number of PREDICTED MASKS with the PRE-TRAINED VERSE MODEL loaded: {len(predicted_masks_model)}")
```

4. Take all the Nifties of the folder where the PREDICTED MASKS with the re-trained model are stored and add them to a list.

Inside the folder *predicted_masks_re_model* there are the predicted masks with the re-trained localization_vertebra model generated, which has been re-trained with 70% of the ground truth masks (train set) of the *ground_truth* folder.

```
In [7]: 1 folder_retrained_path = r"C:\Users\marta\Desktop\predicted_masks_re_model"

In [8]: 1 predicted_masks_re_model = [] # Empty List where the Loaded NIFTI ground truths are stored
2
3 for mask in os.listdir(folder_retrained_path): # Lists all files in the folder
4     if mask.endswith('.nii.gz'): # To ensure the files have the extension '.nii.gz'
5         predicted_re_filepath = os.path.join(folder_retrained_path, mask)
6         # To concatenate in the 'folder_path' the path segment 'mask'
7         predicted_re_mask = nib.load(predicted_re_filepath) # Load a mask of the specified path
8         predicted_masks_re_model.append(predicted_re_mask) # Add the Loaded masks to the empty list created
9
10 print(f"Total number of PREDICTED MASKS with the RE-TRAINED MODEL loaded: {len(predicted_masks_re_model)}")
```

- All ground truths are in the *ground_truth_masks* list.
- All predicted masks with the pre-trained model from VerSe are in the *predicted_masks_model* list.
- All predicted masks with the re-trained model are in the *predicted_masks_re_model* list.

Ground truths and all predicted masks are in NIFTI file format (.nii.gz) and have the **same** corresponding filename for each subject.

5. Compute Dice score between the ground truths (*ground_truth_masks*) and the predicted masks with the pre-trained model from VerSe (*predicted_masks_model*).

The ground truths and all the masks predicted are greyscale images, with a different intensity value for the background and also for each of the vertebrae.

To compute the overlap (Dice score) between these two sets of data, the images must have the background pixels set to a particular value, and the rest of pixels corresponding to all vertebrae set to a different but equal value. The following function created binarizes a greyscale image using a given threshold.

```
In [9]: 1 def binary_mask(image, threshold = 0): # The threshold is defined at 0 as in the ColorTable used in 3D Slicer the
2     # intensity 0 was assigned to the background and intensities 1-24 to the vertebrae
3     """
4     'image': grey scale image
5
6     Pixels with intensity values greater than the threshold indicated in
7     the parameters are set to 255 (white) and those below the threshold
8     are set to 0 (black).
9
10    The function returns a binary image with the data type 'uint8',
11    representing the pixel values of the greyscale image.
12    """
13
14    binary_image = np.where(image > threshold, 255, 0).astype(np.uint8)
15
16    return binary_image
```

```
In [10]: 1 dice_scores = [] # Empty List where the Dice scores for each pair of masks are stored
2
3 for mask1, mask2 in zip(ground_truth_masks, predicted_masks_model):
4     # 'zip' function pairs each mask from 'ground_truth_masks' with the corresponding mask in predicted_masks_model
5
6     grounds = mask1.get_fdata() # Call the get_fdata() method to get the array data of the Loaded masks
7     predicted = mask2.get_fdata()
8
9     binary_ground = binary_mask(grounds) # Apply the binary_mask function created to the pair of masks Loaded
10    binary_predicted = binary_mask(predicted)
11
12    dice = compute_dice(binary_ground, binary_predicted) # Compute the dice score
13    dice_scores.append(dice) # Store the dice score for each pair in the empty List created
14
15
16 for i, score in enumerate(dice_scores):
17     print(f'Dice score for pair {i+1}: {score:.2f}')
18
19
20 average_dice_scores = np.mean(dice_scores) # Compute the average Dice score for all pairs
21 print(f'Average Dice score: {average_dice_scores:.2f}')
```

6. Compute Dice score between the ground truths (*ground_truth_masks*) and the predicted masks with the re-trained model (*predicted_masks_re_model*).

```
In [11]: 1 dice_scores_re = [] # Empty List where the Dice scores for each pair of masks are stored
2
3 for mask1, mask2 in zip(ground_truth_masks, predicted_masks_re_model):
4     # 'zip' function pairs each mask from 'ground_truth_masks' with the corresponding mask in predicted_masks_re_model
5
6     grounds = mask1.get_fdata() # Call the get_fdata() method to get the array data of the Loaded masks
7     predicted_re = mask2.get_fdata()
8
9     binary_ground = binary_mask(grounds) # Apply the binary_mask function created to the pair of masks Loaded
10    binary_predicted_re = binary_mask(predicted_re)
11
12    dice = compute_dice(binary_ground, binary_predicted_re) # Compute the dice score
13    dice_scores_re.append(dice) # Store the dice score for each pair in the empty List created
14
15
16 for i, score in enumerate(dice_scores_re):
17     print(f'Dice score for pair {i+1}: {score:.2f}')
18
19
20 average_dice_scores_re = np.mean(dice_scores_re) # Compute the average Dice score for all pairs
21 print(f'Average Dice score: {average_dice_scores_re:.2f}')
```