Automatic Segmentation and Anatomical Labeling of the Spine in Small Field of View MRI Scans

Master Thesis

Brutenis Gliwa

Supervisors:

Prof. Dr. Martin Becker
Institute of Visual and Analytic Computing
University of Rostock

9 November 2023

Universität Rostock

Abstract

Segmentation and labeling of vertebrae and intervertebral discs (IVDs) in magnetic resonance (MR) images plays a crucial part in automated disease diagnostics and the collection of medical statistics. However, these tasks present significant challenges due to the subtle differences among vertebrae and the substantial variations observed in vertebrae across different patients, including the variability in the number of vertebrae a patient has. These challenges are further compounded by the fact that, due to cost and time intensity, most magnetic resonance imaging (MRI) scans only capture a small field of view (FOV) of the spine.

This thesis presents a novel three-step pipeline for vertebra segmentation and labeling in small FOV MRI scans. The proposed pipeline consists of three main steps: two-class segmentation, instance separation, and anatomical labeling of vertebrae and IVDs. Multiple solutions are explored for each step, with the 2D slice-wise U-Net emerging as the most effective method for both segmentation stages. In the context of this study, subset accuracy is a measure of how precisely our pipeline can identify and label all visible vertebrae in a small FOV MRI scan. Using the pipeline, we achieved a subset accuracy of 85.5%, 92.6% and 94.4% for small FOV MRI scans with 5, 10 and 15 visible vertebra out of a possible 25. A Dice similarity coefficient of 0.799, 0.847 and 0.875 was achieved for the same FOV sizes. Furthermore, our pipeline enables the automatic generation of statistics related to lumbarization (the presence of an additional vertebra, resulting in 26 total vertebrae) and sacralization (the absence of a vertebra, resulting in 24 total vertebrae). Analyzing our dataset of 10,833 patients, we found that 710 patients (6.6%) exhibited lumbarization and 393 patients (3.6%) exhibited sacralization, findings that could have significant implications for understanding spinal variations among the population.

Statement of Originality

I declare that this thesis is the product of my own original work and has not been submitted in similar form to any university institution for assessment purposes. All used external sources have been indicated as such and have been cited in the bibliography.

Rostock, 9 November 2023

Acronyms

CNN convolutional neural network

CT computed tomography

DSC Dice similarity coefficient

FOV field of view

GNN graph neural network

GT ground truth

IoU intersection over union

IVD intervertebral disc

LSTV lumbosacral transitional vertebra

MLP multilayer perceptron

MR magnetic resonance

MRI magnetic resonance imaging

MSE mean square error

NN neural network

PCA principal component analysis

Contents

1. Introduction

The human spine, a complex anatomical structure with a vital role in our daily lives, has been a subject of intense study and research [1]. The ability to accurately segment and label the spine is crucial in various medical applications, including diagnosis, treatment planning, and surgical guidance [2, 3, 4]. However, the intricate nature of the spine, coupled with the variability in its shape and appearance, makes this task challenging. This thesis aims to develop novel techniques for spine segmentation and labeling, leveraging advanced machine learning algorithms to overcome these challenges. The goal is not only to enhance the accuracy of these processes but also to contribute to improved patient outcomes in spinal care.

Section 1.1 provides the motivation for the thesis. In Section 1.2 the problem of anatomical labeling is defined formally. In Section 1.3 the methodology will be briefly introduced. Finally, in Section 1.4 an overview of the rest of the thesis is given.

1.1. Importance of Spine Segmentation in the Medical Field

Spine segmentation is essential for automated analysis of the spine, for example in fracture detection [2] or disease diagnostics such as scoliosis [3]. It has been shown that the majority (87%) of asymptomatic fractures are under-reported by radiologists [5], automatic segmentation combined with other techniques could greatly aid in the detection of such fractures [6, 7]. Spine segmentation is also used for surgery, such as computer-assisted screw trajectory planning for vertebrae [4] or for planning of a vertebrectomy (surgical removal of vertebra) [8]. Furthermore, spine issues are very common, for example, lower back pain has been found to be the most common condition in terms of disability in the Global Burden of Disease 2010 Study [1], and sixth most common in terms of overall burden out of 291 conditions. Another study reported that 61.3% (N=5009) of people in Germany have reported back and neck pain in the last 12 months [9]. Therefore, spinal issues are both common and can be aided with automatic segmentation.

Magnetic resonance imaging (MRI) has emerged as a leading modality for imaging the spine due to its superior soft tissue contrast and non-ionizing radiation [10, 11]. However, analyzing magnetic resonance (MR) images of the spine can be challenging due

to factors such as partial volume effects, intensity inhomogeneity, and the presence of pathology [12]. For instance, partial volume effects can lead to mixed signals in a single voxel, making it difficult to assign it to a specific tissue type. Intensity inhomogeneity, or variations in intensity within an image, can complicate the task of distinguishing between different tissues. Furthermore, the presence of pathology can alter the appearance of tissues in unpredictable ways, adding another layer of complexity to the analysis. Automated methods for spine segmentation and vertebra labeling have the potential to greatly improve efficiency and consistency in these workflows [12]. In particular, methods that can handle small field of view (FOV) MR images - where only part of the spine is visible - are of great interest. Small field of view (FOV) MRI scans are the majority of scans due to the cost and time intensiveness of MRI scans [13]. This thesis aims to develop and evaluate multiple methods for anatomical labeling, with a focus on leveraging advanced machine learning techniques such as convolutional neural networks and deep learning algorithms, which have shown promise in handling complex imaging data.

1.2. Problem Statement - Anatomical Labeling

This section formally defines the task of anatomical labeling in small field of view (FOV) MRI scans. To achieve this MRI images are defined, including small FOV MRI scans.

Let $I \in \mathbb{R}^{w \times d \times h}$ be a MRI image with width $w \in \mathbb{N}$, depth $d \in \mathbb{N}$ and height $h \in \mathbb{N}$. Width is in the sagittal axis (side-to-side), depth is in the frontal axis (front-to-back) and height is in the transverse axis (top-down). A typical complete MRI scan has a shape of $20 \times 400 \times 1000$. Anatomical labeling is a function which maps the image $I \in \mathbb{R}^{w \times d \times h}$ to a set of anatomical labels 0 to 49:

$$L(I^{w \times d \times h}) \rightarrow \{0, 1, 2, ..., 49\}^{w \times d \times h} \tag{1.1}$$

The odd labels $\{1, 3, ..., 49\}$ represent the vertebrae {**C2**, **C3**, ..., **S1**}, whereas the even labels $\{2, 4, ..., 48\}$ represent the intervertebral discs (IVDs) {**C2-C3**, **C3-C4**, ..., **L6-S1**}, and finally 0 represents background: neither vertebra nor IVD. A typical complete MRI will have 45, 47 or 49 unique non-background labels, due to the difference in the number of vertebrae and IVDs (see Section 2.1 for further details about spinal anatomy).

In the context of this thesis, we are particularly interested in small field of view (FOV) MRI images of the spine. FOV for MRI images defines how much of a certain region of the body is visible, in this case the spine. A small FOV, in the context of this thesis, indicates that significant portions of the spine are not visible. Formally, a small FOV MRI image I' is an image, whose correct anatomical labeling contains significantly less unique labels than a complete MRI scan. In this thesis we will use three FOV sizes in

order to evaluate our methods: with 5, 10 and 15 visible vertebrae, or equivalently 10, 20 and 30 unique visible labels. This corresponds to roughly 20%, 40% and 60% of the spine in terms of visible vertebrae, respectively.

In conclusion, the goal of anatomical labeling in small FOV MRI images is to assign a label to each voxel of an MRI image, in which only a subset of the spine is visible.

1.3. Proposed Methodology

In this thesis various approaches are proposed in order to obtain a segmented MR volume with the labeling for each vertebrae as well as IVD, given a small FOV MR image. The main challenge is the incomplete nature of the target images, as the labeling of the vertebra and IVDs is substantially more difficult in comparison to complete images. This is because in complete images once the vertebrae have been correctly segmented and separated it is trivial to assume that the uppermost vertebra is **C2**, all following IVDs and vertebrae can be inferred.

Given the difficulty of this task, the dataset of 162 MRI scans, which was manually annotated by experts [14], was not enough to create an adequate model to solve the problem of labeling in small FOV MRI images. Therefore, the following methodology was split into multiple parts in order to be able to artificially create ground truth data of small FOV MRI images.

The segmentation and labeling process can be divided into three steps: (1) initial semantic segmentation of the vertebrae and IVDs, (2) an instance separation step which then separates each vertebra and IVD into distinct objects, (3) and a third step which labels each instance with its correct anatomical label such as **T2** or **L4-L5**. For each of these steps multiple approaches have been considered, especially for step 3. In the following we briefly describe the best approach for each step.

In the semantic segmentation step a U-Net was used in order to segment the MRI slice-wise into 2 classes: vertebrae and IVDs. In the instance separation step the IVDs were separated with the connected components algorithm, then the vertebrae were separated by projecting a plane along each IVD using principal component analysis (PCA) and using the planes as separators. For anatomical labeling again a U-Net was used, in this case it was trained to segment 49 classes, depicting each individual vertebra and IVD. This U-Net was trained on the larger dataset of 10,833 patients, the first two pipeline steps were necessary to create ground truth data on this large dataset.

1.4. Thesis Structure

This thesis is structured to provide a comprehensive exploration into the field of spine segmentation, detailing both the theoretical background and practical approaches. Following this introductory section, the thesis is organized as follows:

Section 2 presents a detailed background necessary for understanding the broader context of this work. It covers key concepts about anatomy, various metrics and losses used in our methodology, and an overview of different neural network architectures relevant to the research presented in this thesis.

Section 3 reviews the related work in the field. It offers a critical analysis of previous methods in localization, segmentation, and anatomical labeling, thereby situating our work within the current research landscape. There are many methods for anatomical labeling, however, they differ in important aspects: either in the dataset type (computed tomography (CT) instead of MRI), or use only a subset of MRIs, making a direct comparison difficult.

Section 4 describes the data used in this thesis. This section details the data sourcing, comparison to other similar dataset, and gives some examples of normal and erroneous MRI scans. It showcases the large dataset of 10,833 patients, where for each the entire spine is visible.

In Section 5 we delve into the vertebra segmentation and labeling pipeline. This extensive section explains our methodology, from the initial segmentation of vertebrae and intervertebral discs to the final steps of anatomical labeling. It also evaluates the combination of methods used and discusses the complete end-to-end multiclass segmentation process.

Section 6 presents the results of our research. This section is dedicated to the analysis of outcomes from the various steps in the segmentation and labeling pipeline, including a discussion on special cases like sacralization and lumbarization.

Section 7 discusses the implications, strengths, and limitations of our study. It contextualizes our findings within the broader field, offering insights into their significance and potential applications, along with suggestions for future research avenues.

Finally, Section 8 concludes the thesis. It summarizes the key findings, reflects on the research contributions, and offers closing remarks.

References and any additional supporting materials are included at the end of the thesis in the Bibliography and Appendices, respectively.

2. Background

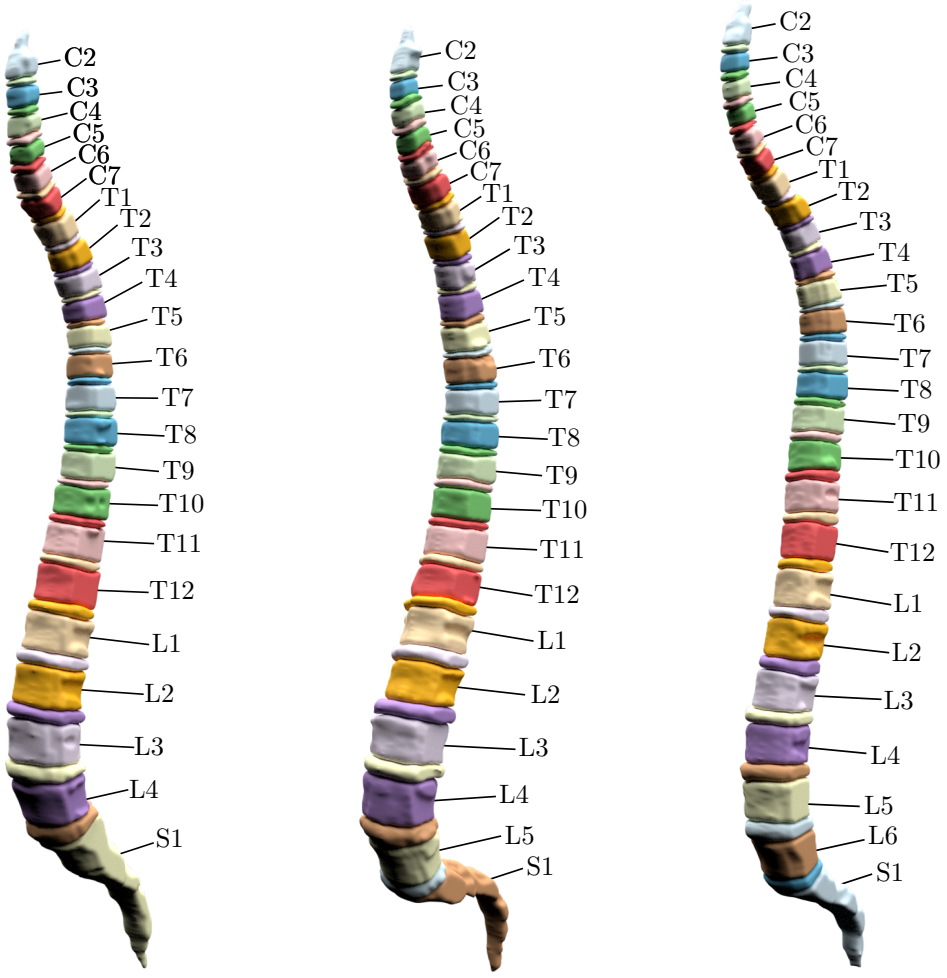
Section 2.1 provides anatomical background about the spine, vertebrae and IVDs. It also explains why around 10% of the population has either one more or one less vertebra, which puts one of the results of the thesis in context. In Section 2.2 the connected components algorithm is described. It is an important algorithm in many parts of the thesis including in the post-processing of the segmentation (Section 5.2.4), instance separation via connected components (Section 5.3.1), and instance separation by splitting along IVDs (Section 5.3.2). In Section 2.3 the metrics subset accuracy, Dice similarity coefficient (DSC) and intersection over union (IoU) are discussed. All three of these metrics are essential in understanding and comparing the results of this thesis, as all results in Section 6 are provided using either of these metrics. Finally, in Section 2.4.1 the basics of neural network (NN) are introduced. The most approaches introduced in this thesis rely on some sort of neural network, understanding how these work and their limitations is essential. Furthermore, neural network (NN) architectures such as convolutional neural networks (CNNs) (Section 2.4.2), graph neural networks (GNNs) (Section 2.4.3) and U-Nets are introduced. CNNs are foundational models for segmentation, and are the core underlying functionality for U-Nets, which are the main segmentation architecture used throughout this thesis for segmentation. GNNs are essential to an alternative approach to anatomical labeling, presented in Section 5.4.3.

2.1. Anatomy

The human vertebral column is typically composed of 33 vertebrae. In adults, however, nine of these vertebrae undergo fusion: the inferior four integrate to form the coccyx (tailbone), while the five immediately superior to them coalesce into the sacrum. This results in an effective count of 26 distinct vertebrae, with the topmost 24 being referred to as pre-sacral or moving vertebrae. Three example spines can be seen in Figure 2.1. In the following the distinction between the vertebrae will be briefly introduced. For a more comprehensive in-depth look into the anatomy of the human spine we refer to *Functional Anatomy of the Spine* by Oliver and Middleditch [15] and *Spinal Anatomy* by Vital and Cawley [16].

These moving vertebrae can be systematically categorized:

- Cervical: **C1** to **C7**



(a) (b) (c)

Figure 2.1.: **3D Anatomical Labeling Examples**. Three example spine segmentations created using the methods presented in this thesis. (a) shows a spine where **L5** has sacralized, whereas (c) shows a spine where **S1** has lumbarized. (b) shows a normal spine. The labels for the vertebrae are shown, the labels for the IVDs can be inferred by the adjacent vertebrae. The colors represent the different objects starting at the top counting down, this makes a comparison between different images easier. The 3D objects have been smoothed for visual clarity.

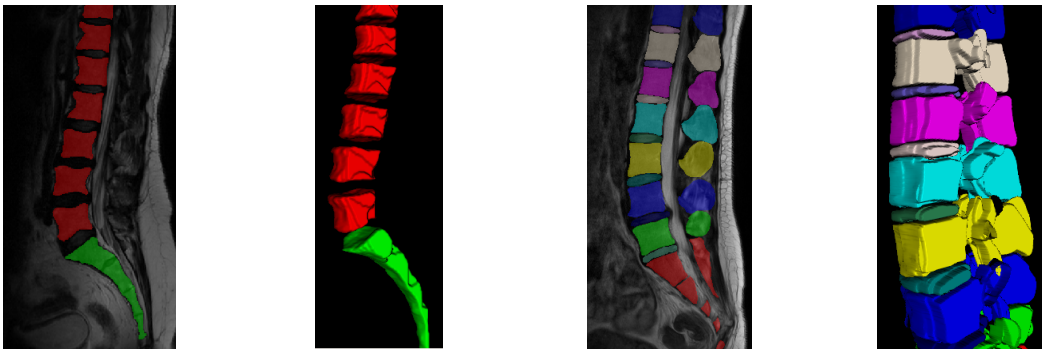
- Thoracic: **T1** to **T12**
- Lumbar: **L1** to **L4**/**L5**/**L6**

In this thesis, the sacrum will be denoted as a single vertebra, represented by **S1**. Whereas the coccyx (tailbone) will not be further discussed, as it is fairly small and often not depicted in the dataset provided for this thesis.

An IVD is conventionally positioned between each vertebral pair (except for **C1** and **C2**, which lack an IVD between them). The nomenclature for these discs is derived

from the vertebrae they adjoin, such as C2-C3, T9-T10, or T12-L1. In this thesis vertebrae will be marked with orange (e.g. C2) and IVDs with blue (e.g. C2-C3).

Most vertebrae (C2 to L5) consist of a bony structure which surrounds the spinal canal. At the anterior there is a larger volume called the vertebral body which has a roughly cylindrical shape with a flat top and bottom (see Figure 2.2b). Connected to the vertebral body at either side is a bony ring-like structure surrounding the spinal canal (see Figure 2.2d). Connected to it posteriorely and transversely are various bony protrusions such as the spinous process and transverse process. In this thesis vertebra will be used interchangeably with vertebral body, because the dataset used in this thesis only has a segmentation for vertebral bodies (Figure 2.2a). For a comparison between a segmentation for only a vertebral body and the full vertebra see Figure 2.2.



(a) (b) (c) (d)

Figure 2.2.: **Comparison Between Segmentations of Full Vertebra and Vertebral Body Only**. The left two images (a) and (b) show segmentations for only the vertebral body (our dataset, further described in Section 4). The right two images (c) and (d) show a segmentation for the full vertebra as well as IVDs (from the SpineSegmentationChallenge dataset [17, 18]).

There are seven cervical vertebrae, denoted as C1 to C7, C1 being the topmost vertebra closest to the skull. C1 is called "atlas" and is different from all other vertebrae in that it does not have a vertebral body. Instead, C2 has a larger vertebral body which protrudes up to C1, allowing C1 to pivot around C2, C1 therefore being named "axis".

The thoracic vertebrae, denoted as T1 to T12 are the next twelve vertebrae after the cervical vertebrae. They are characterized by each vertebra being connected to a rib through the costovertebral joint.

The lumbar vertebrae, denoted as L1 to L4, L5 or L6, are the next four to six vertebrae. They do not have a rib connection and are the largest vertebrae.

In healthy humans, the number of vertebrae in the upper half of the spine tends to remain relatively consistent. However, variations are more common in the lumbar region. Specifically, the lumbar region might not always comprise the standard five vertebrae. Two conditions representing such variations are sacralization and lumbarization.

Sacralization occurs when one lumbar vertebra fuses with the sacrum, resulting in a total of only four lumbar vertebrae (can be seen in Figure 2.1a). Conversely, lumbarization arises when one vertebra does not fuse with the sacrum as it typically would, leading to an extra lumbar vertebra, designated as **L6**, and thus a total of six lumbar vertebrae (can be seen in Figure 2.1c). Sacralization and lumbarization are observed in approximately 4.8% and 4.7% of the population, respectively (see Section 6.4 for further discussion about these). The remaining 90.5% contain five lumbar vertebrae, with few exceptions.

2.2. Connected Components Algorithm

The connected components algorithm is a fundamental method utilized in image processing and computer vision for object segmentation. It delineates and labels continuous regions of an image where the pixel values meet a specified criterion, ensuring every distinct object receives a unique identifier. For our study, the algorithm is indispensable for segmenting various anatomical structures, namely the IVDs and vertebrae, from the 3D medical images. Crucially, the algorithm operates within a 26-neighborhood in 3D, ensuring comprehensive connectivity analysis.

It works by iterating over every voxel in the image, every time a voxel with non-zero value is found, a flood fill algorithm is run for that voxel, which finds any 26-neighborhood adjacent voxels sharing the same value recursively. All those values are marked with the same label and are not considered for further flood fill calls. Then the algorithm continues this for all following voxels, the algorithm is defined in Algorithm 0. For further details we refer to Zhao et al. [19].

2.3. Evaluation Metrics

Evaluating and training machine learning models, particularly in the realm of image processing, necessitates the use of various metrics and loss functions. These quantitative measures allow for the assessment of model performance and direct optimization during training. This section provides an overview of several pivotal metrics and losses used in classification and segmentation tasks. Specifically, Section 2.3.1 delves into accuracy and subset accuracy, Section 2.3.2 explains Dice loss and the Dice similarity coefficient (DSC), and finally, Section 2.3.3 covers the intersection over union (IoU) metric. These three metrics were used in order to evaluate the models presented here. Cross entropy loss was also rarely used in order to train some machine learning models, however, not enough warranting an explanation in this thesis, therefore we refer to Goodfellow et al. [20].

Algorithm 1 3D Connected Components

```
1: procedure ConnectedComponents3D(volume)
2:     label ← 0
3:     labels ← initialize 3D array of 0's with size of volume
4:     for x ← 1 to width(volume) do
5:         for y ← 1 to height(volume) do
6:             for z ← 1 to depth(volume) do
7:                 if volume[x][y][z] ≠ 0 and labels[x][y][z] = 0 then
8:                     label ← label + 1
9:                     FloodFill(volume, labels, x, y, z, label)
10:                end if
11:            end for
12:        end for
13:    end for
14:    return labels
15: end procedure
16: procedure FloodFill(volume, labels, x, y, z, label)
17:    if x, y, z is out of bounds or volume[x][y][z] = 0 or labels[x][y][z] ≠ 0 then
18:        return
19:    end if
20:    labels[x][y][z] ← label
21:    for each (dx, dy, dz) ∈ {−1, 0, 1}^3 do
22:        FloodFill(volume, labels, x + dx, y + dy, z + dz, label)
23:    end for
24: end procedure
```

2.3.1. Accuracy and Subset Accuracy

Accuracy is one of the primary metrics used to evaluate the performance of classification models. Specifically, in a classification context, accuracy represents the proportion of instances that are classified correctly.

For N instances:

$$\text{Acc} = \frac{\text{Number of correctly classified instances}}{N} \tag{2.1}$$

In tasks where an instance can belong to multiple classes (multi-label classification), a stricter metric often used is the *subset accuracy* or *exact match accuracy*. This metric requires that for an instance to be considered correctly classified, every individual class must be predicted correctly. Any misclassification in the classes for an instance results in the instance being considered incorrect. Formally, for N instances:

$$\text{Subset accuracy} = \frac{\text{Number of instances with all labels correctly classified}}{N} \tag{2.2}$$

Subset accuracy is a strict metric, making it especially challenging to maximize in tasks with a large number of classes or in applications where each instance can belong to multiple classes.

However, while accuracy and subset accuracy provide intuitive measures of overall performance, they may not capture the model's performance nuances, especially in imbalanced datasets. Therefore, they're often used alongside other metrics to get a comprehensive understanding of a model's performance.

2.3.2. Dice Similarity Coefficient (DSC) and Dice Loss

The Dice coefficient, also known as the Sørensen–Dice index, or F1 score, is a statistic used to gauge the similarity between two sets [21, 22]. It is commonly used in the field of medical imaging to measure the similarity between the predicted segmentation and the ground truth. For two sets A and B, the Dice coefficient is given by:

$$D(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{2.3}$$

Where $|A \cap B|$ is the cardinality of the intersection of sets A and B, and $|A|$ and $|B|$ are the cardinalities of sets A and B, respectively.

Binary Dice Loss

In the context of image segmentation, let's consider the predicted segmentation as set P and the ground truth as set G. Each set contains pixels that are either part of the object of interest (value 1) or the background (value 0). The Dice coefficient can then be reformulated in terms of these binary pixel values:

$$D(P, G) = \frac{2\sum_i^N P_i G_i}{\sum_i^N P_i + \sum_i^N G_i} \tag{2.4}$$

Where N is the total number of pixels, and P_i and G_i are the pixel values at location i for the predicted and ground truth images, respectively.

To use the Dice coefficient as a loss function for training a neural network, the Dice Loss is defined as:

$$L_{\text{Dice}} = 1 - D(P, G) \tag{2.5}$$

A smaller Dice Loss indicates a better overlap between the predicted segmentation and the ground truth. Thus, during training, the aim is to minimize this loss value to improve the model's segmentation performance.

Multiclass Dice Loss

For binary segmentation tasks, the Dice coefficient measures the overlap between the predicted segmentation and the ground truth for a single class. However, in multiclass segmentation tasks, where an image may contain multiple regions of interest, the Dice coefficient needs to be computed for each class separately.

Given C classes in an image, for each class c, the Dice coefficient is:

$$D_c(P, G) = \frac{2\sum_i^N P_{i,c} G_{i,c}}{\sum_i^N P_{i,c} + \sum_i^N G_{i,c}} \tag{2.6}$$

Where $P_{i,c}$ and $G_{i,c}$ are the pixel values at location i for the predicted and ground truth images, respectively, for class c. N is the total number of pixels.

The average Dice coefficient across all classes can be taken as:

$$D_{\text{avg}}(P, G) = \frac{1}{C} \sum_{c=1}^{C} D_c(P, G) \tag{2.7}$$

To utilize the Dice coefficient as a loss function for multiclass segmentation in neural networks, the Multiclass Dice Loss is:

$$L_{\text{Dice, multiclass}} = 1 - D_{\text{avg}}(P, G) \tag{2.8}$$

Similar to the binary case, the goal during training is to minimize this multiclass Dice Loss to achieve a segmentation result that overlaps well with the multiclass ground truth across all classes.

2.3.3. Intersection over Union (IoU) and Jaccard Loss

intersection over union (IoU), also known as the Jaccard index [23], is a widely-used metric to evaluate the overlap between two regions. In the context of image segmentation, it quantifies the overlap between the predicted segmentation and the ground truth.

Given two sets A and B, the IoU is defined as:

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.9}$$

In binary image segmentation, where each pixel in the predicted segmentation P and ground truth G is either part of the object (1) or the background (0), the IoU can be expressed in terms of pixel counts:

$$\text{IoU}(P, G) = \frac{\sum_i^N P_i \cdot G_i}{\sum_i^N \max(P_i, G_i)} \tag{2.10}$$

For multiclass segmentation with C classes, the IoU can be computed for each class c independently:

$$\text{IoU}_c(P, G) = \frac{\sum_i^N P_{i,c} \cdot G_{i,c}}{\sum_i^N \max(P_{i,c}, G_{i,c})} \tag{2.11}$$

An average IoU across all classes measures overall segmentation accuracy:

$$\text{IoU}_{\text{avg}}(P, G) = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c(P, G) \tag{2.12}$$

Higher IoU values indicate better overlap between the predicted segmentation and the ground truth, with a maximum value of 1 indicating perfect overlap.

Finally, the Jaccard loss, also known as IoU loss is defined as follows:

$$\text{JaccardLoss}(P, G) = 1 - \text{IoU}_{\text{avg}}(P, G) \tag{2.13}$$

2.4. Neural Network Architectures

This section introduces neural networks in Section 2.4.1, which serve as the foundational models for this thesis. CNNs, introduced in Section 2.4.2, are the underlying neural network structure of the U-Net, which was the best performing architecture in this thesis. U-Nets are introduced in Section 2.4.4. GNNs, central to an alternative approach for this thesis, is introduced in Section 2.4.3.

2.4.1. Basics of Neural Networks

Neural networks (NN) are computational models, developed to mimic the structure and function of the human brain, and are designed to discern patterns and derive predictions from input data. Their strength lies in their ability to model intricate, non-linear relationships in data, which has placed them at the forefront of machine learning and artificial intelligence research. In the following the basics of neural networks are explained, for further details we refer to Goodfellow et al. [20]. This network is also referred as a multilayer perceptron (MLP), which we use as a classifier for anatomical labeling in Section 5.4.2.

Mathematically, an NN can be described as a composition of functions. Given an input vector $\mathbf{x}$, the network computes an output $\mathbf{y}$ via the relationship

$$\mathbf{y} = f_L(\dots f_2(f_1(\mathbf{x}; \mathbf{W}_1); \mathbf{W}_2)\dots); \mathbf{W}_L) \tag{2.14}$$

where L is the number of layers in the network, and $\mathbf{W}_i$ represents the weights of the i^{th} layer.

Constituent Elements of a Fully Connected Neural Network:

- **Neurons (Nodes)**: Each neuron in a layer computes a weighted sum of its inputs and applies an activation function. Formally, given inputs $\mathbf{x}$ and weights $\mathbf{w}$, the output o of a neuron is $o = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$, where σ is the activation function and b is a bias term.
- **Layers**: A fully connected neural network is organized into layers: an input layer, several hidden layers, and an output layer. Every neuron in one layer connects to every neuron in the subsequent layer.
- **Activation Function**: Functions like the sigmoid, tanh, or rectified linear unit (ReLU) introduce non-linearities into the network, empowering it to capture intricate relationships.

Training Process:

Neural networks learn by adjusting their weights based on data. The typical learning cycle involves:

1. Forward propagation of input to produce an output.
2. Calculation of the differentiable loss, a measure of error between predicted and actual outputs, often using metrics like mean square error (MSE).
3. Backward propagation using optimization techniques such as gradient descent to adjust the weights in a direction that minimizes the loss.
4. Iterative optimization until a satisfactory performance level is achieved.

These basic building blocks form the basis for the following sections as well.

2.4.2. Convolutional Neural Networks (CNN)

CNNs are a specialized category of neural networks tailored primarily for image data. Unlike fully connected neural networks where every neuron is connected to every other neuron in the adjacent layers, CNNs exploit spatial hierarchies in the data, allowing them to automatically and adaptively learn spatial features. In the following we introduce the essentials of CNNs, for a more detailed look the book *Deep Learning* [20] is recommended.

Mathematically, a CNN is composed of a series of layers that transform an input (like an image) into an output through a differentiable function. The core difference lies in the utilization of convolution operations instead of standard matrix multiplications in at least one of its layers.

Constituent Elements of a CNN:

- **Convolutional layer**: The primary operation of this layer involves sliding a filter or kernel over the input data (such as an image) to produce a feature map or activation map. This operation captures local patterns.
- **Pooling (subsampling) layer**: A downsampling operation that reduces the spatial size, thereby reducing computation and helping to make feature representations more robust.
- **Fully connected layer**: Similar to those in standard neural networks, these layers perform classification based on the features extracted by the preceding convolutional and pooling layers.

Training and Application:

Like other neural networks, CNNs learn by adjusting their parameters to minimize the error between the predicted and actual outputs. The main distinction in their training arises from backpropagation through the convolutional layers.

While fully connected networks can struggle with high-dimensional data like images due to the sheer number of parameters, CNNs are designed to mitigate this. By exploiting spatial hierarchies and sharing parameters across space, they require fewer parameters, making them more efficient for image data. This architecture has been pivotal in tasks

like image and video recognition, semantic image segmentation, and even non-visual tasks like natural language processing.

In summary, while both fully connected networks and CNNs have their distinct strengths, the latter's ability to handle high-dimensional data efficiently and its proficiency in extracting spatial features make it an indispensable tool in computer vision and beyond. In this thesis CNNs will be used throughout for segmentation (Section 5.2.2, Section 5.2.3, Section 5.4.4) and for encoding of features (Section 5.4.3).

2.4.3. Graph Neural Networks (GNN)

Graph neural networks (GNN) represent a confluence of graph theory and deep learning, offering an innovative lens through which to understand and process data inherent in graphs. In domains where data is naturally structured in a graph, such as molecules in chemistry or vertebrae in a spine, GNNs have shown exceptional performance due to their ability to capture the intricate relationships within the data. Here a brief introduction is presented for GNNs, for an in-depth look we recommend *Graph Neural Networks: Foundations, Frontiers, and Applications* [24].

A graph in this context is composed of nodes and edges, with nodes representing entities and edges the relationships or interactions between them. In the realm of GNNs, nodes are not just abstract points but are endowed with features that carry complex information, and edges often encapsulate the dynamics of the inter-node relationships.

At the heart of GNNs lies the concept of message passing or information aggregation, where nodes update their states by gathering and synthesizing information from their immediate neighborhood. This is often expressed through:

$$h_v^{(k)} = \text{UPDATE}\left(h_v^{(k-1)}, \text{AGGREGATE}\left(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}\right)\right), \tag{2.15}$$

where $h_v^{(k)}$ represents the hidden state of node v at iteration k, $\mathcal{N}(v)$ is the neighboring nodes of v, and the functions UPDATE and AGGREGATE are optimized through the training process.

Graph convolutional networks, a variant of GNNs, have been particularly successful, leveraging a convolutional approach to update node states—a technique that captures both local structure and global graph properties.

In the specialized field of medical imaging, and more precisely in spine segmentation, GNNs demonstrate a significant advantage. The human spine is a complex anatomical structure consisting of vertebrae and intervertebral discs, which are naturally connected in a sequential manner. This adjacency lends itself well to a graph-based representation,

where each vertebra and disc can be a node within a graph, and the edges represent the physical connections and relative positioning between them.

$$\text{Spine Graph: } S = (V_{\text{spine}}, E_{\text{spine}}) \tag{2.16}$$

where V_{spine} corresponds to the vertebrae and intervertebral discs, and E_{spine} to the connections between them.

By employing GNNs, one can leverage the inherent graph structure of the spine for segmentation tasks. Nodes in the graph can effectively learn and represent the complex features of the spine's anatomy, such as the shape and size of vertebrae, the position of the discs, and their relationships. GNNs are particularly adept at capturing the dependencies and spatial context between different parts of the spine, which is crucial for accurate segmentation.

Furthermore, the ability of GNNs to integrate local node features with global structural information allows them to generalize well across different patients and imaging modalities. This makes GNNs an exceptionally suitable choice for spine segmentation tasks in clinical applications, where precision and adaptability to varied anatomical presentations are of utmost importance.

In conclusion, the utilization of GNNs in spine segmentation underscores the potential of graph-based deep learning models to revolutionize the analysis of complex biological structures. Their ability to capture the nuanced patterns of anatomical connectivity positions them as a cutting-edge tool in medical image analysis, promising advancements in diagnostic precision and patient care.

2.4.4. U-Net Architectures

Introduced by Ronneberger et al. in 2015 [25], the U-Net architecture stands as a pivotal model in biomedical image segmentation. This architecture, notable for its U-shaped structure, combines a contracting path to capture contextual information and a symmetric expanding path for precise localization of features, essential in tasks like medical image analysis.

At its core, the U-Net architecture integrates the conventional approach of a convolutional neural network in its contracting path, consisting of repeated application of convolutions, each followed by a ReLU activation and a max pooling for downsampling. The innovation, however, lies in its expansive path where up-sampling of the feature map occurs, reintroducing the dimensions of the original input image. This path crucially includes skip connections from the contracting path, reintroducing high-resolution features to enable detailed localization.

Beyond its initial form, U-Net has seen various adaptations and enhancements. The extension to 3D U-Net by Cicek et al. [26] allows the architecture to handle volumetric data, essential for modalities like MRI and CT scans in medical imaging. Further adaptations include the integration of residual connections, seen in the Residual U-Net, which facilitates training deeper networks by mitigating the vanishing gradient problem, a common challenge in deep learning models.

The flexibility of U-Net extends to its combination with other advanced neural network architectures. An example is its integration with Mixed Vision Transformers, as cited in [27], showcasing the architecture's versatility and capacity for continual evolution to meet diverse and challenging segmentation tasks.

U-Net's ongoing development and adaptation highlight not just its robustness and effectiveness but also its enduring position as a foundational model in the realm of image segmentation. In this thesis U-Nets are used throughout for segmentation (Section 5.2, Section 5.4.4).

3. Related Work

Existing work regarding vertebra and IVD identification can be grouped into multiple categories: localization, segmentation as well as anatomical labeling. Localization refers to the process of finding the position of each vertebra or IVD, which is most commonly the centroid. Segmentation involves classifying each pixel or voxel of the image as background, vertebra or IVD. Anatomical instance labeling assigns each object it's anatomically accurate label such as **C2** or **T11-12**. Our work would be categorized in the latter category, however the methods presented there build upon the previous sections. In the following sections related work will be discussed based on these categories.

3.1. Vertebra and Intervertebral Disc Localization

During localization the goal is to find the position of the relevant objects. This generally means finding a 2D or 3D coordinate representing the centroid of either the vertebrae or the IVDs, respectively. Many approaches have been tried, such as heatmap-based [28, 29], coordinate-based and graph-based methods.

Zhigang et al. [30] localize vertebra by a three-step process. At first, the best slice is found where the most vertebrae are visible, then a fourth-degree polynomial is fit through the vertebra centers and finally, the edges of the vertebrae are found using a canny edge detector.

Glocker et al. [29] propose a two-stage method using regression forests and hidden Markov models. the regression forests create a probability map, and the hidden Markov models are used in order to fine-tune the centroids. Using this approach they achieve an identification rate of 81%.

Wang et al. [31] propose a four step vertebra localization and identification module achieving an 97.4% identification rate. For vertebra center localization they use a U-Net, followed by spine rectification, which maps all centroids on a single axis. Then the activation functions for each separate vertebra are combined, and finally an anatomically constrained optimization module finds a possible matching.

In our work localization is implied by the segmentation step in the pipeline, taking the average coordinate of the voxels for each separated instance. However, many of these approaches also label the corresponding centroid with it's anatomical label. This means that, the labels could be assigned to the instances returned in our pipeline, replacing step 3.

3.2. Spine Segmentation

The task of segmentation involves classifying each pixel or voxel of the original image into some classes. Specifically in this case, we define two classes of interest, vertebra and intervertebral disc (IVD)s. Segmentation of vertebra and IVDs can be divided into two categories, traditional, deep learning-based and atlas-based [32, 33, 34] approaches.

Atlas-based methods use an atlas - a reference segmentation representing a general spine as well as possible - in order to create a new segmentation by overlaying the atlas on the target. However, simply overlaying one on top of each other is rarely if ever good enough, therefore other approaches are used in order to deform the atlas. In [32] the spinal canal is used to find the IVDs by the intensity profile of the IVDs (an IVD is significantly dimmer in a CT-scan than the vertebrae). In [33] a probabilistic atlas of 50 2D segmentations of the midsagittal-slice are used in order to achieve better results.

Recently, approaches based on deep learning have become popular. This often involves using a well-tested segmentation network such as U-Net [25] or DeepLabV3 [35], and then classifying the segmented parts further. The approach by Streckenbach et al. [14], which uses the exact same dataset as in this thesis, uses a patch-based 3D U-Net [26] in order to segment the entire spine in multiple passes. Most state-of-the art papers in the following two sections are using one of these segmentation networks as their basis.

In conclusion, with the vast amount of available methods it can be claimed that basic two-class segmentation works very well and does not pose many challenges anymore. In Section 6.2 the approach by Streckenbach et al. [14] will be compared with our approach, as it uses the same dataset.

3.3. Anatomical Spinal Instance Labeling

During anatomical instance identification the target is to create segmentation such that each separate vertebra and IVD has its own anatomically correct label. The distinction in comparison to instance segmentation is valuable, as there the target is to only separate the various objects, whereas here the target is to assign the true anatomical labels. In the following a few state-of-the-art spine segmentation approaches are elaborated.

Pang et al. [17] use the segmentation network DeepLabV3+ [35] in order to create an encoding for a GNN. Using multiple graph convolutions they are able to create a prediction for each voxel in the image for each class. For this they use a dataset of 215 MRI scans of the lumbar region. They focus on good performance, and in order to reduce the size of the models they apply a two stage segmentation. The first stage produces a down-sampled segmentation for each sample in 3D. Then a 2D slice-wise network uses the down-sampled segmentation together with the initial image to

produce a full-scale segmentation. A 3D network for direct full-scale segmentation is not feasible, or at least considerably more expensive performance-wise. They achieve a DSC of 89.22% on the vertebra from **T12** to **S1**. It is of note, however, that their dataset does not include any cases where sacralization or lumbarization appears. Furthermore, each of their samples contains the sacrum **S1**, we argue that the GNN in their approach can learn to simply count up from the sacrum in each test case. The problem where neither the topmost (**C2**) or bottommost (**S1**) vertebra appears in the image is considerably more difficult.

Lessman et al. [36] propose an iterative network, which segments the currently visible vertebra in a 3D slding window using a U-Net [25]. This sliding window is moved to the next vertebra, once the current one is completely found. It keeps an instance memory of each vertebra, which forces the network to predict the next vertebra. In order to predict the anatomical label they extend the compression path of the U-Net and reduce it further into a single value between 1 to 24 as a regression task. They argue using a regression results in the network being penalized more the more its prediction deviates from the actual value. Once each vertebra has its preliminary label, they create a final prediction, which is global prediction of the maximum likelihood, such that it makes anatomical sense.

Payer et al. [37] propose a three-step approach for segmentation of each vertebra in the VerSe19 [7, 38, 39] CT-dataset. The three-step approach encompasses (1) localization of the spine, (2) a heatmap-based approach for the localization of each vertebra and (3) segmentation of vertebrae. Localization of the spine is necessary because the VerSe19 dataset includes large portions of data where non-spine body parts are depicted, such as the legs or head. For the localization of the spine they use a U-Net to perform a heatmap regression in order to predict the center coordinate. For the vertebra localization they use their own SpatialConfiguration-Net [40] to predict each center for each vertebra using heatmap regression. For segmentation they use a U-Net which separates a single vertebra from the background as a binary segmentation task. The input is a cropped region around the vertebra, which is concatenated with a Gaussian heatmap, produced by the previous step. This is done for each vertebra, and then remerged into the final image. This approach won the Large Scale Vertebrae Segmentation Challenge 2019[1].

In conclusion, the approach by Payer et al. [37] is likely the most similar to our approach. Our approach lacks the spine localization step, and instead of the heatmap-based labeling, we use a multiclass segmentation for labeling. The dataset is likely more difficult than ours, as it contains more pathologies and often has rather small field of view (FOV). However, the dataset is much smaller and is CT-based, which makes a direct comparison difficult, as CT-datasets usually do not segment IVDs. In terms of datasets, the approach by Pang et al. [17] is more similar, they also have MRI data and a constant FOV of around 10 visible vertebra. However, their FOV is always at the same lumbar position, with **S1** visible in all 215 samples. Furthermore, they do not have a single case of lumbarization or sacralization, which makes labeling much

[1] https://verse2019.grand-challenge.org/

easier. Therefore, our methods are somewhat similar to existing solutions, but due to a unique dataset a direct one-on-one comparison is difficult. Despite this, in Table 6.2 our results will be compared with these methods.

4. Data

In collaboration with the University Medicine Rostock, the primary dataset utilized for this study comprised 10,833 MRI scans of the complete spine. This dataset was assembled explicitly for research endeavors by the German National Cohort[1], ensuring a representative sample of the German population. However, a slight skew is evident toward the elderly demographic, with the mean age of the sample being 52.25 years. In contrast, the average age in Germany between the years 2012-2015 — the period during which these scans were captured — stood at 44.25 years [41]. For further details about the dataset see [14].

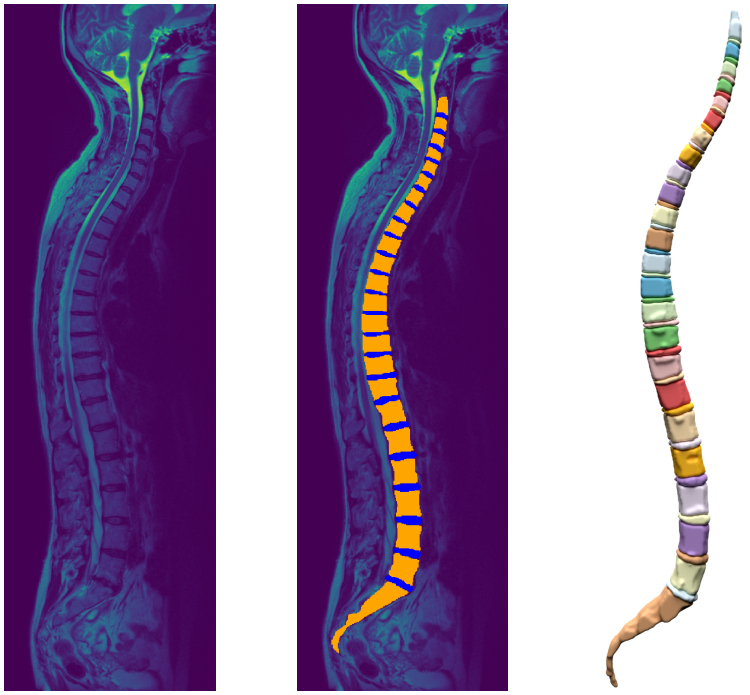


Figure 4.1.: **MRI and Ground Truth**. An example midsagittal MRI slice, it's corresponding manually annotated ground truth, and the ground truth rendered in 3D.

Within the 10,833 scans, a subset of 330 scans possesses manually segmented 3D voxel data for the IVDs and spinal canal. Additionally, 162 of these scans showcase segmented 3D voxel data specific to the vertebral body. This segmentation process was

[1] https://nako.de/

undertaken by a team of medical Ph.D. students affiliated with the University Medicine Rostock. Figure 4.1 presents the middle sagittal slice from a patient's scan, illustrating an example MRI with the masks for the vertebral body and IVDs, as well as a 3D rendering of the annotated data. In our dataset only vertebral bodies were annotated, in other datasets it is common to annotate the entire vertebra (see Section 2.1 for the distinction). This causes some differences in our dataset in comparison to other datasets, most notably C1 is not annotated as it does not have a vertebral body. Therefore when counting vertebra we do not include C1.

The dataset under consideration is exclusively comprised of T2 sagittal scans. T2 MRI scans are distinct in their capability to differentiate between tissues based on their transverse relaxation time, while T1 scans delineate tissues based on their longitudinal relaxation time. This difference in relaxation times facilitates unique contrasts in the resulting images, making T2 scans especially advantageous for visualizing certain anatomical details.

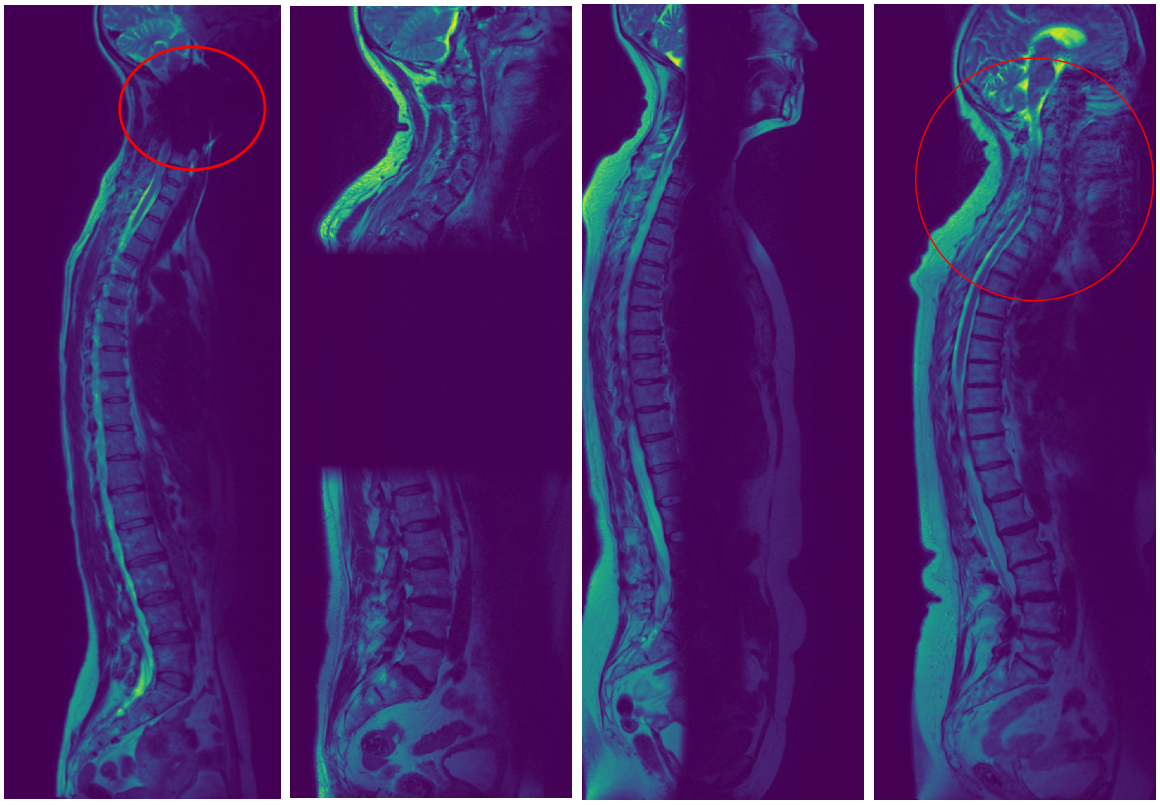
In terms of dimensions, the images in the dataset approximate to a size of $20 \times 400 \times 1000$, with a slice thicknesses of $3.3mm \times 0.8mm \times 0.8mm$ across the sagittal, coronal, and traversal axes, respectively. In contrast, CT scans typically measure $512 \times 512 \times 400 - 1000$. It's noteworthy that one of the axes in CT scans usually greatly varies in scale compared to MRI scans, which hinders the direct comparability between CT and MR data. Furthermore, CT scans work in a fundamentally different way, such that bones are much more visible than soft tissue.

Both the manually truthed as well as the full dataset were split into train and validation sets. The manually truthed set of 162 patients was split into 132/30 train and validation split. The 30 patients in the validation set have been specifically selected, because these patients each were segmented by 3-4 medical specialists, raising the confidence of the segmentation. The entire dataset of 10,833 was also split into train, validation and test. For validation 5% were randomly sampled from the entire dataset, and for the test-set the same 30 from the truthed set were used, resulting in a split of 10261/542/30.

Statistics of Sacralization and Lumbarization in the Dataset

Sacralization and lumbarization are anatomical variations that occur in the lumbosacral region of the human spine. Sacralization refers to the condition where the fifth lumbar vertebra L5 fuses with the first sacral segment S1, essentially becoming part of the sacrum. On the other hand, lumbarization is when the first sacral segment S1 fails to fuse with the rest of the sacrum and functions as an additional, or sixth, lumbar vertebra L6.

In our dataset, we observed these variations in a significant number of patients. Using the methods presented in this thesis, the entire dataset was automatically evaluated. Lumbarization was present in 710 out of 10833 patients, representing approximately



(a) Metallic vertebra interference (b) Missing data in horizontal axis (c) Missing data in vertical axis (d) Blur due to movement

Figure 4.2.: **Difficult Data**. Multiple examples of erroneous or difficult data

6.6% of the total population studied. This means that these individuals had an additional lumbar vertebra due to the non-fusion of S1. Sacralization, on the other hand, was less common, observed in 393 out of 10833 patients, or about 3.6% of the population. These individuals lacked a fifth lumbar vertebra due to its fusion with S1. These results are further discussed and compared with medical studies in Section 6.4.

In literature, when it is unclear whether the last vertebra is sacralized or lumbarized, the term lumbosacral transitional vertebra (LSTV) is commonly used [42, 43]. This term encompasses both sacralization and lumbarization, acknowledging the variability in this region of the spine.

5. Vertebra Segmentation and Labeling Pipeline

Segmentation and labeling of the vertebrae are critical steps in the processing of spinal medical images. The primary goal of these steps is to accurately identify and label each vertebra in the spine, facilitating subsequent analyses such as structural assessment, pathological diagnosis, and surgical planning. Achieving precise segmentation and labeling is essential for ensuring the accuracy of any derived metrics or insights.

The intricacies of spine segmentation and labeling arise from a myriad of challenges, such as:

- **Small FOV MRI scans**: Due to the time-intensity, the majority of MRI scans are only done for a small part of the spine. Seeing only a small part of the spine, makes labeling considerably more difficult, because approaches which could count down from C2 or up from S1 do not work.
- **Similarity between adjacent vertebrae:** Adjacent vertebrae often exhibit strong morphological similarities, making it difficult to distinguish between them. This similarity can lead to overlapping segmentations, where one vertebra is misclassified as its neighbor.
- **Variation between patients:** The same vertebra can present significant variations across different individuals. Factors such as age, genetics, height, and previous medical conditions can influence the shape, size, and orientation of the vertebrae, introducing challenges in creating a generalized labeling algorithm.
- **3-dimensional data:** Spinal medical images are inherently 3-dimensional, capturing the intricate structure of the spine in depth. This 3D nature demands algorithms that can handle volumetric data and take into account spatial relationships in three axes.

In the following sections, we will address these challenges, presenting robust solutions tailored for the unique characteristics of spinal images. In Section 5.1 an overview of the pipeline will be presented, including the three-step approach of the pipeline. In Sections 5.2, 5.3 and 5.4 the three pipeline steps will be introduced. Each step presents a task to be solved, each section details multiple approaches to solve that specific task. The combination of methods of steps 1 to 3 yielding the best results is using (1) slice-wise segmentation (Section 5.2.2), splitting along IVDs (Section 5.3.2), and multiclass segmentation (Section 5.4.4). The results of this combination are further elaborated in Section 6.1.

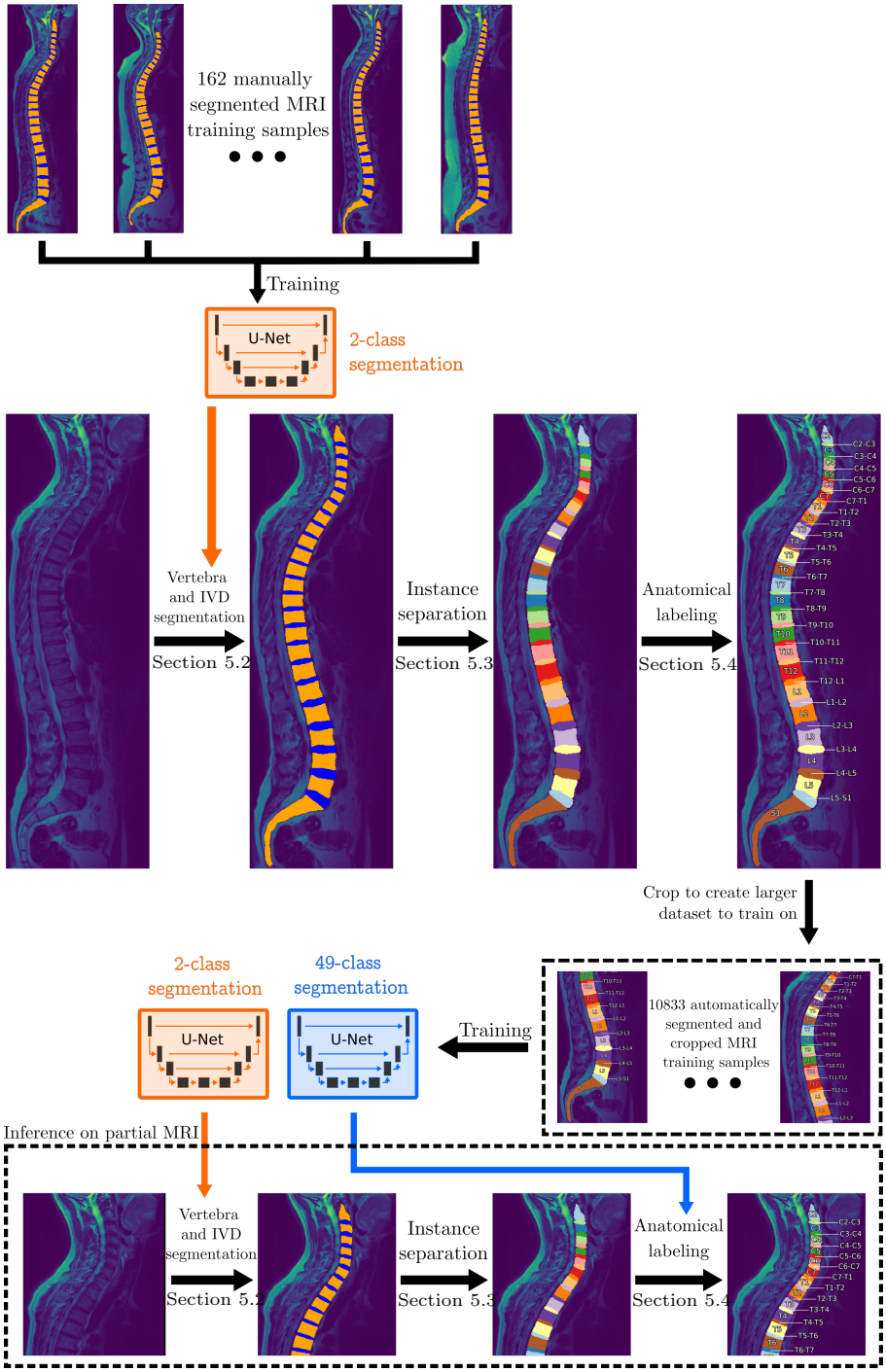


Figure 5.1.: **Proposed Three-step Anatomical Labeling Pipeline**. Initially, 162 samples annotated by experts are utilized to train a preliminary segmentation model. This model is then employed to segment an extensive dataset of 10,833 MRI scans. The segmented data is separated into vertebrae and IVDs, leveraging the fact that all instances can be inferred in a complete MRI scan by assuming the topmost vertebra is C2. By cropping images from the large dataset, a substantial training set is generated. This set is used to train an anatomical labeling segmentation model specifically for small FOV MRI images.

5.1. Overview of Methodology

The following methodology has been developed with the aim of accurately segmenting and labeling MR volume data, particularly focused on the segmentation of vertebrae and intervertebral discs (IVDs). The methodology for achieving the proposed goal of MR volume segmentation with correct labels involves three main steps:

1. **Segmentation of vertebrae and IVDs**: The first step aims to segment the vertebrae and the IVDs without considering their quantity or specific visibility.
2. **Instance separation**: This step involves creating an anatomically sorted list of instances (vertebrae or IVDs), where each instance represents a distinct set of voxels.
3. **Anatomical labeling**: The final step assigns a unique label to each instance from the previous step's list. Since the list is anatomically sorted, labeling a single instance allows for the inference of all other instances by 'counting down' or 'counting up'.

Creating a segmentation and instance separation is essential, because this allows processing and usage of the entire dataset of 10,833 patients for either further training of a better model or other approaches, such as statistics-based ones. Without these steps, the only dataset available would be the 162 patients with manually created segmentations. When segmenting the full image where all vertebra are visible, it is easy to "count-down" in order to determine the label for each instance. In Figure 5.1 the entire pipeline is shown, including the training process for both segmentation models.

The major difficulty of the entire pipeline lies in the labeling of small field of view (FOV) MR images, especially ones where neither the topmost (C2) or the bottommost (S1) vertebra is visible. In such a case the model cannot easily infer the class for the other instances, because then there is no easy anchor, such as C2 or S1, which are visually distinct from the other vertebrae.

Alternatively, a direct multiclass segmentation is possible, but has considerably worse results. Here, two possibilities arise: either training on the smaller dataset of 162 or using the full dataset of 10,833 with the automatically created ground truth data.

5.2. Step 1: Segmentation of Vertebrae and Intervertebral Discs

In this step the MR images are taken as input. A segmentation mask is created as the output for three classes: background, vertebrae and IVDs, with the same shape as the input image.

Only segmenting two classes, vertebra and IVDs, has the following advantages:

- This avoids the challenge of "similarity between adjacent vertebrae" and instead uses it as an advantage. Segmenting any vertebra as a single class is considerably easier and achieves good results even with a smaller dataset.
- The manually annotated ground truth data contain only these labels, this means it is not necessary to further modify or annotate the dataset.

The following two approaches were considered for this task: slice-wise segmentation in Section 5.2.2 and volume segmentation in Section 5.2.3. The one with the better results was the former: slice-wise segmentation. Because both approaches share the same necessary preprocessing and post-processing, these steps are elaborated in Section 5.2.1 and Section 5.2.4 respectively.

5.2.1. Preprocessing

To employ the segmentation network, a preprocessing step was undertaken. It's noteworthy that both slice-based segmentation and volume-based segmentation, outlined in Section 5.2.2 and Section 5.2.3 respectively, necessitated the same preprocessing measures.

Initially, each patient's image underwent a basic normalization where its minimum pixel value was mapped to 0 and the maximum to 1. Furthermore, in the ground-truth normalization process, every vertebra voxel, including S1, was designated an integer value of 1, while each IVD voxel was assigned a value of 2. Patients with only IVD segmentations were ignored. In future work, a method could be considered which uses the IVD segmentations to train only the segmentation part of the networks.

Subsequently, to achieve a consistent dimension of $18 \times 320 \times 896$ across all images, cropping or padding was performed. This size was selected to guarantee no exclusion of segmented spine portions in any ground truth (GT) data samples. For spines exceeding these dimensions, symmetric cropping was applied, while those falling short were symmetrically padded with 0.

Given that only 132 patients were part of the training dataset for this procedure, data augmentations were implemented to introduce more diversity during training. The augmentations were applied with certain probability p, sampled uniformly from the given ranges. These augmentations included geometric transformations such as rotations ($p = 0.5$) within $\theta \in [-20°, 20°]$, translations ($p = 0.5$) of up to 10% in any direction, scaling ($p = 0.5$) between 80% and 120% and horizontal flipping ($p = 0.5$). Furthermore, photometric augmentations, which change the pixel intensities, were applied: gaussian noise ($p = 0.25$) with $\mu = 0, \sigma^2 = 0.01$, gaussian blur ($p = 0.25$) with $\sigma \in [0.1, 2.0]$ and kernel size $k = (3, 3)$, contrast and brightness ($p = 0.25$) with $intensity \in [0.9, 1.1]$.

In conclusion, normalization was applied during all subsequent steps which use the MRI scans as inputs, whereas the augmentations were only applied during training.

5.2.2. Slice-based Segmentation

Our model employs a slice-wise approach for image segmentation. The process commences with preprocessing, as detailed in Section 5.2.1. Following segmentation, all slices are assembled to yield the complete 3D segmentation. A subsequent post-processing step, elaborated in Section 5.2.4, rectifies segmentation errors.

We adopt a sagittal slice-by-slice segmentation due to the prohibitive size of the entire MRI scan for most 3D models. An alternative method of whole volume segmentation is discussed in Section 5.2.3. To enhance the model and accommodate implementation specifics, each slice is concatenated with its two adjacent slices as channels, resulting in an input shape of $3 \times 320 \times 896$. Any missing slices are padded with the middle slice.

We evaluated numerous segmentation architectures: U-Net [25], FPN [44], MA-Net [45], DeepLabV3/DeepLabV3+ [46], Linknet [47], and PAN [48]. U-Net is further elaborated in Section 2.4.4, while the other architectures are detailed in their respective papers. 56 encoders were assessed in various combinations with these segmentation architectures.

The model was trained on a set of 132 patients, each annotated by an expert and comprising 18 sagittal slices. It was evaluated on a validation set of 30 patients. For training, we evaluated Jaccard loss, Dice loss, and cross-entropy loss, with Jaccard loss yielding the best results for slice-wise segmentation. These losses are described in detail in Section 2.3. We used the Adam optimizer [49] with a learning rate of 0.001. Training lasted between 5 to 30 epochs, employing an early stopping strategy if the validation DSC did not improve over 5 epochs. Each epoch evaluated each patient and each slice, resulting in a total of $132 \cdot 18 = 2376$ samples per epoch. Training was conducted on a NVIDIA RTX 3090 GPU and took between 20 minutes to 2 hours per model.

The most effective model for this segmentation task was a U-Net [25] with a ResNet152 [25] as the encoder. Further results are shown in Section 6.2, and a complete list of all experiments can be found in Appendix A.

5.2.3. Volume-based Segmentation

As an alternative to the slice-wise approach detailed in Section 5.2.2. Unlike slice-based segmentation, where each 2D slice is processed independently, volume-based segmentation handles the entire 3D image volume at once. This method allows for the exploitation of 3D spatial context, which is particularly beneficial for understanding complex anatomical structures not available during slice-wise segmentation.

In volume-based segmentation, the entire 3D volume is input directly into the model. This requires a more complex network architecture capable of processing 3D data. For

this task, we employed a 3D U-Net [50, 26], an extension of the U-Net architecture that operates on volumetric data.

Training a model on 3D volumes requires a significant amount of memory and computational power. We utilized a NVIDIA RTX 3090 GPU to manage this task. The model was trained using the dataset comprising full 3D scans from the same 132 patients mentioned in Section 4.

The loss functions used for training were similar to those in slice-based segmentation, including Dice loss, Jaccard loss and cross-entropy loss. The Adam optimizer [49] was used, with a learning rate of 0.001.

Volume-based segmentation offers several advantages over the slice-based approach. Most notably, it provides a more holistic understanding of the 3D structures within the image. This is particularly beneficial for complex and interconnected anatomical regions where 2D slices might not reveal the full context of the structures involved.

Additionally, this approach can help in identifying and understanding anatomical variations and pathologies that are only apparent when observing the full volume. For instance, abnormalities in the shape or connectivity of bones that might be missed in 2D slices can be more easily detected in the 3D volume.

Despite its advantages, volume-based segmentation has some limitations. The primary challenge is the high demand for computational resources and memory, which makes it difficult to deploy on standard clinical systems. Furthermore, the large size of 3D volumes can lead to longer processing times, impacting the workflow in time-sensitive clinical environments.

Volume-based segmentation presents a powerful alternative to slice-based segmentation, especially for complex anatomical analyses. Although it requires more computational resources, its ability to leverage the 3D context of medical images makes it an invaluable tool in the field of medical image analysis. However, in our case it did not deliver better results than slice-wise segmentation due to the limitations in model size when having to train on the entire volume. The comparison between both models can be seen in Table 6.3.

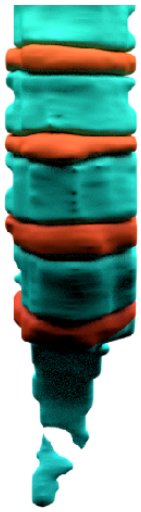
5.2.4. Post-processing

The segmentation is not perfect and especially “blobs” which are disconnected from the main spine cause issues in the later stages for labeling. Three examples can be seen in Figure 5.2, not filtering those would lead to issues in the subsequent pipeline steps, because there it is assumed that all segmented parts belong to the spine.

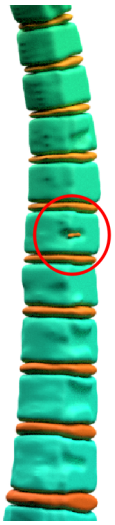
Two post processing steps are performed in order to improve the performance of the next steps:

- **Deletion of entirely disconnected blobs:** this ensures that only a single contiguous object (the spine) is returned after this step. In order to achieve this, each non-zero voxel is temporarily labeled as 1, background voxels remain as 0. Then the connected components algorithm is used, which has been introduced in Section 2.2, to find each connected component. Now that IVDs are labeled as 1, the spine will be found as a single connected component. The connected component with the largest volume is kept, all others are discarded. Finally, the previous labels are returned to the positions of the largest connected component.
- **Heuristic deletion of small class-wise disconnected blobs:** this ensures that a small number of incorrectly classified voxels do not count as an entire vertebra or IVD. Again, connected components is used in order to find all objects, however, this time the labels are kept as is. All connected components below a volume of 50 voxels are removed (set as background). In our train dataset of 132 patients we have found that even the smallest IVDs contained at least 131 voxels (C2-C3). However even this seems to be an outlier, as the second smallest C2-C3 IVD contained 249 voxels, and on average C2-C3 contained 586 voxels. Therefore 50 voxels seems to be safe number without risk of removing relevant instances.

The two post-processing steps are applied sequentially. The first step removes cases such as Figure 5.2a and Figure 5.2c, whereas the second step removes cases such as Figure 5.2b.



(a) A disconnected "blob" below S1



(b) A few incorrectly classified IVD voxels

(c) Due to one slice having worse quality, some vertebrae are incomplete

Figure 5.2.: **Segmentation Errors**. Three examples where post-processing is necessary in order to achieve a correct labeling.

5.3. Step 2: Instance Separation

In the context of our research, after obtaining the segmented 3D medical images, a crucial subsequent step is to differentiate and isolate each distinct anatomical structure within these segmentations.

Splitting the segmentation into separate instances has several advantages:

- Using this instance separation, it is possible to segment and split the entire large dataset of over 10000 patients, allowing for further processing.
- This allows for many other approaches to be viable in the next step, as only a single label needs to be predicted for each instance.
- Because the manually truthed dataset does not differentiate which vertebra is visible, using the instance separation allows training of a multiclass segmentation models on the small manually annotated dataset.

This section elaborates two techniques employed for this purpose: the connected components algorithm in Section 5.3.1 and splitting vertebrae using IVDs in Section 5.3.2. The latter was the one used in the final pipeline. However, the latter also uses the connected components for some parts, therefore they share similarities.

5.3.1. Via Connected Components

As previously detailed in Section 2.2, the connected components algorithm is a foundational method in image processing and computer vision. It identifies and labels contiguous regions within an image based on specific pixel value criteria. For our task, this algorithm plays an integral role in separating individual IVD and vertebrae within the segmented images, with its operations conducted within a 26-neighborhood in 3D to ensure comprehensive connectivity analysis.

1. **Using segmentations**: The algorithm makes use of the segmentations obtained from the previous task. In the segmentation mask for vertebrae and IVDs, each distinct structure in the segmented image is recognized based on its spatial connectivity in the 3D volume using the connected components algorithm.
2. **Sorting**: Post-segmentation, the centroids of all 3D structures are computed. Utilizing these centroids, the segmented structures, encompassing both discs and vertebrae, are systematically sorted. This sorting is purely based on the transversal coordinate of the centroid (top-down axis).
3. **Labeling**: Upon sorting of the distinct structures, each vertebra is labeled with an odd number starting at the top with 1, ensuring that every vertebra receives a unique, odd identifier. Conversely, each IVD is labeled with an even number, starting with 2, to distinguish them from the vertebrae. The distinction between vertebra and IVD can be made because of the segmentation, as it already distinguishes between these two classes.

The primary merit of the connected components algorithm is its unambiguousness and operational efficiency. It offers an uncomplicated methodology to segment adjoining structures devoid of intricate computations. In the context of our study, it proves invaluable, especially for differentiating closely located vertebrae.

However, it's imperative to note certain susceptibilities of the algorithm. Particularly, in the 26-neighborhood in 3D, there's a risk of erroneous segmentation when two vertebrae are in close proximity or touch each other. Such instances can lead to both vertebrae being recognized as a singular connected component, complicating subsequent analyses, examples for such cases can be seen in Figure 5.3. Failure in separation results in failure of separation of all subsequent vertebrae.

Conclusively, the connected components algorithm, coupled with centroid-based sorting, offers a technique for anatomical spinal structure separation and organization within our research framework. However, it is prone to errors where vertebrae are close to each other, producing incorrect results.

(a) Two vertebrae share many voxels

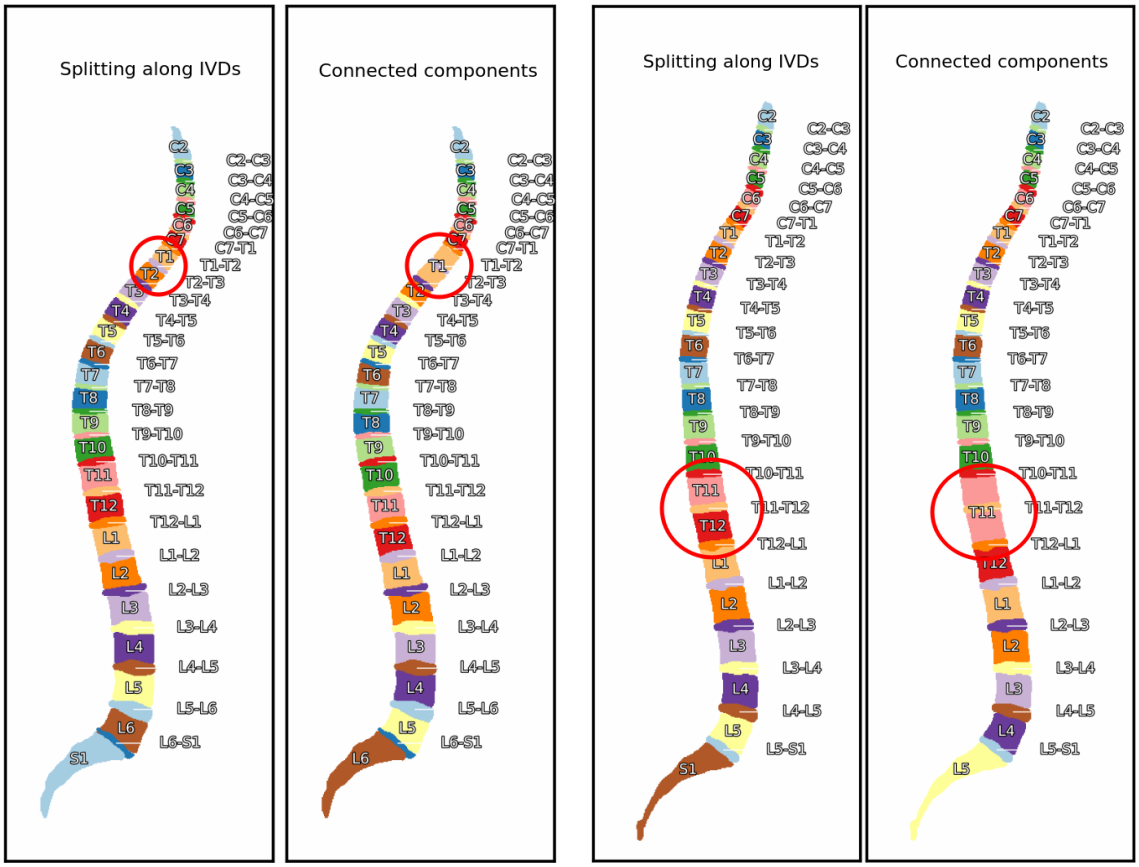
(b) Two vertebrae share one or two voxels

(c) The IVD was only partially segmented, resulting in the vertebrae being connected

Figure 5.3.: **Separation Errors**. Three examples where vertebrae have common voxels, causing a naive connected components approach to fail to separate these vertebra

5.3.2. Split Along Intervertebral Discs

Addressing the challenges posed by vertebrae situated in close proximity to one another is crucial for accurate segmentation. The connected components algorithm, while



(a) Connected components fails to separate **T1** from **T2**

(b) Connected components fails to separate **T11** from **T12**

Figure 5.4.: **Instance Separation Method Comparison.** Two examples where the *connected components* approach (Section 5.3.1) fails to separate two adjacent vertebra, but *split along IVDs* (Section 5.3.2) succeeds.

effective, can occasionally falter when vertebrae are too close. A promising solution to this issue is segmenting by splitting along the IVDs. This method not only harnesses the natural anatomical boundary that IVDs provide between vertebrae but also offers a potential remedy to the close proximity problem.

The algorithm consists of three steps:

1. **Segmentation of IVDs using connected components**: The initial step isolates each IVD using the connected components algorithm. This segmentation ensures that each IVD is distinctly treated in subsequent processes.
2. **Principal component analysis (PCA) on IVD coordinates**: Once each IVD is segmented, PCA is executed on its voxel coordinates. This analysis yields a plane that accurately represents the IVD's orientation and shape within the 3D space.
3. **Splitting vertebrae using the IVD plane**: With the plane obtained from PCA for every IVD, the segmentation of vertebrae is methodically executed.

Starting with a label of 1 for every vertebra voxel, the algorithm iterates over each IVD from top to bottom. For every voxel of the vertebra positioned below an IVD's plane, its label receives an increment of 2. This technique ensures each vertebra acquires a unique label, successfully segmenting them along the IVDs planes.

The *split along intervertebral discs* strategy presents numerous advantages:

- *Natural boundary utilization*: This method often yields anatomically precise segmentations by capitalizing on IVDs as inherent boundaries.
- *Mitigating close proximity challenges*: In contrast to the connected components method, where two neighboring vertebrae might be inaccurately segmented as a single entity, this technique effectively addresses such issues due to its reliance on the planes of IVDs.

Two examples where this approach addresses the shortcomings of the *connected components* approach can be seen in Figure 5.3. Nevertheless, potential challenges merit consideration. The efficacy of this approach is contingent upon the precise segmentation of IVDs. Misoriented or inaccurate IVD segmentation can lead to the derivation of incorrect planes from PCA, culminating in less than optimal vertebrae segmentation.

Conclusively, the methodology presented in this section offers a robust alternative for vertebrae separation, which is not as susceptible to classify adjacent vertebrae as the same class.

5.4. Step 3: Anatomical Labeling

In this task each voxel is assigned its anatomically correct label, such as **C2** or **T12**. The instance separation, as described in Section 5.3, is used as input.

This task presents the toughest challenge, therefore many approaches are presented here. The best performing approach was multiclass slice-wise segmentation shown in Section 5.4.4.

The many approaches can be grouped into two groups: image-based and statistics-based approaches. The image-based approaches use the underlying MRI image together with the segmentation as created in step 1 to label the image. Statistics-based approaches use some extracted information about the vertebrae and IVDs, such as volume, direction, or height, in order to label the image.

The two image-based approaches include multiclass slice-wise segmentation (Section 5.4.4) and local encoder with GNN classifier (Section 5.4.3). The two statistics-based approaches include a direction-based classification (Section 5.4.1) and classification with classical machine learning (Section 5.4.2).

5.4.1. Directional Vector Matching

This section describes a statistical approach using the direction of the spine to figure out which vertebrae and IVDs are currently visible. For this the huge dataset is used to create statistics about the direction of the spine, the currently to-be analyzed spine is then compared at each possible location with the available directions. The best match is where the directions deviate the least, this is then the prediction.

Determining the Direction of the Spine

In this approach a direction for each vertebra and disc is necessary, multiple ways of creating the direction have been considered. Two methods are presented here, Method **(2)** was used in the final version and produced the best results. **(1)** using PCA to find the principal components of the discs and **(2)** using the difference between object centroids as the direction vector.

- In **(1)**, the assumption is made that the IVDs are considerably more flat in one direction. Applying PCA to the coordinates of the IVD, yields three principal component vectors. With the first assumption, the first two vectors will be along the axes with the most variance, i.e. where the disc is the widest. Because the third principal component vector is also orthogonal to the first two vectors, it can only point "up" or "down" along the thinnest part of the disc. Then normalization is applied and ensured that each vector points in the same "up" direction, roughly towards C2. The direction vectors for the vertebrae are calculated as the average of the two adjacent IVD direction vectors.

  This method works well, however it has two problems. Firstly, it is susceptible to outliers. A couple voxels in the wrong position can greatly change the first two principal component vectors. Secondly, when creating the directions this way, the normals do not point in the direction of the spine, especially the IVDs between C2 and C6. The issue with this, is that it becomes considerably more difficult to differentiate between the directions between the upper spine and middle spine, as in this approach these essentially point to the same direction.
- In **(2)**, the centroid of each distinct object is computed by taking the average of all voxels constituting that object. To ascertain the orientation of the spine at a particular object's location, be it a vertebra or disc, we derive a vector by subtracting the centroids of its immediate neighbors. This vector, which points from the lower object to the upper one, is subsequently normalized and adopted as the spine's direction at the midpoint.

  Nevertheless, certain objects, notably C2 and S1, lack either an upper or lower adjacent counterpart. To circumvent this issue, two hypothetical points are introduced to augment the spinal curvature on both extremities. The generation

of these points entails devising three predictive functions, each tasked with forecasting the subsequent X, Y, and Z coordinates, respectively. These functions are based on a quadratic form given by $f(x) = ax^2 + b$. The functions are tailored to fit the n closest data points to the target point under prediction. Given an assumption of roughly equidistant neighboring points, these data points are associated with the x-coordinates $\{0, 1, 2, 3, ...\}$. The predicted coordinate for the target point is then acquired by evaluating the function at $x = -1$, and this process is replicated for all three coordinates of the point under prediction. In our case $n = 4$ was used, i.e. the four uppermost objects as well as the four lowermost objects were used in order to predict an additional coordinate. With these additional points the algorithm can be used as described above.

Using the separated objects as described in Section 5.3 a derived dataset of directions is created using method **(2)** for each vertebra and IVD direction for each spine of the 10,833 patients. The dataset D can be formally defined as follows: $D \in \mathbb{R}^{10833 \times 49 \times 3}$. For each entry $D_i \in D$, it represents a list of 49 vectors: $D_i = \{\vec{v}_{i,1}, \vec{v}_{i,2}, ..., \vec{v}_{i,49}\}$.

Finding the Best Matching Directions in the Dataset

Using the direction dataset as described in the previous section, this approach labels an unknown new small FOV MRI image in the following way: firstly, the first two pipeline steps are performed in order to obtain separate segmented and sorted instances for each vertebra and IVD. As the MRI image is partially visible, not all instances are available, necessitating an additional labeling step. In this case only a single label for the topmost instance is produced, all other labels are inferred by counting down from the inferred label. The labels **C2**, **C2-C3**, ..., **S1** are mapped to the numbers 1 through 49. Therefore the problem can be defined as a function which maps a list of directions for each vertebra and IVD of the current candidate to the label of the topmost structure:

$$f(\vec{d_1}, \vec{d_2}, ..., \vec{d_n}) \rightarrow \{1, 2, ..., 50 - n\} \tag{5.1}$$

Where $n \in \{1, 2, ..., 49\}$ is the number of found instances by the instance separation method and $d_i \in \mathbb{R}^3$ is the direction of the i-th instance. It is of note that the output space decreases when more directions are provided, as the lower instances need to have a valid label which these can be assigned to. For example, for $n = 49$, i.e. the entire spine is visible, only a single output value is possible: 1, which maps to **C2**.

For distance calculation, we use the Euclidean distance between two vectors $\vec{a}$ and $\vec{b}$, given by $\|\vec{a} - \vec{b}\|_2$. To calculate the distance between two lists of vectors of the same length m, $A = \{\vec{a}_1, \vec{a}_2, ..., \vec{a}_m\}$ and $B = \{\vec{b}_1, \vec{b}_2, ..., \vec{b}_m\}$, we sum the distances between the corresponding vectors: $\sum_{j=1}^{m} \|\vec{a}_j - \vec{b}_j\|_2$. For each dataset entry D_i and for every

possible starting position p, we compare the list L with a sublist of D_i of length n. The sublist for entry D_i and position p is defined as $\{\vec{v}_{i,p}, \vec{v}_{i,p+1}, ..., \vec{v}_{i,p+n-1}\}$.

The function $f(\vec{d}_1, \vec{d}_2, ..., \vec{d}_n)$ identifies the index of the dataset entry and the starting position where the distance between L and the sublist is minimized. Formally:

$$f(\vec{d}_1, \vec{d}_2, ..., \vec{d}_n) = \arg\min_{i,p} \sum_{j=1}^{n} \|\vec{d}_j - \vec{v}_{i,p+j-1}\|_2$$

Subject to the conditions:

$$1 \leq i \leq 10000$$

$$1 \leq p \leq 50 - n$$

These conditions ensure the sublist starting at p and containing n vectors remains within the bounds of D_i.

Using this function f a prediction can be created for a small FOV MRI image.

5.4.2. Conventional Machine Learning Classification

In the task of classifying each separated anatomical structure, the challenge intensifies when not all objects are visible in the acquired imaging. Given the variability and possible occlusions within the medical images, adopting machine learning strategies that can make informed predictions even with missing data is pivotal. Conventional machine learning algorithms, with their ability to operate on hand-crafted features, often fit well in such scenarios.

Additional Statistics about each Vertebra and IVD

Before employing any classification strategy, it's essential to understand the inherent characteristics of the structures to be classified. By examining each separated vertebra and IVD, various statistics can be calculated to serve as features for the subsequent classification tasks. The metrics considered include:

- **Height**: The vertical dimension of the structure, as measured in the axis of the spine.
- **Width**: The horizontal dimension, orthogonal to the height, at the widest position.
- **Volume**: The number of voxels occupied by the object as returned by the segmentation.
- **Direction**: The orientation of the vertebra or IVD as a 3D vector, which can be particularly informative given the anatomy's natural curvature and alignment.

These statistics are created for each separated instance of vertebra and IVD for the large dataset of 10,833 patients.

Applying the methods

Once the features are extracted, the classification task primarily focuses on predicting the topmost vertebra present in the image. The subsequent vertebrae can be inferred based on the prediction of the topmost vertebra. With the curated features serving as inputs, two prominent machine learning algorithms are employed: random forest [51] and XGBoost [52].

- **Random forest**: This ensemble learning method constructs multiple decision trees during training and outputs the mode of the classes (classification) of the individual trees for prediction. It's particularly adept at handling large datasets with higher dimensionality, making it well-suited for this task.
- **XGBoost**: Standing for eXtreme Gradient Boosting, XGBoost operates by building an ensemble of prediction models, typically decision trees. It's known for its efficiency and capability to tackle unbalanced datasets. In the context of vertebrae classification, XGBoost can leverage its gradient boosting mechanism to iteratively refine its predictions, based on the features of the vertebrae and IVDs.
- **Multilayer perceptron (MLP)**: An MLP is a class of feedforward artificial neural network that consists of at least three layers of nodes: an input layer, at least one hidden layer, and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLPs utilize a supervised learning technique called backpropagation for training the network. See Section 2.4.1 for a more in-depth explanation.

All three algorithms undergo training using the statistics dataset of 10,833 as created in the previous Section 5.4.2, with train and validation split as described in Section 4. The algorithms were trained on three different context lengths: 5, 10 and 15 visible vertebrae. These result in 10, 20 and 30 visible objects. In total there are 49 possible objects. During training for each patient a random continuous subset of size 10, 20 or 30 is sampled from the statistics dataset. Both random forest and XGBoost are trained with $n_{estimators} = 1000$ as regression tasks. I.e. both methods predict only a single value: the label of the topmost vertebra or IVD. The lower instances are inferred using the topmost one. This approach, however, lacks the ability to differentiate between lumbarization and sacralization, where the order of the vertebrae changes. In future work multivariate random forest and XGBoost should be evaluated.

In this work, we do also use a multilayer perceptron in a classification task, where not only the topmost vertebra is predicted, but also every single instance visible in the image. This is achieved by predicting 49 values, the probability for each instance **C2**, **C2-C3**, ..., **S1** being visible in the current slice. The MLP is trained with four hidden layers with (128, 256, 512, 1024) neurons, respectively. ReLU is used as activation function with cross entropy loss. The results of these can be seen in Section 6.1.

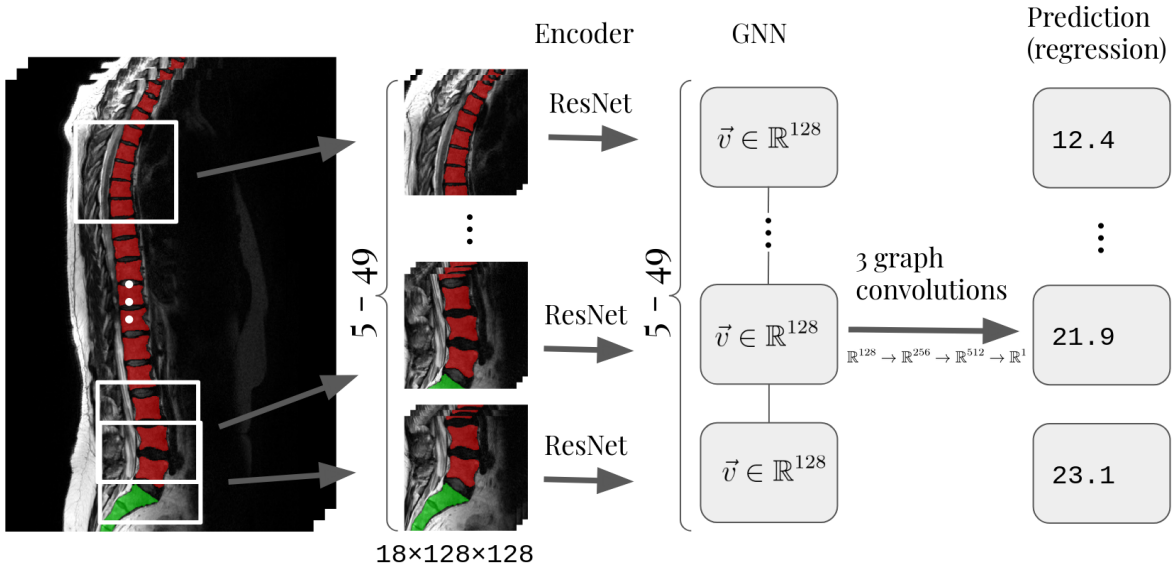


Figure 5.5.: **GNN Approach**. Overview of the local encoding with graph neural networks approach. The MRI is segmented and the vertebrae are separated using the methods proposed in step 1 and 2, then a 3D patch of shape $18 \times 128 \times 128$ is extracted around each vertebra. The number of vertebrae may vary in the input, depending on the FOV size. The 3D patches are used as input for an encoder (e.g. ResNet), the output of which is an encoding vector $\vec{v} \in \mathbb{R}^{128}$ representing that vertebra and its neighborhood. Each vector $\vec{v}$ is used as the features of a GNN node. Adjacent GNN nodes are connected if they represent adjacent vertebrae. Three graph convolutions are performed, resulting in vectors $\mathbb{R}^{256}$, $\mathbb{R}^{512}$ and $\mathbb{R}^{1}$, respectively. The final output is a single number, representing a regression task, where each node predicts the index of that vertebra as a number between 1 and 26. Finally, the actual prediction is the anatomically possible prediction which has the least mean square error (MSE) when compare to the outputs of the GNN.

5.4.3. Local Encoding with Graph Neural Networks

Classical machine learning techniques, while effective, operate on handcrafted features that often require meticulous preprocessing and engineering. In contrast, deep learning architectures like GNNs enable an end-to-end learning mechanism that inherently understands structural relationships in the data. In the context of vertebrae and IVD classification, leveraging such relationships becomes especially pivotal. The proposed approach integrates local encoding with GNNs to achieve this goal.

The initial step involves encoding the anatomical structures present in the input image. The neighborhood for each segmented structure (e.g., a vertebra or an IVD) is passed through a ResNet-based encoder. This encoder transforms the spatial information from the 3D imaging data (of dimension $18 \times 128 \times 128$) into a high-dimensional feature vector

$\mathbf{v} \in \mathbb{R}^{128}$. As illustrated in Figure 5.5, this encoding process happens locally for each structure, ensuring that unique and specific features are captured for each.

Once the local encoding vectors are obtained, they are treated as nodes in a graph. These nodes are interconnected, signifying the spatial relationships between the anatomical structures. GNNs are particularly adept at managing such relational data. They have been described in Section 2.4.3.

The core operation involves graph convolutions. In this architecture, three graph convolution operations refine and update the feature vectors. This iterative process ensures that each node's representation (i.e., each vertebra or IVD) is influenced by its neighboring nodes, effectively capturing the spatial relationships and dependencies among them.

5.4.4. Multiclass Segmentation

In this section, we detail a methodology for anatomically labeling each vertebra and IVD using a segmentation model, building upon the approach in Section 5.2.2. The adaptation to a multiclass setting introduced the need for a significant increase in training data due to the introduction of additional classes. The foundational model remains unchanged from the one shown in Section 5.2, but with an expansion of classes from 3 to 50. Class 0 is retained for background representation. The subsequent 49 classes are designated for vertebrae and IVDs, sequenced in alternating order starting with $\{1 :=$ **C2**$, 2 :=$ **C2-C3**$, ..., 47 :=$ **L6**$, 48 :=$ **L6-S1**$, 49 :=$ **S1**$\}$. Given the phenomena of sacralization and lumbarization, as explained in Section 2.1, a majority of spines are observed to lack **L6** and **L5-L6**, with a smaller subset also missing **L5** and **L4-L5**. This absence in some lumbar region images complicates the NN's learning process regarding sequential class progression. Nevertheless, the consistent classification of **S1** as class 49 offers advantages that compensate for the aforementioned challenges.

Considering the model's complexity and the augmented class count, relying solely on the GT training data, as characterized in Section 4, proved inadequate. The model's learning proficiency was compromised in the absence of supplementary data, necessitating additional measures. To address this, the complete dataset of 10,833 was used to train and evaluate this model.

Post-processing

The output of this model is not used directly as the final output, instead it is treated as a probability map and is further post-processed, to obtain a final anatomically possible label assignment (see columns "Output" and "Post-processed" in Figure 5.6). This is done by performing step 1 and 2 first, thereby obtaining the separated instances from the given small FOV MRI image. With this knowledge, all possible

instance label assignments with the same length are created as the number of separated instances. For example, if there are five instances, the first being a vertebra, then (**C2**, **C2-C3**, **C3**, **C3-C4**, **C4**) is a possible label assignment, and so is (**C3**, **C3-C4**, **C4**, **C4-C5**, **C5**), and so on. Care has to be taken for the special case of **S1**, as it can appear after **L4**, **L5** or **L6**, therefore a few more possible label assignments arise for labels in the lumbar region. For example, in the case of 5 instances both (**L3**, **L3-L4**, **L4**, **L4-L5**, **L5**) and (**L3**, **L3-L4**, **L4**, **L4-S1**, **S1**) are possible label assignments. In Figure 5.6 three examples can be seen for multiclass segmentation

For each possible label assignment, an error is calculated, representing how far off the label assignment is from the multiclass segmentation output. The label assignment with the lowest error is then used as final label assignment. In the following we will detail how this label assignment is calculated. Due to the numerical nature and the sorting of the labels, adjacent labels are only separated by a value of one, therefore in the following instead of the readable labels, their numerical equivalents will be used, e.g.: instead of (**L3**, **L3-L4**, **L4**, **L4-L5**, **L5**), $(3, 4, 5, 6, 7)$ is used.

The error for a single instance label is calculated as follows:

$$\text{Error}_{\text{instance}}(a, L) = \frac{\sum_{i=0}^{||L||}(a - L_i)^2}{||L||} \tag{5.2}$$

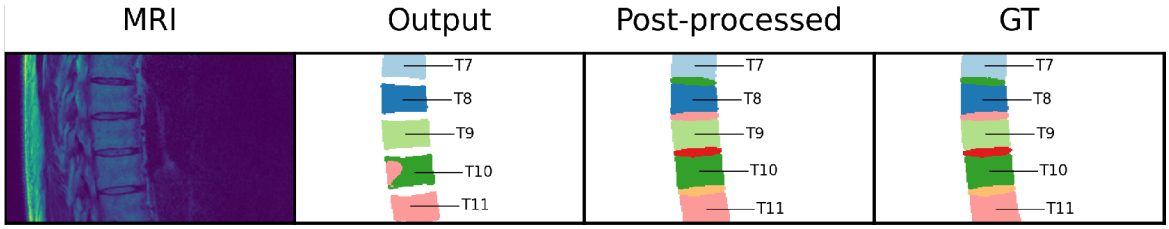
Where $a \in \{1, 2, ..., 49\}$ is the assigned label for the current instance. $L \in \{0, 1, 2, ..., 49\}^m$ is a list of all voxel labels as numeric values for the current instance, which have been segmented by the multiclass segmentation. L is calculated by iterating over each voxel in the instance, as separated by the algorithm in step 2, and for each voxel retrieving the segmented label.

Thus, for a given label assignment $A \in \{1, 2, ..., 49\}^n$, with $n \in \{1, 2, ..., 49\}$ and the constraints above, an error for an entire label assignment can be computed:

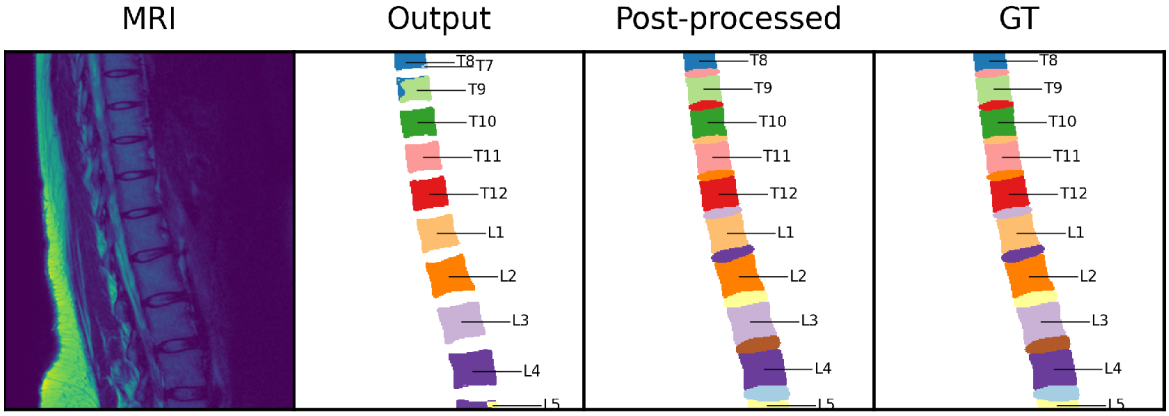
$$\text{Error}_{\text{total}}(A, \mathbb{L}) = \sum_{i=0}^{||A||} \text{Error}_{\text{instance}}(A_i, \mathbb{L}_i) \tag{5.3}$$

With $\mathbb{L} \in \{0, 1, 2, ..., 49\}^{m \times n}$ being a list of voxel lists, where each list in $\mathbb{L}$ is a list like L above.

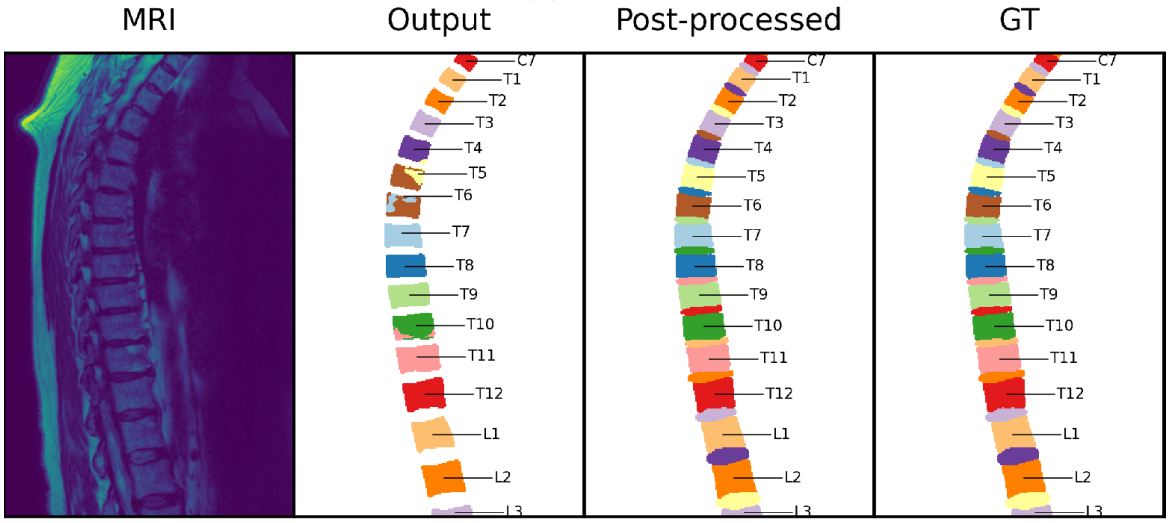
Then $\text{Error}_{\text{total}}$ is computed for all label assignments, and the one with the lowest error is picked. Finally, the output of the multiclass segmentation can at this point be discarded, as only the best label assignment matters. For each separated instance, the corresponding label is assigned to each voxel belonging to that separated instance. This is then the final output of this method.



(a) V=5



(b) V=10



(c) V=15

Figure 5.6.: **Multiclass Segmentation Output.** The midsagittal slice of three different FOV sizes. The column *MRI* shows in the cropped input image. *Output* shows the direct output of the multiclass segmentation model. *Post-processed* shows the output of the optimal label assignment, merged with the IVD segmentation (described in Section 5.4.4), *GT* shows the ground-truth.

6. Results

In this section we show various experiments done in order to evaluate our work. Section 6.1 shows the main contribution of the thesis: anatomical labeling of small field of view MRI images. This encompasses the entire pipeline: (1) segmentation of vertebrae and IVDs, (2) instance separation and (3) anatomical labeling. Due to the amount of possible combinations of approaches for the entire pipeline, not every combination is shown there. Instead, for step (1) and (2) the best methods are taken, and only step (3) is exchanged (where applicable). In order to find the best methods for step (1), in Section 6.2 a comparison between the two segmentation approaches, slice-wise and volume segmentation, is made. Analogously, for step (2), in Section 6.3 the results of the two algorithms for separating instances using connected components and by splitting along the IVDs is presented. Finally, in Section 6.4 the statistics about lumbarization and sacralization, which have been found using the anatomical labeling pipeline, are shown.

6.1. Anatomical Labeling Pipeline Results

This section presents the results of the anatomical labeling pipeline for small field of view (FOV) MRI images, as detailed in this thesis. The effectiveness of the pipeline on small FOV MRI images is evaluated by focusing on three FOV sizes, expressed as the number of visible vertebrae: $V = 5$, $V = 10$, and $V = 15$. This implies that either 5, 10, or 15 out of a possible 24-26 vertebrae are visible in the given MRI scan. These sizes correspond approximately to 20%, 40%, and 60% of the spine, respectively. However, due to the variation in vertebra sizes, this may vary (see Figure 5.6 for examples of each size).

6.1.1. Experiment Setup

To obtain consistent FOVs, the ground truth is used to determine the range of the image to be used. For a given MR image and corresponding ground truth, $n = 26 - V$ small FOV subset images are sampled.

For each of these subset images, the same width and depth are maintained, but a different height is used. This means that for the sample with index $i \in \{1, 2, ..., n\}$, all values

$\in \{2i-1, 2i, ..., 2i+2V-3\}$ that are in the original ground truth must also be in the subset ground truth. For example, with an FOV size of $V = 3$ and subset sample index $i = 2$, all ground truth values between 3 and 7, both inclusive, must also be in the subset ground truth sample, which corresponds to the labels {C3, C3-C4, C4, C4-C5, C5}.

Finally, the minimal height for the subset is used such that the above condition holds. It is of note that due to the variance in the sizes of the vertebra this height varies greatly depending on where in the spine it was sampled. For example, for $V = 5$, subset images in the lumbar region might have a height of ≈ 260 voxels, whereas in the cervical region it might be ≈ 120 voxels.

6.1.2. Anatomical Labeling Method Comparison

In this section the subset accuracy metric (Section 2.3.1) is used, which refers to an accuracy where every single class prediction was correct. I.e. subset accuracy is 0%, unless the accuracy per class is 100% for a given sample. For the case of spine anatomical labeling it means, that each instance (a vertebra or IVD) was assigned the correct label. The label of an instance is determined by sampling the voxels from the ground truth in the prediction, if the majority of sampled voxels are of the same label, then that is the label for that instance.

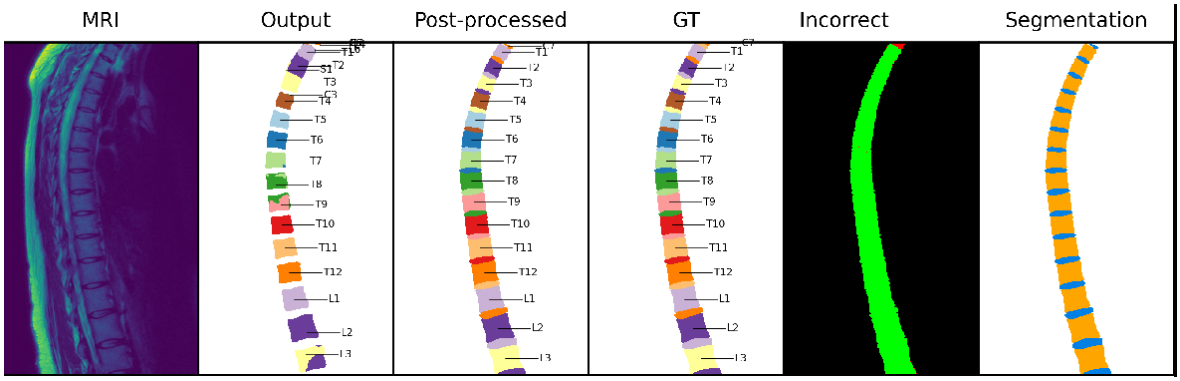


Figure 6.1.: **An Example Considered Incorrect Without Discarding Incomplete Vertebra or IVDs.** The column *MRI* shows the cropped input image. *Output* shows the direct output of the multiclass segmentation model. *Post-processed* shows the output of the optimal label assignment, merged with the IVD segmentation (described in Section 5.4.4), *GT* shows the ground-truth. The image labeled *Incorrect* shows the voxels which are correct in green, and which are incorrect in red, and background in black. (*GT* is compared with *Post-processed*). *Segmentation* shows the output of the segmentation as created in Section 5.2. Notice the red voxels at the top of the *Incorrect* column, a single barely visible IVD was misclassified, resulting in a subset accuracy of 0 and the *Incorrect* label for this sample.

In many cases vertebrae or IVDs close to the edge of the image are barely visible, and therefore are segmented or labeled incorrectly. This is especially difficult when evaluating subset accuracy, which requires every single vertebra and IVD instance to be labeled correctly (further described in Section 2.3.1). As in that case, a segmentation such as Figure 6.1 is considered entirely incorrect. In order to avoid this, 15 voxels at the top and bottom in the transversal (top-down) axis are disregarded when assessing subset accuracy (roughly half of a vertebra).

Section	Methods	Subset Accuracy for FOV		
		@ V=5	@ V=10	@ V=15
5.4.1	Direction-based	66.5%	71.3%	75.9%
5.4.2	Random Forest Regression	77.3%	81.4%	88.3%
5.4.2	XGBoost Regression	79.5%	87.1%	**94.4%**
5.4.2	MLP Classification	83.9%	85.8%	93.6%
5.4.3	Encoder+GNN	11.8%	4.1%	6.4%
5.4.4	Multiclass Segmentation	**85.5%**	**92.6%**	**94.4%**
5.4.4	*T132* Multiclass Segmentation†	80.0%	81.6%	92.9%

Table 6.1.: **Subset Accuracy for Anatomical Labeling Pipeline.** Anatomical labeling evaluation for the entire pipeline on the validation set (542 patients) based on the methods proposed in step 3 Section 5.4. As the pipeline consists of three steps, only step 3 (anatomical labeling) is changed in this table, for step 1 (Section 5.2: segmentation of vertebrae and IVDs) and step 2 (Section 5.3: instance separation) the best methods are used with the corresponding step 3 method. The results for the best methods for step 1 and 2 are shown in Section 6.2 and Section 6.3, respectively. The row marked with † is ablation study, showcasing the performance loss of not using the pipeline. *T132* means that a model was only trained on the smaller dataset of 132 MRI scans, showcasing what a lack of anatomical labeling pipeline for additional data creation entails.

In Table 6.1 a comparison is made between the various methods introduced in Section 5.4 using subset accuracy. Multiclass segmentation (Section 5.4.4) was the best performing method for all three FOV sizes $V \in \{5, 10, 15\}$, with a subset accuracy of 85.5%, 92.6% and 94.4%, respectively. The multiclass segmentation is closely followed by the traditional machine learning approaches (Section 5.4.2) based on a regression task about classifying statistics created from the instances, the best approach of these being XGBoost. The Encoder+GNN approach performed quite poorly, the reason likely being that the neighborhood context size for each vertebra is too small to obtain enough information about the location of the vertebra.

The multiclass segmentation model is further analyzed in Figure 6.2, including a per vertebra and IVD accuracy for four different FOV sizes. Furthermore, in Table 6.2 the multiclass segmentation model is compared with a few state of the art approaches. Most of them are difficult to compare, however there is one which has a very similar

MRI dataset, by Chang et al. [53], it has relatively consistent FOV of $V \in \{5, 6, 7, 8\}$, and also only uses vertebral bodies instead of full vertebrae as segmentation target.

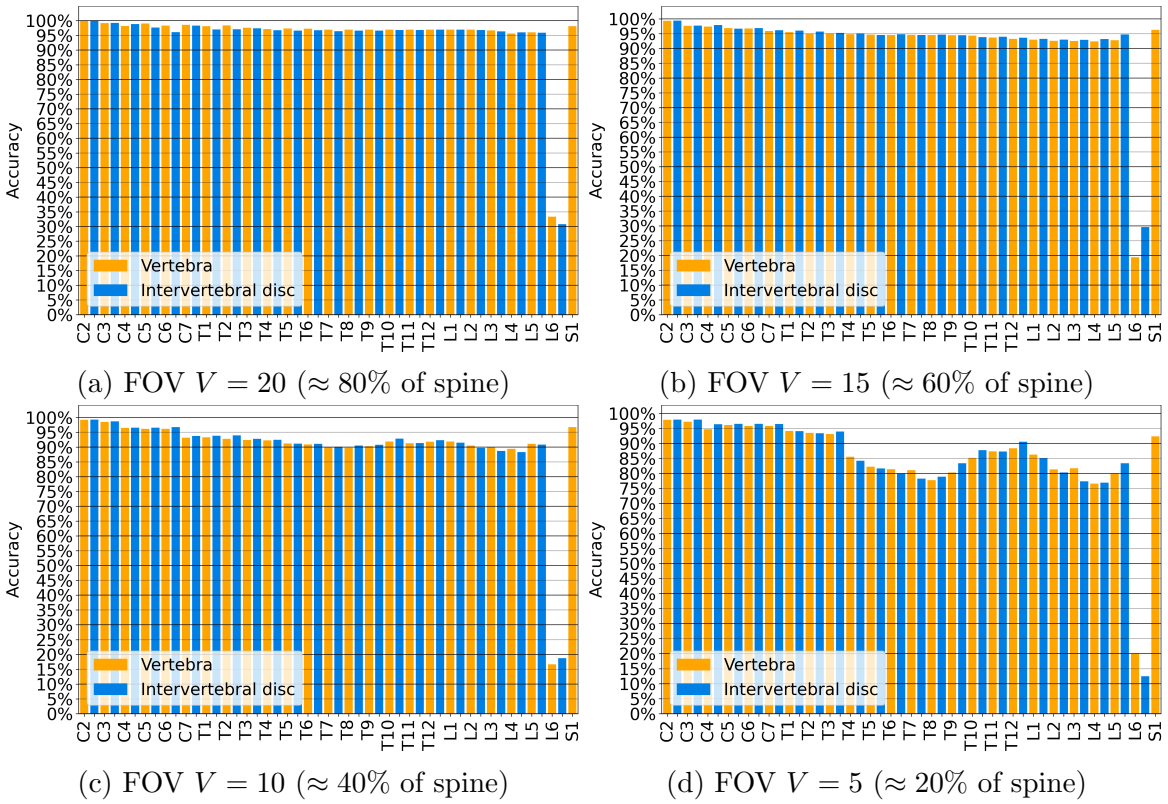
Method	Dataset	FOV	Accuracy	DSC
Lessman et al. [36]	15 CTs	$V \approx 18$	100%	0.958
	35 low-dose chest CTs	$V \approx 12$	95.7%	0.921
Payer et al. [37]	VerSe19: 374 CTs	$V \in \{5, 6, ..., 26\}$	95.0%	0.904
VerteFormer [54]	VerSe19: 374 CTs	$V \in \{5, 6, ..., 26\}$	-	0.865
	VerSe20: 300 CTs	$V \in \{5, 6, ..., 26\}$	-	0.869
Spine-transformers [55]	VerSe19: 374 CTs	$V \in \{5, 6, ..., 26\}$	96.7%	0.901
SpineParseNet [17]	215 lumbar MRIs	$V \approx 8$	-	0.875 ± 0.038
Spine-GAN [56]	253 lumbar MRIs	$V \approx 8$	-	0.870 ± 0.010
Chang et al. [53]	292 T10 to S1 MRIs	$V \in \{6, 7, 8, 9\}$	$89.3\% \pm 5.2$	0.871 ± 0.041
Ours	10,833 complete MRIs	$V = 5$	85.5%	0.799
		$V = 8$	90.9%	0.839
		$V = 10$	92.6%	0.847
		$V = 15$	94.4%	0.875

Table 6.2.: **Approximate Comparison of State-of-the-Art Segmentation Models for MRI and CT Images.** This comparison spans various datasets and FOV sizes. The FOVs are estimated based on the datasets, while the accuracy and Dice similarity coefficient (DSC) values are sourced from the respective papers. The term 'accuracy' is often used interchangeably with 'identification rate', which denotes the accuracy for each individual vertebra. There are considerably more CT-based methods, likely due to the availability of complete spinal datasets for CT. Three MRI approaches are also listed. Among these, the dataset by Chang et al. [53] is likely the most similar, since they use vertebral bodies instead of the full vertebra for segmentation, and contain several different FOV ranges.

6.2. Step 1: Segmentation Method Comparison

In this section, we compare the results from the segmentation step to identify the optimal method for the rest of the pipeline. We consider two methods: slice-wise segmentation (Section 5.2.2) and volume-based segmentation (Section 5.2.3). Both methods employ similar architectures, such as U-Net, in either the 2D variant [25] or 3D variant [26]. The models are trained on a dataset manually annotated by expert radiologists, comprising 132 MRI scans for training. After training, each model is used to segment the 30 MRI validation scans. The annotation is then used to compute scores for DSC and IoU.

Table 6.3 presents the condensed results for both methods. The DSC and IoU scores (defined in Section 2.3.2 and Section 2.3.3, respectively) are shown for each experiment. These metrics quantify the accuracy of the segmentation, with 1.0 indicating perfect segmentation and 0.0 indicating no correct segmentation.



(a) FOV $V = 20$ ($\approx 80\%$ of spine)

(b) FOV $V = 15$ ($\approx 60\%$ of spine)

(c) FOV $V = 10$ ($\approx 40\%$ of spine)

(d) FOV $V = 5$ ($\approx 20\%$ of spine)

Figure 6.2.: **Multiclass Segmentation Accuracy per Vertebra and IVD for Various FOVs.** The multiclass segmentation method (Section 5.4.4) was evaluated for four different field of view (FOV) sizes $V \in \{5, 10, 15, 20\}$ for each vertebra and IVD. The FOV size refers how many vertebrae are visible in the scan used as input for the multiclass segmentation. In all four FOVs the labeling accuracy for **L6** and **L6-S1** was poor: between 12% and 34%. In all FOVs the accuracy for **C2** and **S1** was good, which can be explained due to them being of unique shape and lacking a predecessor/successor vertebra. The overall accuracy for the rest of the vertebrae and IVDs was fairly consistent for $V \in \{10, 15, 20\}$. However, for $V = 5$, two dips can be noted around **T8** and **L4**, which can likely be explained due to the lack of curvature and other unique identifying characteristics in that region. Each of these plots was evaluated on roughly 1500 small FOV MR images sampled from the validation set of 542 patients. For each image, all visible vertebrae and IVDs were recorded, appearing between 150 and 1500 times for each plot, except for **L6** and **L6-S1**, which appeared between 20 and 40 times per plot. The variance in the occurrence is due to the four different FOV sizes, and the sampling method being biased towards representing central vertebrae and IVDs.

The best architecture was a slice-wise 2D U-Net with a resnet152 encoder and Jaccard loss, achieving a DSC of 0.919. This surpasses the approach by Streckenbach et al. [14], which used the same dataset and test-set, allowing for a direct comparison. The complete table of all 112 slice-wise segmentation experiments can be found in Appendix A.

In conclusion, unless specified otherwise, this slice-wise U-Net with a resnet152 encoder model is used as the main segmentation model.

Section / Author	Architecture	Encoder	Loss	DSC	IoU
5.2.2	**U-Net** [25]	**resnet152** [57]	**JaccardLoss**	**0.919**	**0.852**
5.2.2			DiceLoss	0.915	0.844
5.2.2			CrossEntropyLoss	0.902	0.822
5.2.2		mit_b5 [27]		0.905	0.829
5.2.2		resnet34 [57]		0.917	0.848
5.2.2		timm-regnetx_160 [58]		0.917	0.848
5.2.2	FPN [44]			0.909	0.834
5.2.2	DeepLabV3Plus [46]			0.912	0.839
5.2.2	Linknet [47]			0.918	0.849
5.2.2	PAN [48]			0.910	0.836
5.2.3	3D U-Net [50]	CNN		0.910	0.835
5.2.3	3D U-Net [50]	CNN	DiceLoss	0.905	0.829
Streckenbach et al. [14]	(Patch) 3D U-Net [26]	CNN	CrossEntropyLoss with FocalLoss	0.907	-

Table 6.3.: **Step 1: Segmentation Results Comparison.** The topmost row shows the best method, the following entries show an ablation study: cells left blank are the same as in the topmost row. The complete results can be found in Appendix A. The last row shows an approach by Streckenbach et al. [14] for the same dataset and ground truth as used in this thesis. They used a 3D patch-based approach for segmentation.

6.3. Step 2: Instance Separation Method Comparison

This section compares the two methods for instance separation, namely *connected components* and *split along IVDs*, as presented in Section 5.3. These methods are used to separate vertebrae and IVDs in the segmentation to obtain a top-down sorted list of instances. This is crucial for creating a ground truth for the large dataset of 10,833 MRI scans, improving the post-processing of the multiclass segmentation approach, and generating necessary statistics for the conventional machine learning models.

Evaluating these methods in isolation is challenging due to the absence of ground truth data for separated instances and the fact that both methods work 100% correctly on the test set upon manual inspection. Furthermore, a similar approach as before with the split of FOV sizes of $V = 5$, $V = 10$ or $V = 15$ vertebrae, does not provide additional information. The instance separation methods do not depend on additional context,

such as the multiclass segmentation does. The only prerequisite is that there are at least four centroids in the current segmentation for the *split along IVDs* approach, as otherwise the next centroids can not be inferred.

In order to decide which of these methods performs better, a heuristic is used to gauge the performance of these methods: the entire dataset of 10,833 is segmented with the best segmentation model as created in step 1, then both methods are used on each segmented scan to separate the vertebrae and IVDs, and finally, invalid separations are counted.

For the purpose of this heuristic evaluation method, a separation is considered invalid if any of the following conditions is true:

- The uppermost instance is not **C2**
- The lowermost instance is not **S1**
- The number of vertebrae is not 23, 24 or 25
- The number of discs is not 22, 23 or 24
- The number of vertebrae + discs is not 45, 47 or 49

Section	Method	Valid	Invalid	Invalid (without segmentation errors)
5.3.1	*Connected components*	9865 (91.1%)	968 (8.9%)	940 (8.7%)
5.3.2	*Split along IVDs*	10750 (99.2%)	83 (0.8%)	55 (0.5%)

Table 6.4.: **Step 2: Instance Separation Heuristic Evaluation Comparison**. The results of step 2 in the pipeline, based on the segmentation of the previous step. A validity metric is used, as there is no ground truth data in order to evaluate the methods directly. Validity specifies whether the resulting spine is possible from an anatomical perspective, if not, it is very likely an incorrect output of that method.

In this thesis, all invalid segmentations are reviewed by the authors to eliminate major segmentation errors or significant mistakes in the MRI scans. The primary objective is to evaluate step 2 in isolation. The results of this evaluation are presented in Table 6.4.

The approach *split along IVDs*, yields significantly fewer invalid instance separations (55) compared to the *connected components* approach (940). Upon manual inspection, 28 scans were identified as completely erroneous. These errors were either due to poor data quality (refer to Figure 4.2 in Section 4) or incorrect segmentation output.

In conclusion, the *split along IVDs* approach will be used by default in the anatomical labeling pipeline, unless specified otherwise. The heuristic was primarily used to justify the selection of this method.

6.4. Sacralization and Lumbarization

This section presents our findings on the occurrence of lumbarization and sacralization, compared with existing literature. Our analysis, based on a sample size of 10,833 individuals, shows occurrences of 6.6% (710 cases) for lumbarization and 3.6% (393 cases) for sacralization. These results are placed within the context of various other studies, as summarized in Table 6.5, demonstrating a higher incidence of lumbarization in our study compared to others.

It's important to acknowledge that our study has limitations. The segmentation and classification of lumbarization and sacralization were automated, lacking the manual, expert evaluation typically associated with more traditional studies. Consequently, this may affect the precision of identifying anatomical variants compared to assessments done by medical experts. However, despite this limitation, our large-scale dataset provides a valuable contribution to the understanding of these spinal anomalies. Our findings highlight the variability of spinal anatomy in a broader population, offering a significant dataset for future automated diagnostic methods and aiding in the development of more nuanced algorithms capable of handling anatomical variations in clinical settings.

Study	Sample size	Lumbarization	Sacralization
Price et al. [59]	268	4.1% (11)	-
Luboga et al. [60]	591	3.4% (20)	-
Doo et al. [61]	1340	3.2% (43)	5.1% (68)
Nakajima et al. [62]	226	6.2% (14)	2.7% (6)
Farshad-Amacker et al. [63]	770	4.8% (37)	4.4% (34)
Hahn et al. [64]	200	4.5% (9)	7.5% (15)
Peh et al. [65]	129	7.0% (9)	6.2% (8)
Hughes et al. [43]	500	9.2% (46)	4.2% (21)
Total	4024 / 3165	4.7% (189)	4.8% (152)
Ours	10833	6.6% (710)	3.6% (393)

Table 6.5.: **Occurrence of Lumbarization and Sacralization in Literature in Comparison to our Results.** The split of the sample size in the "Total" row denotes the totals for lumbarization and sacralization, which accounts for the lack of information in some studies about sacralization.

7. Discussion

In this section, we set the work presented in this thesis within the broader context of current research (Section 7.1) and explore avenues for future work (Section 7.2).

7.1. Situating the Anatomical Labeling Pipeline within the Medical Imaging Field

The novel three-step approach introduced in this thesis for segmentation in small field of view (FOV) magnetic resonance imaging (MRI) scans fills a previously unfilled niche. There have been several good approaches for anatomical labeling CT in arbitrary FOV sizes [37, 55, 66, 54], however for MRIs there is little in that regard. Chang et al. [53] do have some variation in the FOV size and location, but the data is only lumbar, making the uses of such a model fairly limited. Being able to use our anatomical labeling model for any kind of spinal MRI greatly helps its usefulness and applicability. However, the dataset is still the closest to ours, due it being MRI, having small FOV and using vertebral body segmentation, making a comparison feasible, although not entirely accurate. For a fairer comparison, in Table 6.2 FOV size of $V = 8$ was used in the pipeline, resulting in a subset accuracy of 90.9% and a DSC of 0.839. In comparison Chang et al. have noted a slightly lower accuracy of 89.3% and a slightly higher DSC of 0.871. All in all, these results are comparable, meaning that the anatomical labeling pipeline has similar results to some of the state-of-the-art methods for anatomical labeling, in addition to being more general, as our approach has to be able to label any part of the spine.

The methods selected for each step: slice-wise segmentation for step 1, instance separation utilizing intervertebral disc (IVD) for step 2, and slice-wise multiclass segmentation for step 3, were shown to be particularly effective. This efficacy is notable especially in cases of small FOV segmentations, where approximately five vertebrae are visible, and neither the **S1** nor the **C2** vertebrae can be identified. This focus on small FOV segmentations sets our work apart from many existing studies, which often rely on datasets where these vertebrae are visible, thus simplifying the task of anatomical labeling [17, 56].

Another significant contribution of this thesis is the insights gained on sacralization and lumbarization, derived from one of the largest datasets of its kind: out of the 10,833

patients, 710 (6.6%) were categorized as lumbarization and 393 (3.6%) were categorized as sacralization. These conditions, which involve variations in the number of lumbar vertebrae, present notable challenges in both clinical diagnosis and medical imaging. Our extensive dataset enabled a detailed examination of these variations, offering a rare opportunity to observe and analyze these anomalies in greater depth. Although, a key limitation is that this categorization was achieved automatically, without expert supervision, which might affect the accuracy of these categorizations.

However, it is important to acknowledge the limitations of this study. We have only used a single dataset, which is very large, but it only contains T2 MRI scans. The segmentation models could struggle with T1 MRI scans without further training. Another limitation is that the best method for instance separation in step 2, *split by IVD*, relies on vertebral bodies as segmentations. For a dataset such as is used by Pang et al. [17] a different instance separation algorithm would have to be developed. Furthermore, currently *split by IVD* fails for segmentations with less than four centroids, even though the actual multiclass segmentation might even have a chance of labeling these points.

7.2. Future Work

Looking ahead, there are many exciting directions for further research. In the following some of them are listed.

1. Experimentation with alternative datasets, especially those differing significantly in size, quality, or anatomical coverage, could provide deeper insights into the generalizability and adaptability of the proposed methods. Applying the developed techniques to CT datasets presents another valuable research path. In CT datasets, IVDs would play a much smaller role, due to CTs scans not capturing soft-tissues well. CT scans further provide a challenge in that they are more cuboid in shape, for example, a typical CT scan is of size $512 \times 512 \times 600$, whereas an MRI is usually around $20 \times 400 \times 1000$. The major difference is in the sagittal axis, where CTs are roughly 25 times larger, making the image considerably more time-intensive to compute. Another significant difference in the datasets would be more prevalent pathologies, as CT scans are potentially dangerous to humans due to radiation, they are only done when there are already problems. For example, the VerSe19 [7, 38, 39] CT-dataset includes many pathologies.

2. The MRI dataset used in this thesis could be augmented and balanced further, in order to improve generalization of the segmentation models. The augmentations, especially ones which crop the input MR image, could help with the issue where vertebrae near the edges of image are not segmented correctly, necessitating an evaluation adjustment for subset accuracy such as shown in Section 6.1.2. Furthermore, the anatomical labeling currently performs very badly for detecting

lumbarization (the presence of **L6** and **L6-S1**) in small FOV MRI scans. Creating a more balanced dataset, such that lumbarization and sacralization appear more commonly could improve the performance for this task. However, some radiologists claim that the only reliable way to detect LSTVs is by counting from the top [61], which might make this task difficult.

3. Extending the methods developed in this thesis, especially instance separation in Section 5.3, for the full vertebra is necessary in order to be able to evaluate the methods on other datasets. Currently, a plane is used to separate two adjacent vertebral bodies, using this approach for the full vertebra would cause problems, due to the plane incorrectly splitting the back portion of the vertebra. The full vertebra includes the parts surrounding the spinal canal, whereas the vertebral body only includes the voluminous part in front of the vertebra. For a comparison between a segmentation for a full vertebra and only the vertebral body, see Figure 2.2 in Section 2.1.

4. An improved end-to-end anatomical labeling method could be developed to streamline the process and potentially improve performance. For example, a fully neural approach to label everything would likely generalize better and have better performance than some of the algorithms developed in this thesis.

5. The developed methods could be integrated into disease prediction models to provide more comprehensive and accurate predictions. Most MRI scans are small FOV scans performed for localized detection of pathologies. The proposed methods in this thesis accurately segment and label small FOV scans, creating a strong foundation for further models.

6. The performance of the developed methods could be compared to human performance to provide a benchmark and identify areas for improvement. Once models such as the one proposed in this thesis reach similar or even better performance than expert radiologists do, they could be used to segment and label MR images without much supervision.

7. Statistical analyses based on annotated images could be conducted to gain deeper insights into the data and the performance of the methods. Due to the large size of the dataset, many insights such as average vertebra width, height, volume, and orientation could be extrapolated. Using the age metadata, certain statistics on how vertebrae or IVDs change during one's life could be calculated, for example, what the average IVD height is as compared to the patient's age.

8. The methods could be applied to other MRIs beyond spines to test their versatility and adaptability. Some of the approaches, such as the instance separation in Section 5.3, are very specific to the task of spine anatomical labeling, but others could also work for other body parts, such as the multiclass segmentation in

Section 5.4.4.

9. Exploring refinements in the algorithm to enhance its accuracy and efficiency, or adapting the methodology to incorporate recent advances in machine learning and hardware improvements, could further improve outcomes and broaden the range of practical applications. For example, in the future, volume segmentation, as shown in Section 5.2.3, will likely perform better than slice-wise segmentation. There are many features in the sagittal dimension, such as ribs being connected to the thoracic vertebrae, which can help models label vertebrae correctly.

In conclusion, the methods developed in this thesis have shown promising results in the segmentation and labeling of spinal structures in MRI scans. However, there are several avenues for future research and improvement. These include experimenting with alternative datasets, further augmenting and balancing the MRI dataset used, extending the methods for full vertebra segmentation, and developing an improved end-to-end method. Additionally, applying these methods to other MRI scans beyond spines could test their versatility and adaptability. Statistical analyses based on annotated images could provide deeper insights into the data and the performance of the methods. Lastly, comparing the performance of these methods to human performance could provide a benchmark and identify areas for improvement. These advancements could potentially lead to more comprehensive and accurate disease prediction models, and streamline the process of MRI analysis.

8. Conclusion

This thesis has presented a robust and efficient three-step segmentation pipeline that competes with the best methods currently available for the segmentation of small field of view (FOV) magnetic resonance (MR) images. The pipeline's performance was evaluated using small FOV MRI scans, with the number of tested field of view sizes being $V \in \{5, 10, 15\}$. The subset accuracy achieved was 85.5%, 92.6%, and 94.4% respectively, demonstrating the pipeline's effectiveness and adaptability to different imaging conditions.

In addition to the development of the segmentation pipeline, this thesis also conducted an extensive study on lumbarization and sacralization using a large dataset. The segmentation pipeline was instrumental in facilitating this study. Out of the 10833 spines analyzed, lumbarization was observed in 710 (6.6%) cases, and sacralization in 393 (3.6%) cases. These findings, while slightly deviating from several previously analyzed studies, provide valuable insights into the prevalence of these conditions and underscore the importance of accurate segmentation in facilitating such analyses.

The source code developed for this thesis has been made publicly available under an open-source license on GitHub[1]. This not only ensures transparency and reproducibility of the results presented in this thesis but also provides a valuable resource for the scientific community. Researchers and developers can use, modify, and build upon this code for their own studies, potentially leading to further advancements in the field of small FOV MRI anatomical labeling.

In conclusion, this thesis has made significant contributions to the field of small FOV MRI image segmentation and anatomical labeling, both through the development of a state-of-the-art segmentation pipeline and through the insights gained from the lumbarization and sacralization study. The open-source availability of the code further enhances the impact of this work, paving the way for future research and development in this area.

[1] https://github.com/LiquidFun/Spine

Acknowledgements

The thesis was only possible thanks to Dr. Marc-André Weber and Dr. Felix Streckenbach who proposed the idea, provided access to the data, and gave in-depth insight into current research on the topic from a medical perspective.

Thanks to Prof. Martin Becker and Gundram Leifert for extensive supervision which included: organization, guidance, proof-reading, and many ideas during development and writing of the thesis.

Thanks to Lukas Großehagenbrock for proofreading the thesis.

OpenAI's ChatGPT was used throughout this thesis for proofreading and improving the phrasing. For Section 2 it was used extensively to write many parts.

9. Bibliography

[1] D. Hoy et al. "The global burden of low back pain: estimates from the Global Burden of Disease 2010 study". In: *Annals of the Rheumatic Diseases* 73.6 (Mar. 2014), pp. 968–974. DOI: 10.1136/annrheumdis-2013-204428.

[2] J. Yao, J. E. Burns, H. Munoz, and R. M. Summers. "Detection of Vertebral Body Fractures Based on Cortical Shell Unwrapping". In: (2012), pp. 509–516. DOI: 10.1007/978-3-642-33454-2_63.

[3] D. Forsberg, C. Lundström, M. Andersson, L. Vavruch, H. Tropp, and H. Knutsson. "Fully automatic measurements of axial vertebral rotation for assessment of spinal deformity in idiopathic scoliosis". In: *Physics in Medicine and Biology* 58.6 (Feb. 2013), pp. 1775–1787. DOI: http://dx.doi.org/10.1088/0031-9155/58/6/1775.

[4] D. Knez, B. Likar, F. Pernus, and T. Vrtovec. "Computer-Assisted Screw Size and Insertion Trajectory Planning for Pedicle Screw Placement Surgery". In: *IEEE Transactions on Medical Imaging* 35.6 (June 2016), pp. 1420–1430. DOI: 10.1109/tmi.2016.2514530.

[5] A. L. Williams, A. Al-Busaidi, P. J. Sparrow, J. E. Adams, and R. W. Whitehouse. "Under-reporting of osteoporotic vertebral fractures on computed tomography". In: *European Journal of Radiology* 69.1 (Jan. 2009), pp. 179–183. DOI: 10.1016/j.ejrad.2007.08.028.

[6] A. Sekuboyina, J. Kukačka, J. S. Kirschke, B. H. Menze, and A. Valentinitsch. "Attention-Driven Deep Learning for Pathological Spine Segmentation". In: (2018), pp. 108–119. DOI: https://doi.org/10.1007/978-3-319-74113-0_10.

[7] M. T. Löffler, A. Sekuboyina, A. Jacob, A.-L. Grau, A. Scharr, M. E. Husseini, M. Kallweit, C. Zimmer, T. Baum, and J. S. Kirschke. "A Vertebral Segmentation Dataset with Fracture Grading". In: *Radiology: Artificial Intelligence* 2.4 (July 2020), e190138. DOI: https://doi.org/10.1148/ryai.2020190138.

[8] L. N. Metz and S. Burch. "Computer-Assisted Surgical Planning and Image-Guided Surgical Navigation in Refractory Adult Scoliosis Surgery". In: *Spine* 33.9 (Apr. 2008), E287–E292. DOI: 10.1097/BRS.0b013e31816d256e.

[9] E. Von Der Lippe, L. Krause, M. Porst, A. Wengler, J. Leddin, A. Müller, M.-L. Zeisler, A. Anton, and A. Rommel. "Prevalence of back and neck pain in Germany. Results from the BURDEN 2020 Burden of Disease Study". en. In: (2021). DOI: 10.25646/7855.

[10] P. Q. Duy, I. Ikuta, M. H. Johnson, M. Davis, and V. M. Zohrabian. "MRI in Spine Trauma". In: (2020), pp. 31–86. DOI: https://doi.org/10.1007/978-3-030-43627-8_3.

[11] J. F. Talbott, J. F. Burke, A. Callen, V. Shah, J. Narvid, and S. S. Dhall. "Imaging of Spinal Trauma with MRI: A Practical Guide". In: (2022), pp. 181–201. DOI: https://doi.org/10.1007/978-3-030-92111-8_13.

[12] S. Zhang and D. Metaxas. "On the Challenges and Perspectives of Foundation Models for Medical Image Analysis". In: (2023). DOI: https://doi.org/10.48550/arXiv.2306.05705.

[13] A. Kalluvila, N. Koonjoo, D. Bhutto, M. Rockenbach, and M. S. Rosen. "Synthetic Low-Field MRI Super-Resolution Via Nested U-Net Architecture". In: (2022). DOI: https://doi.org/10.48550/arXiv.2211.15047.

[14] F. Streckenbach, G. Leifert, T. Beyer, A. Mesanovic, H. Wäscher, D. Cantré, S. Langner, M.-A. Weber, and T. Lindner. "Application of a Deep Learning Approach to Analyze Large-Scale MRI Data of the Spine". In: *Healthcare* 10.11 (Oct. 2022), p. 2132. DOI: https://doi.org/10.3390/healthcare10112132.

[15] J. Oliver and A. Middleditch. "Functional Anatomy of the Spine". In: (1991). DOI: https://doi.org/10.1016/C2009-0-49465-0.

[16] J. M. Vital and D. T. Cawley. "Spinal Anatomy". In: (2020). Ed. by J. M. Vital and D. T. Cawley. DOI: https://doi.org/10.1007/978-3-030-20925-4.

[17] S. Pang, C. Pang, L. Zhao, Y. Chen, Z. Su, Y. Zhou, M. Huang, W. Yang, H. Lu, and Q. Feng. "SpineParseNet: Spine Parsing for Volumetric MR Image by a Two-Stage Segmentation Framework With Semantic Image Representation". In: *IEEE Transactions on Medical Imaging* 40.1 (Jan. 2021), pp. 262–273. DOI: 10.1109/TMI.2020.3025087.

[18] S. Pang, C. Pang, Z. Su, L. Lin, L. Zhao, Y. Chen, Y. Zhou, H. Lu, and Q. Feng. "DGMSNet: Spine segmentation for MR image by a detection-guided mixed-supervised segmentation network". In: *Medical Image Analysis* 75 (Jan. 2022), p. 102261. DOI: https://doi.org/10.1016/j.media.2021.102261.

[19] X. Zhao, Y. Chao, H. Zhang, B. Yao, and L. He. "An Efficient Connected-Component Labeling Algorithm for 3-D Binary Images". In: *IEEE Open Journal of the Computer Society* 4 (2023), pp. 1–12. DOI: 10.1109/OJCS.2022.3233088.

[20] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[21] T. Sørensen. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter. Munksgaard, 1948. URL: https://books.google.dk/books?id=rpS8GAAACAAJ.

[22] L. R. Dice. "Measures of the Amount of Ecologic Association Between Species". In: *Ecology* 26.3 (July 1945), pp. 297–302. DOI: https://doi.org/10.2307%2F1932409.

[23] P. Jaccard. "The Distribution of Flora in the Alpine Zone". In: *New Phytologist* 11.2 (Feb. 1912), pp. 37–50. DOI: https://doi.org/10.1111%2Fj.1469-8137.1912.tb05611.x.

[24] L. Wu, P. Cui, J. Pei, and L. Zhao. "Graph Neural Networks: Foundations, Frontiers, and Applications". In: (2022). DOI: https://doi.org/10.1007/978-981-16-6054-2.

[25] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: (2015), pp. 234–241. DOI: https://doi.org/10.1007/978-3-319-24574-4_28.

[26] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". In: (2016). DOI: https://doi.org/10.48550/arXiv.1606.06650.

[27] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers". In: (2021). DOI: https://doi.org/10.48550/arXiv.2105.15203.

[28] D. Richmond, D. Kainmueller, B. Glocker, C. Rother, and G. Myers. "Uncertainty-Driven Forest Predictors for Vertebra Localization and Segmentation". In: (2015), pp. 653–660. DOI: https://doi.org/10.1007/978-3-319-24553-9_80.

[29] B. Glocker, J. Feulner, A. Criminisi, D. R. Haynor, and E. Konukoglu. "Automatic Localization and Identification of Vertebrae in Arbitrary Field-of-View CT Scans". In: (2012), pp. 590–598. DOI: https://doi.org/10.1007/978-3-642-33454-2_73.

[30] Z. Peng, J. Zhong, W. Wee, and J.-h. Lee. "Automated Vertebra Detection and Segmentation from the Whole Spine MR Images". In: (2005). DOI: DOI:10.1109/IEMBS.2005.1616983.

[31] F. Wang, K. Zheng, L. Lu, J. Xiao, M. Wu, and S. Miao. "Automatic vertebra localization and identification in CT by spine rectification and anatomically-constrained optimization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5280–5288.

[32] D. Forsberg. "Atlas-Based Segmentation of the Thoracic and Lumbar Vertebrae". In: (2015), pp. 215–220. DOI: https://doi.org/10.1007/978-3-319-14148-0_18.

[33] S. K. Michopoulou, L. Costaridou, E. Panagiotopoulos, R. Speller, G. Panayiotakis, and A. Todd-Pokropek. "Atlas-Based Segmentation of Degenerated Lumbar Intervertebral Discs From MR Images of the Spine". In: *IEEE Transactions on Biomedical Engineering* 56.9 (Sept. 2009), pp. 2225–2231. DOI: https://doi.org/10.1109/TBME.2009.2019765.

[34] C. Wang and D. Forsberg. "Segmentation of Intervertebral Discs in 3D MRI Data Using Multi-atlas Based Registration". In: (2016), pp. 107–116. DOI: https://doi.org/10.1007/978-3-319-41827-8_10.

[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: (2018). DOI: https://doi.org/10.48550/arXiv.1802.02611.

[36] N. Lessmann, B. van Ginneken, P. A. de Jong, and I. Išgum. "Iterative fully convolutional neural networks for automatic vertebra segmentation and identification". In: *Medical Image Analysis* 53 (Apr. 2019), pp. 142–155. DOI: https://doi.org/10.1016/j.media.2019.02.005.

[37] C. Payer, D. Štern, H. Bischof, and M. Urschler. "Coarse to Fine Vertebrae Localization and Segmentation with SpatialConfiguration-Net and U-Net". In: (2020). DOI: 10.5220/0008975201240133.

[38] A. Sekuboyina et al. "VerSe: A Vertebrae labelling and segmentation benchmark for multi-detector CT images". In: *Medical Image Analysis* 73 (Oct. 2021), p. 102166. DOI: https://doi.org/10.1016/j.media.2021.102166.

[39] H. Liebl et al. "A computed tomography vertebral segmentation dataset with anatomical variations and multi-vendor scanner data". In: *Scientific Data* 8.1 (Oct. 2021). DOI: 10.1038/s41597-021-01060-0.

[40] C. Payer, D. Štern, H. Bischof, and M. Urschler. "Integrating spatial configuration into heatmap regression based CNNs for landmark localization". In: *Medical Image Analysis* 54 (May 2019), pp. 207–219. DOI: https://doi.org/10.1016/j.media.2019.03.007.

[41] *Durchschnittsalter der Bevölkerung ab 1871*. URL: https://www.bib.bund.de/DE/Fakten/Fakt/B19-Durchschnittsalter-Bevoelkerung-ab-1871.html (visited on 09/08/2023).

[42] P. G. Tini, C. Wieser, and W. M. Zinn. "The Transitional Vertebra of the Lumbosacral Spine: Its Radiological Classification, Incidence, Prevalence, and Clinical Significance". In: *Rheumatology* 16.3 (1977), pp. 180–185. DOI: 10.1093/rheumatology/16.3.180.

[43] R. J. Hughes and A. Saifuddin. "Numbering of Lumbosacral Transitional Vertebrae on MRI: Role of the Iliolumbar Ligaments". In: *American Journal of Roentgenology* 187.1 (July 2006), W59–W65. DOI: 10.2214/AJR.05.0415.

[44] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature Pyramid Networks for Object Detection". In: (2016). DOI: https://doi.org/10.48550/arXiv.1612.03144.

[45] T. Fan, G. Wang, Y. Li, and H. Wang. "MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation". In: *IEEE Access* 8 (2020), pp. 179656–179665. DOI: 10.1109/ACCESS.2020.3025372.

[46] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking Atrous Convolution for Semantic Image Segmentation". In: (2017). DOI: https://doi.org/10.48550/arXiv.1706.05587.

[47] A. Chaurasia and E. Culurciello. "LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation". In: (2017). DOI: https://doi.org/10.48550/arXiv.1707.03718.

[48] H. Li, P. Xiong, J. An, and L. Wang. "Pyramid Attention Network for Semantic Segmentation". In: (2018). DOI: https://doi.org/10.48550/arXiv.1805.10180.

[49] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: (2014). DOI: https://doi.org/10.48550/arXiv.1412.6980.

[50] E. Kerfoot, J. Clough, I. Oksuz, J. Lee, A. P. King, and J. A. Schnabel. "Left-Ventricle Quantification Using Residual U-Net". In: (2019), pp. 371–380. DOI: https://doi.org/10.1007/978-3-030-12029-0_40.

[51] T. K. Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[52] T. Chen and C. Guestrin. "XGBoost". In: (Aug. 2016). DOI: https://doi.org/10.1145/2939672.2939785.

[53] H. Chang, S. Zhao, H. Zheng, Y. Chen, and S. Li. "Multi-vertebrae Segmentation from Arbitrary Spine MR Images Under Global View". In: (2020), pp. 702–711. DOI: https://doi.org/10.1007/978-3-030-59725-2_68.

[54] X. You, Y. Gu, Y. Liu, S. Lu, X. Tang, and J. Yang. "VerteFormer: A single-staged Transformer network for vertebrae segmentation from CT images with arbitrary field of views". In: *Medical Physics* 50.10 (May 2023), pp. 6296–6318. DOI: https://doi.org/10.1002/mp.16467.

[55] R. Tao, W. Liu, and G. Zheng. "Spine-transformers: Vertebra labeling and segmentation in arbitrary field-of-view spine CTs via 3D transformers". In: *Medical Image Analysis* 75 (Jan. 2022), p. 102258. DOI: https://doi.org/10.1016/j.media.2021.102258.

[56] Z. Han, B. Wei, A. Mercado, S. Leung, and S. Li. "Spine-GAN: Semantic segmentation of multiple spinal structures". In: *Medical Image Analysis* 50 (Dec. 2018), pp. 23–35. DOI: https://doi.org/10.1016/j.media.2018.08.005.

[57] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: (2015). DOI: https://doi.org/10.48550/arXiv.1512.03385.

[58] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár. "Designing Network Design Spaces". In: (2020). DOI: https://doi.org/10.48550/arXiv.2003.13678.

[59] R. Price, M. Okamoto, J. L. Huec, and K. Hasegawa. "Normative spino-pelvic parameters in patients with the lumbarization of S1 compared to a normal asymptomatic population". In: *European Spine Journal* 25.11 (Sept. 2016), pp. 3694–3698. DOI: 10.1007/s00586-016-4794-8.

[60] S. Luboga. "Supernumerary lumbar vertebrae in human skeletons at the Galloway Osteological Collection of Makerere University, Kampala". In: *East African medical journal* 77.1 (Jan. 2000), pp. 16–19. ISSN: 0012-835X. URL: http://europepmc.org/abstract/MED/10944832.

[61] A. R. Doo, J. Lee, G. E. Yeo, K. H. Lee, Y. S. Kim, J. H. Mun, Y. J. Han, and J.-S. Son. "The prevalence and clinical significance of transitional vertebrae: a radiologic investigation using whole spine spiral three-dimensional computed tomographic images". In: *Anesthesia and Pain Medicine* 15.1 (Jan. 2020), pp. 103–110. DOI: 10.17085/apm.2020.15.1.103.

[62] A. Nakajima, A. Usui, Y. Hosokai, Y. Kawasumi, K. Abiko, M. Funayama, and H. Saito. "The prevalence of morphological changes in the thoracolumbar spine on whole-spine computed tomographic images". In: *Insights into Imaging* 5.1 (Sept. 2013), pp. 77–83. DOI: 10.1007/s13244-013-0286-0.

[63] N. A. Farshad-Amacker, A. Aichmair, R. J. Herzog, and M. Farshad. "Merits of different anatomical landmarks for correct numbering of the lumbar vertebrae in lumbosacral transitional anomalies". In: *European Spine Journal* 24.3 (Sept. 2014), pp. 600–608. DOI: 10.1007/s00586-014-3573-7.

[64] P. Y. Hahn, J. J. Strobel, and F. J. Hahn. "Verification of lumbosacral segments on MR images: identification of transitional vertebrae." In: *Radiology* 182.2 (Feb. 1992), pp. 580–581. DOI: 10.1148/radiology.182.2.1732988.

[65] W. C. G. Peh, T. H. Siu, and J. H. M. Chan. "Determining the Lumbar Vertebral Segments on Magnetic Resonance Imaging". In: *Spine* 24.17 (Sept. 1999), p. 1852. DOI: 10.1097/00007632-199909010-00017.

[66] Y. Zhang, X. Ji, W. Liu, Z. Li, J. Zhang, S. Liu, W. Zhong, L. Hu, and W. Li. "A Spine Segmentation Method under an Arbitrary Field of View Based on 3D Swin Transformer". In: *International Journal of Intelligent Systems* 2023 (Oct. 2023). Ed. by M. R. Khosravi, pp. 1–16. DOI: https://doi.org/10.1155/2023/8686471.

[67] M. Tan and Q. V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: (2019). DOI: https://doi.org/10.48550/arXiv.1905.11946.

[68] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan. "MobileOne: An Improved One millisecond Mobile Backbone". In: (2022). DOI: https://doi.org/10.48550/arXiv.2206.04040.

[69] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. "Aggregated Residual Transformations for Deep Neural Networks". In: (2016). DOI: https://doi.org/10.48550/arXiv.1611.05431.

[70] A. van Hilten, S. A. Kushner, M. Kayser, M. A. Ikram, H. H. H. Adams, C. C. W. Klaver, W. J. Niessen, and G. V. Roshchupkin. "GenNet framework: interpretable deep learning for predicting phenotypes from genetic data". In: *Communications Biology* 4.1 (Sept. 2021). DOI: https://doi.org/10.1038/s42003-021-02622-z.

[71] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr. "Res2Net: A New Multi-Scale Backbone Architecture". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.2 (Feb. 2021), pp. 652–662. DOI: https://doi.org/10.1109/TPAMI.2019.2938758.

[72] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: (2014). DOI: https://doi.org/10.48550/arXiv.1409.1556.

[73] P. Iakubovskii. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2019.

A. Appendix Complete Step 1 Results

Section	Architecture	Encoder	Loss	DSC	IoU
5.2.2	Unet [25]	efficientnet-b0 [67]	DiceLoss	0.909	0.834
5.2.2	Unet [25]	efficientnet-b1 [67]	DiceLoss	0.904	0.826
5.2.2	Unet [25]	efficientnet-b2 [67]	DiceLoss	0.900	0.819
5.2.2	Unet [25]	efficientnet-b3 [67]	DiceLoss	0.909	0.834
5.2.2	Unet [25]	efficientnet-b4 [67]	DiceLoss	0.917	0.848
5.2.2	Unet [25]	efficientnet-b5 [67]	DiceLoss	0.910	0.837
5.2.2	Unet [25]	mit_b0 [27]	DiceLoss	0.902	0.822
5.2.2	Unet [25]	mit_b1 [27]	DiceLoss	0.896	0.813
5.2.2	Unet [25]	mit_b2 [27]	DiceLoss	0.904	0.825
5.2.2	Unet [25]	mit_b3 [27]	DiceLoss	0.886	0.797
5.2.2	Unet [25]	mit_b4 [27]	DiceLoss	0.903	0.825
5.2.2	Unet [25]	mit_b5 [27]	DiceLoss	0.904	0.825
5.2.2	Unet [25]	mit_b5 [27]	CrossEntropyLoss	0.902	0.823
5.2.2	Unet [25]	mit_b5 [27]	JaccardLoss	0.905	0.829
5.2.2	Unet [25]	mobileone_s0 [68]	DiceLoss	0.898	0.817
5.2.2	Unet [25]	mobileone_s1 [68]	DiceLoss	0.901	0.821
5.2.2	Unet [25]	mobileone_s2 [68]	DiceLoss	0.899	0.818
5.2.2	Unet [25]	mobileone_s3 [68]	DiceLoss	0.903	0.824
5.2.2	Unet [25]	resnet18 [57]	DiceLoss	0.911	0.837
5.2.2	Unet [25]	resnet34 [57]	DiceLoss	0.917	0.848
5.2.2	Unet [25]	resnet34 [57]	CrossEntropyLoss	0.913	0.841
5.2.2	Unet [25]	resnet34 [57]	JaccardLoss	0.917	0.848
5.2.2	Unet [25]	resnet50 [57]	DiceLoss	0.911	0.838
5.2.2	Unet [25]	resnet101 [57]	DiceLoss	0.910	0.836
5.2.2	Unet [25]	resnet152 [57]	DiceLoss	0.915	0.844
5.2.2	Unet [25]	resnet152 [57]	CrossEntropyLoss	0.902	0.822
5.2.2	Unet [25]	resnet152 [57]	JaccardLoss	**0.919***	**0.852***
5.2.2	Unet [25]	resnext50_32x4d [69]	DiceLoss	0.904	0.827
5.2.2	Unet [25]	resnext101_32x8d [69]	DiceLoss	0.897	0.814
5.2.2	Unet [25]	timm-efficientnet-b0 [67]	DiceLoss	0.907	0.832
5.2.2	Unet [25]	timm-efficientnet-b1 [67]	DiceLoss	0.914	0.842
5.2.2	Unet [25]	timm-efficientnet-b2 [67]	DiceLoss	0.912	0.840
5.2.2	Unet [25]	timm-efficientnet-b3 [67]	DiceLoss	0.913	0.842
5.2.2	Unet [25]	timm-efficientnet-b4 [67]	DiceLoss	0.911	0.837
5.2.2	Unet [25]	timm-gernet_l [70]	DiceLoss	0.897	0.815
5.2.2	Unet [25]	timm-gernet_s [70]	DiceLoss	0.909	0.834
5.2.2	Unet [25]	timm-regnetx_002 [58]	DiceLoss	0.907	0.831
5.2.2	Unet [25]	timm-regnetx_004 [58]	DiceLoss	0.901	0.822
5.2.2	Unet [25]	timm-regnetx_006 [58]	DiceLoss	0.904	0.826
5.2.2	Unet [25]	timm-regnetx_008 [58]	DiceLoss	0.906	0.829

5.2.2	Unet [25]	timm-regnetx_016 [58]	DiceLoss	0.897	0.815
5.2.2	Unet [25]	timm-regnetx_032 [58]	DiceLoss	0.908	0.833
5.2.2	Unet [25]	timm-regnetx_040 [58]	DiceLoss	0.902	0.823
5.2.2	Unet [25]	timm-regnetx_064 [58]	DiceLoss	0.900	0.820
5.2.2	Unet [25]	timm-regnetx_080 [58]	DiceLoss	0.906	0.829
5.2.2	Unet [25]	timm-regnetx_120 [58]	DiceLoss	0.901	0.821
5.2.2	Unet [25]	timm-regnetx_160 [58]	DiceLoss	0.908	0.832
5.2.2	Unet [25]	timm-regnetx_160 [58]	CrossEntropyLoss	0.899	0.818
5.2.2	Unet [25]	timm-regnetx_160 [58]	JaccardLoss	0.917	0.848
5.2.2	Unet [25]	timm-res2net50_14w_8s [71]	DiceLoss	0.910	0.837
5.2.2	Unet [25]	timm-res2net50_26w_4s [71]	DiceLoss	0.908	0.832
5.2.2	Unet [25]	timm-res2net50_26w_6s [71]	DiceLoss	0.899	0.818
5.2.2	Unet [25]	timm-res2net50_26w_8s [71]	DiceLoss	0.893	0.807
5.2.2	Unet [25]	timm-res2net50_48w_2s [71]	DiceLoss	0.906	0.829
5.2.2	Unet [25]	timm-res2net101_26w_4s [71]	DiceLoss	0.906	0.829
5.2.2	Unet [25]	timm-res2next50	DiceLoss	0.880	0.787
5.2.2	Unet [25]	vgg11 [72]	DiceLoss	0.915	0.845
5.2.2	Unet [25]	vgg11_bn [72]	DiceLoss	0.907	0.831
5.2.2	Unet [25]	vgg13 [72]	DiceLoss	0.902	0.823
5.2.2	Unet [25]	vgg13_bn [72]	DiceLoss	0.907	0.832
5.2.2	Unet [25]	vgg16 [72]	DiceLoss	0.912	0.840
5.2.2	Unet [25]	vgg16_bn [72]	DiceLoss	0.914	0.842
5.2.2	Unet [25]	vgg19 [72]	DiceLoss	0.916	0.846
5.2.2	Unet [25]	vgg19_bn [72]	DiceLoss	0.915	0.844
5.2.2	FPN [44]	mit_b5 [27]	JaccardLoss	0.905	0.828
5.2.2	FPN [44]	mit_b5 [27]	CrossEntropyLoss	0.895	0.810
5.2.2	FPN [44]	mit_b5 [27]	DiceLoss	0.906	0.829
5.2.2	FPN [44]	resnet34 [57]	CrossEntropyLoss	0.905	0.826
5.2.2	FPN [44]	resnet34 [57]	DiceLoss	0.901	0.820
5.2.2	FPN [44]	resnet34 [57]	JaccardLoss	0.889	0.801
5.2.2	FPN [44]	resnet152 [57]	CrossEntropyLoss	0.886	0.796
5.2.2	FPN [44]	resnet152 [57]	JaccardLoss	**0.909**	**0.834**
5.2.2	FPN [44]	resnet152 [57]	DiceLoss	0.904	0.825
5.2.2	FPN [44]	timm-regnetx_160 [58]	CrossEntropyLoss	0.890	0.804
5.2.2	FPN [44]	timm-regnetx_160 [58]	JaccardLoss	0.906	0.829
5.2.2	FPN [44]	timm-regnetx_160 [58]	DiceLoss	0.906	0.830
5.2.2	MAnet [45]	mit_b5 [27]	CrossEntropyLoss	0.895	0.811
5.2.2	MAnet [45]	mit_b5 [27]	JaccardLoss	0.905	0.828
5.2.2	MAnet [45]	mit_b5 [27]	DiceLoss	0.897	0.815
5.2.2	MAnet [45]	resnet34 [57]	CrossEntropyLoss	0.909	0.833
5.2.2	MAnet [45]	resnet34 [57]	JaccardLoss	0.897	0.814
5.2.2	MAnet [45]	resnet34 [57]	DiceLoss	**0.918**	**0.849**
5.2.2	DeepLabV3Plus [46]	resnet34 [57]	CrossEntropyLoss	0.904	0.825
5.2.2	DeepLabV3Plus [46]	resnet34 [57]	JaccardLoss	0.909	0.833
5.2.2	DeepLabV3Plus [46]	resnet34 [57]	DiceLoss	0.907	0.830
5.2.2	DeepLabV3Plus [46]	resnet152 [57]	CrossEntropyLoss	0.894	0.810
5.2.2	DeepLabV3Plus [46]	resnet152 [57]	JaccardLoss	**0.912**	**0.839**
5.2.2	DeepLabV3Plus [46]	resnet152 [57]	DiceLoss	0.909	0.834
5.2.2	DeepLabV3Plus [46]	timm-regnetx_160 [58]	CrossEntropyLoss	0.894	0.810
5.2.2	DeepLabV3Plus [46]	timm-regnetx_160 [58]	JaccardLoss	0.910	0.836
5.2.2	DeepLabV3Plus [46]	timm-regnetx_160 [58]	DiceLoss	0.911	0.837
5.2.2	Linknet [47]	resnet34 [57]	CrossEntropyLoss	0.901	0.820
5.2.2	Linknet [47]	resnet34 [57]	JaccardLoss	0.902	0.821

5.2.2	Linknet [47]	resnet34 [57]	DiceLoss	0.909	0.834
5.2.2	Linknet [47]	resnet152 [57]	CrossEntropyLoss	0.904	0.826
5.2.2	Linknet [47]	resnet152 [57]	JaccardLoss	**0.918**	**0.849**
5.2.2	Linknet [47]	resnet152 [57]	DiceLoss	0.906	0.829
5.2.2	Linknet [47]	timm-regnetx_160 [58]	CrossEntropyLoss	0.902	0.823
5.2.2	Linknet [47]	timm-regnetx_160 [58]	JaccardLoss	**0.918**	**0.849**
5.2.2	Linknet [47]	timm-regnetx_160 [58]	DiceLoss	0.914	0.842
5.2.2	PAN [48]	resnet34 [57]	CrossEntropyLoss	0.896	0.811
5.2.2	PAN [48]	resnet34 [57]	DiceLoss	0.907	0.830
5.2.2	PAN [48]	resnet34 [57]	JaccardLoss	0.899	0.817
5.2.2	PAN [48]	resnet152 [57]	CrossEntropyLoss	0.897	0.815
5.2.2	PAN [48]	resnet152 [57]	JaccardLoss	**0.910**	**0.836**
5.2.2	PAN [48]	resnet152 [57]	DiceLoss	0.907	0.832
5.2.2	PAN [48]	timm-regnetx_160 [58]	CrossEntropyLoss	0.890	0.804
5.2.2	PAN [48]	timm-regnetx_160 [58]	JaccardLoss	0.901	0.821
5.2.2	PAN [48]	timm-regnetx_160 [58]	DiceLoss	0.903	0.825
5.2.2	DeepLabV3 [46]	resnet34 [57]	CrossEntropyLoss	0.875	0.778
5.2.2	DeepLabV3 [46]	resnet34 [57]	JaccardLoss	**0.893**	**0.808**
5.2.2	DeepLabV3 [46]	resnet34 [57]	DiceLoss	0.886	0.796

Table A.1.: The complete list of experiments for step 1. $\star$ shows the best overall result: Unet with resnet152 and JaccardLoss, achieving a DSC of 0.919. The architectures were used as implemented by Segmentation Models Pytorch [73]. An entire grid search was not feasible over all implemented architecture (n=9), encoder (n=549) and loss (n=8) combinations, therefore at first all encoders were tried with Unet and Dice loss. Then various other architectures were tried with a select few encoders which performed best.